Recommandation de Ressources Éducatives Libres dans le projet X5GON

Victor Connes

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies du numérique, de l'Information et de la Communication
Spécialité : *Informatique*

Par
Victor CONNES

Recommandation de Ressources Éducatives Libres dans le projet X5GON

Recommandation à visée pédagogique dans un contexte d'apprentissage non formel

Thèse présentée et soutenue à Université de Nantes, le 06/01/2023
Unité de recherche : LS2N, Nantes Université

Rapporteurs avant soutenance :

Jean-Cristophe Janodet	Professeur à l'université d'Ivry
Marc Tommasi	Professeur à l'université de Lille

Composition du Jury :

Président :	Elisa Fromont	Professeure à l'Université Rennes 1
Examinateurs :	Marc Tommasi	Professeur à l'université de Lille
	Marie Lefèvre	Maître de conférences à l'Université Claude Bernard Lyon
Dir. de thèse :	Colin DE LA HIGUERA	Professeur à Nantes Université
Co-dir. de thèse :	Hoël LE CAPITAINE	Maître de conférences à Nantes Université

Acknowledgement

Je tiens à remercier
I would like to thank. my parents..
J'adresse également toute ma reconnaissance à ....
....

Table des matières

INTRODUCTION

0.1 Contexte

L'apprentissage en ligne a connu ces dernières années un engouement sans précédent ; selon le bureau d'études économiques KPMG, le taux de croissance est de 900% depuis 2000, ce qui en ferait le taux de croissance le plus rapide dans l'industrie de l'éducation [1]. Selon ces mêmes projections, cette tendance est vouée à s'amplifier sur la période 2020-2025 avec une croissance annuelle de 29% sur la période. Cela est d'autant plus remarquable que l'étude a été faite avant la pandémie COVID-19 qui a encore accéléré la tendance. Dans de nombreux pays tels que les États-Unis, la France, l'Italie, la Pologne ou l'Ukraine, la réponse initiale des systèmes éducatifs du primaire, du secondaire et universitaire à la pandémie de COVID-19 a été la fermeture des écoles, collèges et campus et le passage rapide à l'enseignement à distance [2].

L'éducation ouverte à l'heure des FLOT. Les FLOT (Formation en Ligne Ouverte à Tous) [3] sont des outils majeurs de l'apprentissage en ligne (Ichimura and Suzuki, 2017). Les FLOT hébergent des cours gratuits en ligne, souvent sur un modèle *freemium* comportant l'accès gratuit aux ressources et aux formations, mais un système de certification payant. La formation est accessible gratuitement en ligne sans discrimination, les FLOT proposent un éventail large de cours : une majorité des cours proposés sont des ressources ouvertes. Cela a un intérêt double pour les plateformes : du point de vue économique, les ressources ouvertes permettent aux plateformes de réduire considérablement le coût de création des cours. D'un point de vue scientifique, les cours utilisés proviennent souvent d'universitaires de renom, spécialistes reconnus dans leur domaine, ce qui assure l'excellence scientifique des formations proposées. Néanmoins, on associe aussi au terme

1. source : KPMG https://assets.kpmg/content/dam/kpmg/pdf/2015/09/corporate-digital-learning-2015-KPMG.pdf
2. source : Mordor https://www.mordorintelligence.com/industry-reports/massive-open-online-course-mooc-market
3. de l'anglais MOOC (Massive Online Open Couseware)

FLOT d'autres types de formation en ligne ne reposant pas toujours sur des ressources ouvertes ; dans les faits, cette association est contradictoire avec l'acronyme « ouvert » et l'appellation « plateforme d'apprentissage » en ligne semble plus adaptée pour désigner les plateformes n'ayant pas recours à des ressources ouvertes. En pratique, les mêmes plateformes hébergent parfois des formations ouvertes et non ouvertes ; pour cette raison, l'ambiguïté semble persister dans les usages autour du terme FLOT.

À l'inverse des formations classiques, il est possible de suivre les FLOT depuis n'importe quel lieu et sans contrainte horaire (Li et al., 2015; Seaton et al., 2014). Cet aspect permet de mobiliser un très large public, notamment des publics non-étudiants (Seaton et al., 2014). Les FLOT offrent habituellement des contenus de formation limités en durée ; ils consistent en une séquence d'activités d'apprentissage balisées pour atteindre les objectifs de la formation (Coffield et al., 2004). Dans beaucoup de cas, les vidéos sont combinées à des contenus en ligne et un quizz valide chaque module ; la formation est alors pensée pour qu'un apprenant valide au minimum un module par semaine et un ensemble de mécanismes d'incitations sont prévus pour préserver l'assiduité et la motivation de l'apprenant (Ichimura and Suzuki, 2017). L'une des grandes promesses des FLOT réside dans la possibilité de créer une communauté pédagogique collaborative en connectant autour d'un sujet donné grâce aux outils du Web2.0 une communauté hétérogène et dispersée géographiquement. Enfin, les plateformes de FLOT tirent largement profit des ressources libres qu'elles hébergent. En pratique, les ressources libres constituent une large majorité des ressources disponibles. En effet, ces formations sont souvent proposées par des universités de renom ; elles permettent de crédibiliser la qualité des contenus sur la plateforme ; certaines d'entre elles devenues populaires monnayent également l'hébergement des contenus auprès des universités. On désigne ces ressources ouvertes sous l'appellation de **REL pour Ressources Éducatives Libres**, mais nous allons le voir ensuite, la notion de REL constitue une catégorie bien plus englobante.

0.2 Les RELs

La notion de REL est une initiative d'ampleur mondiale qui a émergé avec le développement du World Wide Web porté par divers acteurs de l'éducation (enseignants, pédagogues, universités, fondations...). Elle a pour but de créer et de distribuer des ressources éducatives libres et gratuites (les REL). Ce concept s'inscrit dans une dynamique plus large pour intégrer les technologies de l'information et de la communication dans

les programmes pédagogiques et démocratiser l'accès aux savoirs dans une société de la connaissance. Il est souvent affilié aux concepts de logiciel libre, d'open source, des données ouvertes et du libre accès (Wiley, 2014). L'éducation durable et la large diffusion du savoir sont définis comme le quatrième objectif de développement durable de l'humanité à l'horizon 2030 par l'UNESCO [4]. L'UNESCO est devenue un acteur important dans l'écosystème des REL qui fait de cet outil un élément central de sa stratégie.

Pour cette raison, nous allons utiliser la définition des REL proposée par l'UNESCO :

— « des matériels d'enseignement, d'apprentissage et de recherche sur tout support, numérique ou autre, existant dans le domaine public ou publiés sous une licence ouverte permettant l'accès, l'utilisation, l'adaptation et la redistribution gratuite par d'autres, sans restriction ou avec des restrictions limitées »

(UNESCO, 2017).

Cette définition très englobante permet de caractériser différents types de contenus comme des REL tels que : des manuels en libre accès, des notes de cours, des présentations, des vidéos, des fichiers audio, des illustrations, des animations, des examens, des exercices, des pages web, des logiciels... Cela fait des REL une matière première pour l'éducation largement composite. La grande force des REL est qu'elles peuvent être largement diffusées et adaptées parce qu'elles sont gratuites pour les usagers et ne font pas l'objet des restrictions de droit d'auteur habituelles. Les REL dépassent donc très largement le cadre des FLOT ou plus généralement du libre accès, en mettant l'accent sur la libre diffusion et modification des ressources. Cette ouverture est le plus souvent indiquée à l'aide d'une licence ouverte telle que les licences *Creative Commons*.

0.2.1 Les licences ouvertes

D'un point de vue légal, chaque fois qu'une œuvre est créée, elle est automatiquement protégée par le droit d'auteur. La protection du droit d'auteur empêche d'autres personnes d'utiliser l'œuvre de certaines manières, par exemple en la copiant ou en la mettant en ligne sans l'autorisation du créateur ; on dit que l'œuvre est « tous droits réservés » au sens où l'auteur possède l'intégralité des droits de propriétés intellectuels de l'œuvre. Ces droits s'étendent sur une durée variable en fonction des pays et des situations, mais dépassent souvent très largement la vie de l'auteur (Wiley, 2014). Dans beaucoup de pays, dont la France, la règle générale veut qu'ils échoient 70 ans après la mort de l'auteur. Une fois ce

4. https ://en.unesco.org/sustainabledevelopmentgoals

délai écoulé, les œuvres tombent alors dans le domaine public : on dit qu'elles sont libres de droits, chacun est alors libre de diffuser, copier, utiliser l'œuvre à sa convenance sans autorisation de l'auteur. Lorsqu'elles peuvent être utilisées pour un usage éducatif, les œuvres dans le domaine public sont un exemple de REL. Néanmoins, la grande majorité des REL sont en pratique des contenus sous licence ouverte. Lorsqu'un créateur attribue une licence ouverte à son œuvre, il précise comment il souhaite que les autres la réutilisent. Cela fait passer le droit d'auteur de « tous droits réservés » à « certains droits réservés ». Il existe une large variété de licences ouvertes et une variété encore plus large de licences non ouvertes. Les licences *Creative Commons* sont un bon exemple de licence libre, elles reposent sur quatre critères majeurs [5] :

L'attribution : L'attribution, souvent abrégée par le mot anglais « BY », signifie que toute personne souhaitant utiliser l'œuvre doit fournir le crédit approprié, ainsi qu'un lien vers la licence et indiquer si des modifications ont été apportées. En pratique, on parle parfois de « droit de citation ».

L'utilisation non-commerciale : L'utilisation non-commerciale, souvent abrégée par « NC », exprime l'impossibilité d'utiliser l'œuvre à des fins commerciales. Ainsi, une licence « NC » impose que l'œuvre ne peut pas être utilisée dans le cadre d'un usage ultérieur commercial durant la période de validité de la licence.

La contamination : La contamination ou dérivation, souvent abrégée par « SA » (pour *share alike* signifiant partager avec les mêmes droits), exprime la nécessité de repartager tout travail dérivé de l'œuvre originale avec la même licence. Ainsi, une œuvre « SA » garantit que toute œuvre dérivée portera la licence originale : ainsi si l'œuvre est sous licence libre, toutes les œuvres dérivées seront, elles aussi, sous licence libre. Par exemple, si la licence possède à la fois les critères « NC » et « SA », l'auteur impose que l'œuvre ou une œuvre dérivée ne pourra jamais être utilisée à des fins commerciales.

La non-dérivation : La non-dérivation (« ND ») pour *Non-Derivative* désigne l'impossibilité de partager du contenu provenant d'une version modifiée, réarrangée ou agrégée de l'œuvre sans accord préalable de l'auteur. En particulier, l'œuvre doit être utilisée uniquement dans les conditions de son usage initial. Toutes les œuvres sous licences portant la mention « ND » ne sont pas considérées comme des REL, car leurs auteurs ne permettent pas la modification et le repartage qui

5. Source : https://creativecommons.org/licenses/

sont au centre de la philosophie des REL dont nous discuterons en Section 0.2.2). Par exemple, la non-dérivation empêche la traduction ou la critique de l'œuvre.

À partir de ces quatre critères, plusieurs licences peuvent être définies avec des contraintes différentes de réutilisation et de partage (voir Figure 1 pour un tableau récapitulatif des licences *Creative Commons*). Certaines de ces licences ouvertes permettent la modification et le partage, les œuvres protégées par ces licences pouvant être utilisées dans une démarche éducative sont alors des REL. Pour la suite de ce document, lorsque nous parlerons de contenu ouvert, nous ferons spécifiquement mention d'un contenu sous licence ouverte ou disponible dans le domaine public.

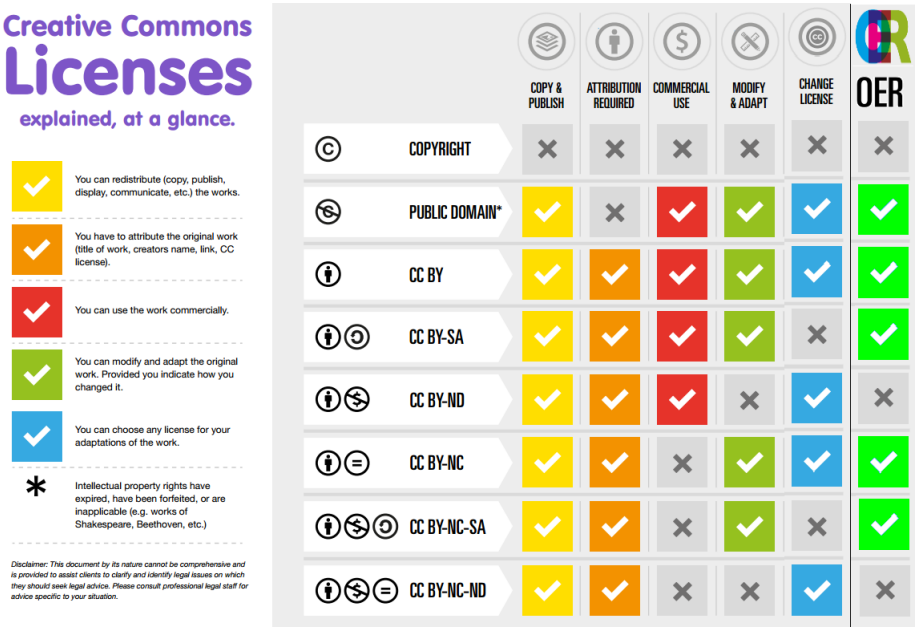


FIGURE 1 – Licences Creative Commons. Source : Image modifiée depuis https://creativecommons.org/

0.2.2 Le cercle vertueux des REL

La philosophie des REL repose sur un cercle vertueux dans la diffusion de la connaissance en 5 étapes, les 5R (Wiley, 2014) :

Rétention : Chacun doit avoir la capacité d'accéder au contenu et de conserver sa propre copie. La capacité de retenir de l'information (de l'anglais Retain) est un aspect central dans un processus d'apprentissage. La publication de contenu éducatif

libre permet sa conservation, sa classification et son accès de manière extrêmement large. Chacun doit pouvoir librement en conserver une copie, et s'approprier ainsi largement le contenu ou le diffuser. Ce droit de conserver sa propre copie gratuitement est une spécificité des contenus ouverts (Wang and Towey, 2017). En effet, pour la plupart des œuvres restant sous le régime des droits d'auteurs, le droit d'accès est un droit réservé au possesseur des droits, souvent monnayable. C'est par exemple le cas lorsque l'on va voir un film au cinéma. De la même manière, le droit à la rétention est, lui aussi, un droit monnayable, c'est le cas lorsque l'on achète une VOD. Dans le cas des contenus ouverts, ces droits sont légués au domaine public par l'auteur : ainsi, il n'existe plus une source centrale de distribution de l'œuvre, mais toute une diversité d'acteurs susceptibles de posséder une copie de l'œuvre. Chacun de ces acteurs est alors libre d'indexer cette œuvre et de la mettre en lien avec d'autres œuvres selon ses propres volontés. Cela apporte deux avantages : le premier concerne une robustesse plus importante en termes de conservation de l'œuvre, une œuvre ainsi recopiée et distribuée (en particulier de manière numérique) a moins de chance d'être perdue ou altérée.

Le deuxième avantage dérive directement du nombre et de la variété des acteurs possédant une copie de l'œuvre, chaque nouvel acteur est susceptible d'apporter un regard différent sur l'œuvre ou de l'utiliser dans un contexte jusqu'ici inédit. Mieux encore, chaque acteur est susceptible d'apporter sa propre contribution sur l'œuvre, c'est notre point suivant : la révision.

Révision : De cette manière, le contenu peut largement être amélioré, modifié, traduit ou adapté par la communauté pour des usages différents des usages initiaux. Par exemple, chaque apprenant peut à son tour diffuser les connaissances acquises, cela donne à l'apprenant un rôle actif dans le processus de création du contenu pédagogique. Cette possibilité de révision peut par exemple permettre une mise à jour de la ressource : imaginons une carte des câbles de télécommunications. Par nature, cette carte est amenée à évoluer lors de l'implantation d'un nouveau câble. Le plus souvent, lorsque l'œuvre n'est pas ouverte, cette mise à jour n'est pas faite, seul le possesseur des droits a la possibilité de produire une œuvre dérivée de l'œuvre originale. Mettre à jour la carte engendre pour lui un surcoût de temps important incluant un travail de veille et de republication. Si la carte est une œuvre ouverte, ce travail est partagé sur l'ensemble de la communauté.

Ré-mixer : Chacun doit avoir la liberté d'agréger ou de réarranger divers contenus

pour créer une nouvelle ressource. Il est également possible d'agglomérer différentes REL pour créer quelque chose de nouveau -typiquement, lors de la création d'un cours, d'un site web ou d'un contenu de vulgarisation.

Ré-utiliser : Tout un chacun doit avoir la liberté de réutiliser ou de diffuser sa copie révisée, réarrangée. Cela fournit en particulier un ensemble de ressources clé en main pour les enseignants qui peuvent facilement les intégrer à leurs cours. Plus généralement, il est possible d'inclure le contenu dans un site ou une vidéo en mentionnant simplement l'auteur et la licence.

Redistribuer : enfin, le nouveau contenu peut, lui aussi, être mis en ligne et entamer un nouveau tour de boucle.

Ce cercle vertueux a pour conséquence de rendre l'accès aux ressources éducatives moins onéreux (par la gratuité des droits d'accès et d'utilisation) tout en fournissant des contenus personnalisés et adaptés au contexte de chacun, notamment en termes de mode, de langue et de format d'apprentissage (Wang and Towey, 2017). L'adaptation de REL existantes et la création de nouvelles ressources sont également des occasions de rendre le matériel de cours plus accessible, inclusif et représentatif pour les apprenants, une flexibilité qui manque souvent dans les manuels scolaires traditionnels. Comme les REL ne sont pas destinées au marché commercial, elles peuvent traiter de sujets et ouvrir des perspectives qui seraient autrement négligées. C'est là un autre point fort des REL : permettre de découpler le processus de création de savoir d'une démarche commerciale (Jhangiani et al., 2018). Dans le prochain paragraphe, nous allons nous intéresser plus en détail aux intérêts des REL du point de vue économique et social.

0.2.3 Intérêts économiques et sociaux des RELs

D'un point de vue économique et social, les REL présentent plusieurs avantages. Tout d'abord, elles réduisent le coût d'accès à l'éducation, produisant une éducation plus égalitaire. Plusieurs études ont montré que, dans le cas des manuels scolaires, les ressources éducatives libres étaient au moins d'aussi bonne qualité que les manuels traditionnels ; elles avaient l'avantage de réduire le nombre d'abandons chez les étudiants en proposant du contenu plus inclusif et levant la charge financière de l'achat des manuels (Clinton and Khan, 2019; Jhangiani et al., 2018; Hendricks et al., 2017). Pour les étudiants qui ont déjà du mal à faire face à la hausse des frais de scolarité et de logement, les dépenses supplémentaires liées aux manuels scolaires peuvent aussi constituer un obstacle à

l'accès à l'enseignement supérieur. Dans une étude menée par Florida Virtual Campus en 2012 (Donaldson et al., 2012), 65% des personnes interrogées ont indiqué qu'elles n'avaient pas acheté de manuels à un moment donné de leur scolarité en raison de leur coût. La même enquête a également montré que 35% des étudiants ont réduit leur charge de cours semestrielle en raison du coût des manuels et que 23% d'entre renonçaient régulièrement à acheter des manuels uniquement en raison de leur coût. Selon une étude de consommation menée par NBC News en 2015, de janvier 1977 à juin 2015, les prix des manuels scolaires aux États-Unis ont augmenté de 1 041 % ; ce qui correspond à plus de trois fois le taux d'inflation américain [6]. Les exemples cités jusqu'ici sont particulièrement pertinents dans les systèmes éducatifs anglo-saxons - en particulier américains, plus onéreux qu'en France. Néanmoins, les études sur la qualité des contenus peuvent largement être généralisées dans un contexte plus large. Les REL offrent d'autres avantages, notamment sur un plan pédagogique.

L'un des principaux avantages en terme d'enseignement est le suivant : puisque les ressources ouvertes sont entièrement révisables et ré-arrangeables, elles peuvent être personnalisées pour s'adapter à la manière dont un instructeur souhaite enseigner un cours. Lorsqu'ils utilisent des ressources traditionnelles statiques non modifiables, ces derniers peuvent être contraints d'enseigner leurs cours d'une manière conforme aux ressources disponibles, plutôt que d'une façon plus appropriée pour eux et pour les apprenants (Jhangiani et al., 2018). Cela crée une norme de l'éducation forcément défavorable pour les apprenants non traditionnels. Au contraire, les REL permettent une personnalisation et une large diffusion susceptible de trouver du contenu adapté pour tout type de public. L'utilisation des REL offre la liberté de réviser le matériel en supprimant le contenu non pertinent ou en ajoutant son propre contenu. Elles donnent ainsi la flexibilité de combiner ensemble différentes ressources, permettant ainsi un matériel contextualisé à un cours spécifique. Comme le monde, y compris les étudiants, peuvent participer à la création, à la révision et à la distribution des REL, les enseignants peuvent également utiliser ces ressources pour s'engager dans la « pédagogie ouverte », c'est-à-dire des travaux qui tirent parti des REL pour créer des expériences d'apprentissage plus engageantes. Traditionnellement, les étudiants travaillent dur sur des devoirs qu'ils remettent à leurs instructeurs, que ces derniers notent et qu'ils ne reverront jamais. Au lieu de cela, les enseignants peuvent, par exemple, demander aux étudiants d'éditer des REL en vue de leurs redistributions, ou

6. https://www.nbcnews.com/feature/freshman-year/college-textbook-prices-have-risen-812-percent-1978-n399926

d'accorder une licence ouverte à leurs propres travaux pour qu'ils puissent être utilisés par d'autres étudiants, permettant ainsi un partage avec un plus large public de leur travail. Les étudiants deviennent ainsi des participants actifs au partage des connaissances scientifiques et s'intègrent naturellement à des communautés d'apprentissages favorisant leur motivation (Clinton and Khan, 2019). Les REL offrent aussi des avantages aux membres des communautés au-delà des campus, des collèges et des universités, en permettant la création et le partage de connaissances en dehors des limites des classes traditionnelles. Il peut être difficile pour ceux qui ne font pas partie des communautés savantes d'accéder et de participer au matériel d'apprentissage ou à la recherche. Cela est d'autant plus dommageable qu'on observe une demande croissante pour des contenus de vulgarisation de la connaissance au grand public (Blanchard et al., 2018). La création de matériel de recherche et d'enseignement sous licences libres contribue donc à faire tomber ces barrières. En ce sens, les REL font parties des réponses à la nécessité de formation tout au long de la vie et au travail de partage de connaissances du milieu universitaire. Cela permet un accès plus large à l'information et à la recherche, ainsi qu'une participation accrue aux travaux d'érudition, notamment des initiatives de recherche participative, ce qui aide les universités à étendre leurs missions fondamentales à l'ensemble de la société.

0.2.4 Engagement en faveur des REL

Les REL ont bénéficié d'un fort soutien politique, notamment, avec la déclaration de Paris de 2012 en faveur des REL appelant les gouvernements du monde entier à accorder des licences ouvertes pour le matériel éducatif financé par des fonds publics, en vue d'une utilisation dans le domaine public [7]. En 2017, l'UNESCO a réaffirmé son engagement en faveur des REL en rédigeant le plan d'action de Ljubljana [8]. Ce plan détaille 41 actions recommandées pour intégrer les ressources sous licence ouverte afin d'aider tous les états membres à construire des sociétés du savoir et à atteindre l'objectif de développement durable 4 de 2030 sur « l'éducation de qualité et tout au long de la vie ». Enfin, en 2019, l'UNESCO a publié un ensemble de recommandations à destination des états et des institutions identifiant les bonnes pratiques permettant de passer aux ressources éducatives libres [9] et réaffirmant encore son engagement. Ces actions politiques ont abouti à l'adop-

7. https://en.unesco.org/oer/paris-declaration
8. https://en.unesco.org/news/ljubljana-oer-action-plan-2017
9. http://portal.unesco.org/en/ev.php-URL_ID=49556&URL_DO=DO_TOPIC&URL_SECTION=201.html

tion par les pays membres de l'UNESCO de politiques éducatives en faveur des REL.

Dans ce contexte, le nombre de ressources éducatives libre ne cesse de croître, de 50 millions en 2006, à 400 millions en 2010, puis 882 millions en 2014 [10]. Différents acteurs ont contribué à cet essor : des gouvernements, des universitaires, des collectifs ou individuels. On distingue aujourd'hui plus de 9 millions de sites hébergeant des REL.

0.2.5 Quelques limitations de l'écosystème REL

L'écosystème des REL est aujourd'hui très éparpillé et cela conduit à des inégalités tant sur la provenance des ressources (37% Amérique du Nord, 34% Europe, 16% Asie Pacifique, 10% Amérique Latine, 2% Afrique du Nord, 1% reste de l'Afrique) que sur les langues utilisées (prédominance de l'Anglais et des langues occidentales). La grande majorité des REL sont en anglais et basées sur la culture occidentale, ce qui limite leur pertinence et risque de reléguer les pays moins développés au rôle de consommateurs. Cependant, un certain nombre de projets existent désormais dans les pays en développement pour développer des REL basées sur leurs propres langues et cultures.

Des difficultés sont également rencontrées au niveau de l'adoption par les professeurs et les instructeurs. Il s'agit d'un mélange d'obstacles réels ou causés par la perception qu'ont les enseignants des REL et de la pédagogie ouverte (Wang and Towey, 2017). Les REL sont souvent perçues comme difficiles à trouver (Wang and Towey, 2017; Belikov and Bodily, 2016). Les utilisateurs qui ne les connaissent pas peuvent percevoir comme des obstacles à l'adoption le fait de savoir où trouver le matériel de cours pertinent, ainsi que le temps nécessaire pour ce faire. Cela s'explique par la large variété des pratiques et des acteurs, mais aussi par le manque de système d'indexation et de référencement des REL.

En effet, vue la croissance rapide du nombre de ressources et de dépôts de données, il devient difficile pour un utilisateur de trouver celles pouvant le concerner, qui représentent souvent une faible fraction de celles disponibles, donc noyées dans la masse. Dans ce contexte, les algorithmes de filtrage de l'information proposent aux utilisateurs d'effectuer un premier filtrage des données, pour ne présenter qu'un ensemble de ressources minimal potentiellement intéressant. Parmi ces algorithmes, on distingue deux grandes familles : les *moteurs de recherche* qui tentent de répondre à une requête explicite de l'utilisateur et les *systèmes de recommandation* qui agissent sans requête spécifique de l'utilisateur. Les méta-données (informations descriptives sur les ressources) peuvent améliorer les perfor-

10. Source : https://stateof.creativecommons.org/

mances de ces algorithmes de filtrage : dans le cas des REL on observe souvent une absence d'uniformisation et de rigueur dans leur utilisation, qui rendent très complexe leurs utilisations ultérieures (Simão de Deus and Francine Barbosa, 2020). En pratique, nombre de ressources sont annotées de manière *ad-hoc* ou simplement non annotées (Simão de Deus and Francine Barbosa, 2020). Lorsque l'annotation avec des méta-données n'est pas faite au moment de la publication, il reste la possibilité d'annoter manuellement les données en ligne. Cette tâche est extrêmement pénible, chronophage et coûteuse en cas de nombreuses ressources. De plus, l'annotation nécessite souvent une double expertise : en premier lieu, celle sur le ou les domaines traités par la ressource, mais aussi une celle sur les conventions d'annotations utilisées dans ce domaine. D'autres approches, telles que les métadonnées générées automatiquement et les folksonomies constituent des pistes de recherches prometteuses, en particulier pour le cas des articles scientifiques (Nasar et al., 2018) ou sur les vidéos (Maratea et al., 2013). Reste à voir s'il s'agit de solutions capables de s'adapter à une large variété de contenus, de langues et de domaines.

Un autre aspect important de l'écosystème REL qui peut complexifier la recherche de ressources pertinentes est le manque de connexion entre les répertoires. Parmi les 9 millions de sites web qui constituent aujourd'hui l'écosystème des REL, on observe une large variété d'acteurs et de pratiques. L'absence d'harmonisation- notamment en termes de méta-donnée- fait que ces répertoires sont très peu interconnectés. Par conséquent, il s'avère particulièrement difficile d'effectuer des recherches multirépertoires. Ainsi, lorsqu'un utilisateur effectue sa recherche, il se retrouve souvent limité aux frontières d'un seul répertoire et ne bénéficie donc pas de la diversité censée faire la force de l'éducation ouverte.

La difficulté pour rechercher les ressources a tendance à profiter aux plus gros acteurs, naturellement les plus visibles tend à renforcer les inégalités.

Développer un écosystème plus interconnecté favorisant la diversité et l'accès aux ressources de qualité est donc un challenge important pour les REL.

0.3 Le projet X5gon

C'est dans ce contexte que le projet européen X5GON a été lancé par 8 partenaires (University College of London, Institut Jozef Stefan de Ljubljana, fondation Knowledge4all, le ministère de l'Éducation Slovène, Universitât Osnabruck, l'Université de Nantes, Posta Slovenije, Universitat Politecnica de Valencia). Son but était le développement d'un réseau

de REL intégrant, un travail d'indexation des ressources, le déploiement de technologie de recommandation et de personnalisation de l'apprentissage pour guider les utilisateurs à travers les ressources du réseau. Pour ce faire, X5GON a défini 5 engagements majeurs correspondant aux 5 X (X pour « accross »). Nous allons ici les détailler brièvement, puis nous analyserons plus en détail chacun d'entre eux dans la Section 0.5 :

Inter-modalité *(X-modal)* : Le système doit pouvoir tenir compte de l'aspect multimodal des contenus.

Multi-site *(X-site)* : Il doit avoir la capacité d'être facilement adaptable à ces ressources éducatives ouvertes. En ce sens X5GON s'attribue une tâche de promoteur des ressources éducatives, mais pas de producteurs ni d'hébergeur.

Multi-disciplinarité *(X-domain)* : Le système doit être capable de pouvoir s'adapter à la large diversité des domaines observés dans les ressources éducatives.

Multi-langue *(X-lingual)* : Le système doit être capable d'indexer et de recommander du contenu multilingues en favorisant la plus grande pluralité, en s'appuyant sur les technologies de traduction et de transcription automatique qui permettent de réduire les barrières inter-langues.

Multi-culturalité *(X-cultural)* : Le système doit pouvoir s'adapter aux différentes cultures tant en termes de pratiques d'apprentissage que de préférences.

Si les 5 engagements ont des points de convergence, ils reflètent chacun un aspect spécifique de l'écosystème REL. Afin de répondre à ces engagements, X5GON a proposé le déploiement de plusieurs services automatiques basés sur des technologies d'intelligence artificielle pour la réalisation de plusieurs tâches charnières :

— la traduction et la transcription des ressources, en particulier dans les langues peu dotées,
— l'indexation automatique des ressources,
— le développement de moteur de recherche de REL,
— le développement de système de recommandation de contenu.

La motivation des porteurs de ce projet pour l'utilisation de méthodes automatiques est double : avec plus de 9 millions de ressources, l'annotation manuelle par les utilisateurs est une tâche extrêmement chronophage, d'autant plus que le nombre de ressources ne cesse d'augmenter. D'autre part, les méthodes automatiques dans des domaines tels que la recommandation et la traduction ont fait des progrès très importants ces dernières années. Les méthodes actuelles permettent aujourd'hui des exploits qui paraissaient encore invrai-

semblables 10 ans auparavant. Par exemple, il est aujourd'hui possible à deux personnes ne parlant pas une langue commune de converser en temps réel, chacune dans leur propre langue, sans avoir recours à des traducteurs par l'intermédiaire d'une application (Jia et al., 2019). Les progrès en la matière sont tels que dans certains cas (qui ne reflètent pas l'ensemble des contraintes réelles, mais donnent une idée des progrès dans le domaine) la traduction automatique semble surpasser la traduction faite par des professionnels lors d'évaluations en double anonymat. Il semble en particulier que les algorithmes produisent souvent des traductions moins fluides que les professionnels, mais qui respecte mieux le sens du texte original. Les traductions des algorithmes approchent tellement celle des professionnels qu'il semble difficile- même pour un panel d'experts- d'en faire la distinction (Popel et al., 2020). Ces méthodes automatiques ont des avantages indéniables en termes de coûts et de mise à l'échelle, ce qui permettrait le déploiement pour le plus grand nombre.

Dans le projet, la cellule nantaise a été responsable du moteur de recommandation. Dans un contexte comme celui d'X5GON, nous définissons la tâche de recommandation sous le terme de *recommandation à visée pédagogique pour un contexte d'apprentissage non formel*, nous expliquerons en détail les motivations nous ayant conduit à choisir ce terme dans la section suivante (Section 0.4). Nous allons voir que la résolution de cette tâche constitue un idéal à atteindre : ce document et les contributions que nous proposons sont construits dans le but d'une atteinte de cet objectif. Dans la prochaine section, nous allons prendre le temps de le définir plus en détail.

0.4 Recommandation à visée pédagogique pour un contexte d'apprentissage non formel

La tâche de recommandation est aujourd'hui une tâche centrale pour beaucoup d'entreprises. À titre d'exemple, les grandes entreprises multinationales du numérique - notamment celle communément désignées sous le terme GAFAM (Google, Amazon, Facebook, Apple et Microsoft) - concentrant une partie importante de l'activité web, sont parmi les entreprises les plus prospères aux mondes. Elles dégagent toute une partie importante de leurs revenus des algorithmes de recommandation. Ceux-ci sont une sous-famille des algorithmes de recherche d'information. Ils sont devenus indispensables dans un contexte dans lequel le nombre de ressources disponibles est pléthorique et où il est très difficile de trouver du contenu pertinent sans l'aide d'un processus automatique. À la différence des

moteurs de recherche qui doivent proposer du contenu en réponse à une requête explicite, typiquement une phrase composée de quelques mot-clés, les systèmes de recommandation cherchent à proposer du contenu sans demande explicite, mais en tirant profit de l'ensemble des informations contextuelles disponibles. Celles-ci peuvent être propres à l'utilisateur (historique, préférences, informations démographiques...), au contenu (popularité, type, durée...) ou au contexte (jour de la semaine, pays...).

Certains pensent souvent que les systèmes de recommandation sont en pratique moins influents que les moteurs de recherche. Cet aspect implicite est néanmoins une force importante des systèmes de recommandation, ce qui fait d'eux les algorithmes parmi les plus utilisés. À titre d'exemple, 80 000 requêtes chaque seconde sont faites sur le moteur de recherche Google (65% de parts de marché), 90 000 requêtes par seconde sur le moteur de recommandation de Youtube (également le plus gros du marché) [11]. Sur ce même Youtube 70% des plus d'un milliard de vidéos vues chaque jour proviennent d'une recommandation [12]. Bien entendu, la comparaison du nombre de requêtes avantage les systèmes de recommandations : pour chaque requête sur un moteur de recherche, il faut une demande spécifique et explicite de l'utilisateur. À l'inverse, les requêtes d'un système de recommandation sont automatiques et ne nécessitent pas d'effort explicite de l'utilisateur. Reste que nous sommes plus fréquemment exposés aux filtrages des systèmes de recommandations qu'à ceux des moteurs de recherche. Ce filtrage a pourtant un rôle primordial sur l'information que nous choisissons de voir, comme en témoignent les 70% de vues provenant de l'algorithme de recommandation sur la plateforme Youtube. De plus, l'aspect implicite de la tâche de recommandation réduit le contrôle de l'utilisateur sur ce filtrage, ce qui rend ces algorithmes très influents sur la propagation de l'information. Nous verrons plus en détail les algorithmes de recommandation et les spécificités de cette tâche en Section 1.3.

Nous allons ici introduire brièvement quelques notions clés que nous utiliserons tout au long du document. Tout d'abord, nous parlerons souvent de *recommandation à large-échelle*. Nous utilisons ce terme pour désigner spécifiquement les systèmes de recommandation capables de répondre à des millions de requêtes par seconde pour des millions d'utilisateurs sur des millions de ressources. Nous parlerons plus précisément des spécificités de ces algorithmes capables de traiter une aussi large quantité de requêtes, en particulier en Section 1.3.4.3 et discuterons de la pertinence de ce type de système dans

11. https://www.blogdumoderateur.com/chiffres-google/
12. https://www.blogdumoderateur.com/chiffres-youtube/

notre contexte dans la section suivante (Section 0.5).

Par ailleurs, nous utiliserons beaucoup les termes *recommandation dans un cadre non structuré*, pour désigner une tâche de recommandation dans laquelle les ressources ne sont pas homogénéisées en termes de méta-données ou comportent peu de méta-données, et celui *recommandation pour un contexte d'apprentissage non formel* pour désigner une tâche de recommandation où le cadre de l'activité d'apprentissage n'est pas défini au préalable par l'enseignant. Par opposition, nous appellerons un *cadre de recommandation formel*, une tâche de recommandation pour une activité pédagogique balisée - par exemple sur un sous-ensemble prédéfini de ressources sélectionné au préalable par des enseignants. Cette distinction provient d'une taxonomie des formes d'apprentissage reconnu par l'Organisation de Coopération et de Développement Economiques (OCDE) qui distingue trois formes d'apprentissages :

L'apprentissage formel : L'apprentissage formel se déroule dans un environnement structuré dont le but explicite est l'enseignement, l'apprenant et les intervenants étant tous deux conscients de s'engager dans un processus d'apprentissage. On parle souvent par extension d'éducation formelle qui se déroule typiquement dans un environnement scolaire, avec des classes composées de plusieurs élèves qui apprennent ensemble avec un enseignant formé et certifié dans la matière.

L'apprentissage informel : L'apprentissage informel est caractérisé par un faible degré de planification et d'organisation du temps, du contexte et des objectifs d'apprentissage. L'apprentissage résulte de l'engagement et de la réflexion dans des activités quotidiennes, dans lesquelles l'apprentissage n'est pas l'objectif principal. Il est entrepris de manière autonome, soit individuellement, soit collectivement, mais sans instructeur ou formateur. Il se produit souvent de manière spontanée et inconsciente, sans intentionnalité de l'apprenant.

L'apprentissage non-formel : L'apprentissage non formel comprend diverses situations d'apprentissages structurées qui n'ont pas le niveau de certification ou de reconnaissance associée à l'apprentissage formel, et plus structurées que celles associées à l'apprentissage informel. Elles se distinguent souvent par une volonté de l'apprenant d'accroître ses connaissances ou ses compétences sur un sujet spécifique. C'est typiquement le cas des FLOT, mais aussi des colonies pour enfants par exemple. Ce troisième cadre d'apprentissage recoupe parfaitement les activités d'apprentissage en lien avec la recommandation proposé par X5GON.

Enfin, nous parlerons aussi de *recommandation à visée pédagogique*. Nous définissons

ce terme en opposition au terme *recommandation à usage commercial* dans laquelle la plateforme cherche à conjuguer recommandation et objectif commercial : un grand nombre de ventes d'articles, un grand nombre de ressources consultées et donc de revenus publicitaires, etc. La recommandation à usage commercial est un cas d'application typique des méthodes à grande échelle. Dans la recommandation à visée pédagogique, l'objectif unique est d'améliorer l'expérience d'apprentissage d'un utilisateur. Définir mathématiquement cet objectif afin de pouvoir l'évaluer est une tâche complexe, créer des modèles capables de maximiser cet objectif l'est encore plus. Dans la prochaine section, nous allons voir les verrous scientifiques en lien avec la recommandation à visée pédagogique pour un contexte d'apprentissage non formel sur des données non structurées. Pour simplifier la lecture dans la suite, nous raccourcirons en *recommandation à visée pédagogique pour un contexte d'apprentissage non formel* en l'absence d'ambiguïté.

0.5 Verrous scientifiques

La question du système de recommandation dans le projet soulève des problématiques singulières. Premièrement, le système doit répondre aux 5 engagements majeurs. Chacun de ces engagements soulève plusieurs questions de recherche que nous allons discuter dans les sections suivantes (Sections 0.5.1, 0.5.2, 0.5.3, 0.5.4 et 0.5.5). En plus de ces engagements, la recommandation dans ce projet doit répondre à des contraintes techniques, en particulier le démarrage à froid et la nécessité d'un fonctionnement à large-échelle, qui seront discutés en Section 0.5.6. Parmi l'ensemble des questions de recherche émanant des contraintes du projet, deux questions en particulier feront l'objet de ce document (Section 0.6).

0.5.1 Engagement en faveur de l'inter-modalité

Les difficultés liées à l'intermodalité résident dans la large variété des REL : un podcast, une présentation vidéo, un poster, une image, un gif, un cours complet sous forme de page web, un manuscrit de thèse de 200 pages sont autant d'exemples de REL. Représenter informatiquement ces ressources d'une façon permettant de les comparer et d'évaluer leurs similarités sémantiques de manière non supervisée est une tâche de recherche complexe, même lorsque les ressources partagent la même modalité. Cette tâche est d'autant plus compliquée lorsque les données sont multimodales : beaucoup d'approches dans le

domaine s'appuient sur de l'annotation manuelle pour faire de la tâche une tâche semi-supervisée (Liao et al., 2021). Cette annotation est extrêmement chronophage dans notre contexte à large-échelle. De plus, nombre de ressources sont composites, ainsi un cours peut être un site web composé de textes, de ressources et d'images, ou alors une vidéo, dans laquelle un diaporama est projeté. Bien sûr, les diapositives composant le diaporama peuvent, elles aussi, contenir des images, des formules et des vidéos. Être capable d'identifier une brique de base insécable d'apprentissage est difficile ; dans notre cas, la plupart des ressources sont multimodales.

Cette multimodalité s'exprime également au niveau du mode d'apprentissage, et pas uniquement au niveau du mode de communication. Ainsi certaines ressources sont des exercices, d'autres des cours, d'autres encore des cas études... Dans notre contexte, les données sont non structurées, cela signifie en particulier que les liens entre les ressources ne sont pas connus. Par exemple, une information spécifiant qu'une image A est utilisée dans une vidéo B et dans un cours C permet de faire un lien naturel entre les ressources. De plus, une information spécifiant qu'un exercice convient bien à une notion abordée dans ce pdf serait très intéressante pour mieux représenter les contenus entre eux et proposer éventuellement des autoévaluations des compétences d'un apprenant. Malheureusement - de manière quasi-systématique du fait du caractère non structuré de nos données - ces informations qui permettraient de caractériser les liens entre les ressources sont complètement absentes. Cette absence est due en pratique à la très faible utilisation des méthodes d'étiquetages par métadonnées spécifiques à l'éducation.

Des méthodes automatiques cherchent à développer des technologies de détection automatique de référence, d'organisation de la connaissance, ou de transcription de contenu d'une modalité dans un autre. À ce jour, les technologies les plus avancées dans ce domaine utilisent souvent le texte comme représentation finale : des méthodes tentent de décrire les images en textes ou de transcrire le discours d'une vidéo. Pour cette raison, il a été choisi au sein du projet X5GON d'utiliser le texte comme représentation pivot de nos ressources en utilisant ces méthodes. Ce choix est d'autant plus pertinent que la plupart des REL récupérées dans le projet sont aujourd'hui soit des vidéos, soit des données textuelles (comme détaillé en Section 2.4.1).

0.5.2 Engagement multi-site

L'aspect multisite pose des problématiques à plusieurs niveaux. Tout d'abord, du point de vue de l'indexation, les ressources étant éparpillées et les pratiques de stockage

diverses (répertoires de liens, bases de données, ressources sur forme de page web), il est difficile de créer automatiquement un corpus de ressources éducatives libres. Pour comprendre la diversité des pratiques quant à l'hébergement des REL il faut tenir compte de la diversité des acteurs et des formats. Dans les prochains paragraphes, nous allons détailler quelques cas typiques, en nous intéressant en particulier pour chaque cas aux questions suivantes :

— Comment trouver la licence ?
— Comment connaître le public cible ?
— Est-il facile de définir la limite de la ressource ?
— Quelle est la finalité de la ressource (exercice, vulgarisation, cours...) ?
— Quel sont les liens entre les ressources ?

Ces questions et ces exemples ne constituent pas une revue exhaustive de l'ensemble des pratiques et des problématiques en lien avec l'aspect multisite. Néanmoins, ils ont pour finalité de permettre aux lecteurs de comprendre les difficultés inhérentes à cette diversité pour l'indexation des REL et leur utilisation dans le cadre de processus de traitement automatique.

Considérons par exemple un dépôt d'archives ouvertes. Sur un tel dépôt, il n'y a pas nécessité de préciser le niveau de difficulté de la ressource : il est implicite que toutes les ressources sont de niveau universitaire, ce sont des articles scientifiques. Il en va de même pour leurs finalités. De la même manière, il n'y a pas de variations de format. Une page web de présentation d'un article est utilisée de manière uniforme pour chaque ressource disponible. On note aussi qu'il est facile de définir la limite de la ressource, typiquement le document obtenu grâce au lien de téléchargement. Enfin, la licence est souvent définie au niveau de la plateforme qui ne contient que du contenu ouvert (c'est le cas d'arXiv par exemple).

Considérons maintenant les chaînes Youtube : en premier lieu, le niveau est rarement spécifié explicitement de manière uniforme. La vidéo en question peut tout aussi bien être une présentation lors d'une conférence universitaire de haut niveau qu'une présentation destinée à des élèves de maternelle. En second lieu, la licence se trouve ici au niveau de la ressource. En effet, l'ensemble de la plateforme n'est pas dédié à l'éducation. En troisième lieu, le type de ressource est clair ; nous avons affaire à des vidéos, mais la vidéo constitue-t-elle la limite de la ressource ? La réponse est : pas toujours, certaines vidéos font partie d'une série, elle-même composant un cours. Enfin, la finalité n'est pas non plus claire : nous pouvons avoir affaire à de la vulgarisation, des cours, des exemples corrigés

ou des restitutions de conférences. Détecter la finalité de manière automatique est une tâche difficile. Dans certains cas, une même chaîne peut d'ailleurs héberger des exemples de chacune des catégories.

Envisageons maintenant un cas plus complexe, le site d'un particulier. Celui-ci peut par exemple être un vulgarisateur/chercheur tenant un site de vulgarisation ou un professeur partageant librement ses ressources sur son site personnel. Dans ce cas, la limite de la ressource n'est pas toujours claire, la page web en elle-même est souvent la ressource et les liens qui la composent peuvent faire ou ne pas faire partie de la ressource. L'apposition de la licence est aussi variable : il est difficile de savoir où s'arrête et où commence son application. Le public cible a peu de chance d'être indiqué, il est souvent très variable. Par exemple, certaines informations peuvent correspondre uniquement à la classe à laquelle le professeur donne son cours actuellement (notes, listes d'émargement...) et ne constituent donc pas des ressources éducatives libres. La finalité de la ressource -même lorsqu'elle est claire pour un œil humain- peut être difficile à détecter de manière automatique, car non renseignée de manière uniforme. Il en va de même pour les liens entre les ressources pas toujours explicites. Quels exercices correspondent à quel cours ? Ou à quelle partie dans le cours ?, etc.

Un autre cas intéressant est celui des Learning Management System (LMS) ou Environnement Numérique d'apprentissage (ENA). Les ENA sont des logiciels qui permettent la gestion d'un parcours d'apprentissage. Parmi les plus connus, on peut citer Moodle qui possède plus de 50% de parts de marché en Europe, en Amérique du Sud et en Océanie [13]. Les ENA sont largement utilisées par les universités, les collèges, les lycées et les écoles ; ils sont capables de prendre en charge la partie administrative et logistique de la formation, notamment présentielle. Les ENA contiennent souvent des Environnements Numérique de Travail (ENT), dans lesquels les élèves peuvent retrouver, selon les pratiques des enseignants : des tests d'autoévaluations, des ressources pertinentes telles que les diaporamas du cours, des feuilles d'exercices, des annales, etc. Les ENA sont des outils majeurs qui font le lien entre l'enseignement présentiel et distanciel. Ils sont très largement implantés dans le domaine de l'éducation ; c'est donc en toute logique qu'ils constituent un site de dépôt fréquent des REL. De manière générale, on observe sur ces ENT les mêmes difficultés que sur les sites de particuliers : emplacement et champ d'application des licences, limite et pertinence des ressources, etc. Un problème supplémentaire et paradoxal est qu'il est souvent difficile d'accéder aux ressources. En effet, si le contenu est libre d'accès en théorie,

13. Source : https://eliterate.us/academic-lms-market-share-view-across-four-global-regions/

en pratique les ENA sont conçues pour permettre uniquement l'accès aux étudiants d'une formation spécifique. C'est un fait logique quand on pense que des informations sensibles peuvent être hébergées sur ces plateformes. Néanmoins, cela aboutit en pratique régulièrement à des ressources sensibles non-protégées et à des REL très difficilement accessibles. Par exemple, pour accéder un cours sur un Moodle, il peut parfois être nécessaire de se connecter à la plateforme avec des identifiants singuliers (ex : nom d'utilisateur=visiteur, mot de passe=visiteur). Ce type de pratique soulève une question quant à l'accessibilité réelle des ressources, et ce, d'autant plus que l'information est parfois difficile à trouver. De manière, plus générale, elles constituent un frein évident au partage.

Bien sûr, nous avons jusqu'ici traité des cas relativement simples où le créateur de la ressource est l'acteur direct de la publication. Mais le propre de l'éducation ouverte est le partage. Nombre d'institutions (université, association, organisme public...) ont pris le parti de créer des répertoires de liens, on peut citer par exemple le catalogue AUNEGe [14]. Dans ce cas, la diversité explose à nouveau, chaque lien pouvant pointer vers un acteur aux pratiques différentes. De plus, le risque de lien mort et de perte de licence augmente significativement. Le public cible est, lui aussi, variable et dépend de l'organisme sans forcément être précisé. Bien entendu, dans ce cas, toutes les finalités sont envisageables pour les ressources : exercices, cours, vulgarisation etc. sans forcément d'annotation. Une question se pose également quant à la détection des doublons, spécifiquement lorsque les ressources sont librement modifiables et ne font pas l'objet d'une citation de l'auteur originel.

Ces quelques cas d'exemples typiques montrent comment l'absence d'homogénéisation des pratiques, mais surtout la diversité des acteurs et des usages, rend difficile l'indexation des REL, et encore plus de manière automatique. Néanmoins, l'aspect multisite ne pose pas uniquement des problèmes au niveau de l'indexation.

Au niveau de la recommandation, X5GON a la volonté de fournir un système de recommandation de REL facilement implémentable sur chaque site le souhaitant. Cela s'inscrit dans une volonté du projet de connecter les sites entre eux et non de centraliser les utilisateurs. X5GON ambitionne de jouer un rôle de promoteur des REL permettant de conserver une architecture distribuée et non centralisée. En pratique, cela peut facilement se faire à partir de technologie d'injection de code dynamique. Cela ne pose donc pas de soucis au point de vue technologique. Néanmoins, cette variété de sites implique une variété des usages ainsi que des utilisateurs. Certains peuvent être des apprenants avec toute

14. Site AUNEGe : https://aunege.fr/ressources/ressources-pedagogiques/

la diversité possible, notamment en termes de niveau et d'objectif d'apprentissage, mais aussi des enseignants, voire des institutions. À ce niveau, la contrainte multisite recoupe la nécessité d'être capable de recommander pour un contexte d'apprentissage non formel, à la manière de ce que peuvent faire par exemple des sites de visionnage de vidéos tel que Youtube. Mais avec la contrainte supplémentaire de proposer de la recommandation à visée pédagogique, mais également de l'absence de métadonnées cohérentes et complètes sur l'ensemble des ressources (caractère non structuré des données).

0.5.3 Engagement en faveur de la multi-disciplinarité

La multi-disciplinarité implique une très large variété dans les thèmes et les domaines abordés dans les ressources. Mais aussi une possible variété des pratiques d'apprentissage et des attentes des apprenants dans les différentes disciplines. Par exemple, une pratique commune en informatique, y compris pour les formations en ligne, est d'aménager des moments de mise en application pratique au travers des exercices de programmation. Ces derniers correspondent souvent à des tâches simples pour lesquelles il est facile d'évaluer la qualité du travail de manière objective, et parfois même automatique. Dans d'autres disciplines, par exemple en gestion, ces mêmes moments d'application pratique correspondront à des études de cas qui ne peuvent être évaluées que par retour des pairs ou d'un enseignant. Un autre exemple typique pourrait être la sur-présence des contenus audio dans des domaines comme la musique ou les langues, en contraste avec leur faible présence dans d'autres domaines tels que la peinture ou les mathématiques.

Le vocabulaire employé peut lui aussi largement varier d'un domaine à l'autre. C'est un enjeu important pour les modèles de recommandation -souvent basés sur des modèles de langue- que de capturer la diversité des vocabulaires -en particulier les termes spécifiques qui s'avèrent souvent être d'une importance critique dans la perception du domaine. Il est par exemple difficile de s'initier à la théorie des ensembles sans comprendre le terme *endomorphisme*, ou à la chimie des matériaux sans comprendre le terme*polymère*. Dans bien des cas, un terme est à comprendre relativement à un domaine : on parle de désambiguïsation des synonymes. En guise d'illustration, le terme « terminal » revêt un sens bien différent dans le domaine de l'informatique et dans celui du médical.

Contrairement à de la recommandation dans un cadre général, dans celle à visée pédagogique, certains apprenants veulent apprendre des connaissances spécifiques, parfois très spécialisées, et non simplement visionner du contenu en lien avec leurs préférences habituelles. Pour cette raison, pouvoir repérer les concepts abordés dans chaque ressource

représente un enjeu important de la recommandation à visée pédagogique. Pour ce faire, la méthode la plus simple consiste à se baser sur de l'annotation humaine permettant l'apprentissage d'une méthode automatique. Bien sûr, recueillir cette expertise dans une large variété de domaines, sur un très grand nombre de ressources, avec potentiellement des contenus demandant un haut niveau d'expertise, constitue un verrou scientifique important. D'autant plus que la Recherche a montré que les représentations domaine spécifiques fonctionnaient mieux que les représentations généralistes quand il s'agissait d'exprimer des similarités sémantiques entre les documents. C'est en particulier le cas dans certains domaines non dotés où les ressources sont moins bien représentées par les modèles généralistes.

0.5.4 Engagement en faveur du multi-linguisme

La variété des langues pose un enjeu en termes de représentations des ressources. Tout d'abord, parce que les performances des algorithmes de représentations multilingues ainsi que ceux de traduction sont variables en fonction des langues. En particulier, les langues dites peu dotées -comprendre faiblement présentes dans les corpus- sont très mal représentées par les modèles actuels. À l'inverse, sur des couples de langues bien dotées, certains auteurs affirment que leurs modèles produisent des traductions comparables à celles des humains (Miró et al., 2018). Le projet X5GON s'inscrivant dans l'esprit des mouvements de libre accès, la possibilité pour un utilisateur de consulter du contenu en langue peu dotée dans la sienne, ainsi que celle de le faire dans une autre langue que la sienne - même si celle-ci est peu dotée -sont des enjeux majeurs. De plus, le contenu pédagogique engage souvent un vocabulaire plus spécifique. Pour cette raison, il est nécessaire d'avoir des modèles de langues spécifiques à ce domaine. Les deux aspects combinés (i) faible nombre de contenus et (ii) nécessité de contenu pédagogique dans la langue complexifient encore la question de l'équité de la représentation entre les langues.

Dans le projet, cette contrainte est en particulier gérée par le partenaire de l'Université de Valence. Il possède une forte expertise dans le domaine de la transcription et de la traduction. Ses travaux sont aujourd'hui l'état de l'art pour ces tâches sur du contenu pédagogique. En particulier, sur les langues bien dotées, la qualité de transcription ou traduction proposée ne sont pas discernables du véritable texte écrit par des humains (Miró et al., 2018; Popel et al., 2020). De manière générale, l'anglais est la langue source et cible qui obtient les meilleures performances (Miró et al., 2018). Cela s'explique par deux phénomènes : premièrement, (i) l'anglais est de loin la langue la plus représentée dans

les corpus. (ii) Deuxièmement, c'est la langue par défaut employée dans la recherche sur les modèles de transcription et de traduction. Les performances sur l'anglais font que dans bien des cas, il est préférable, pour traduire d'une langue A à une langue B, de l'utiliser comme langue pivot : cela est principalement vrai lorsque A et B sont des langues peu dotées. Pour cette raison, dans le projet, les partenaires de Valence fournissent un équivalent textuel (il peut s'agir d'une traduction ou d'une transcription) en anglais de chaque ressource disponible. Ce texte est ensuite utilisé par les différents modèles du projet comme représentation de la ressource.

0.5.5 Engagement en faveur du multiculturalisme

Le multiculturalisme comme la multi-disciplinarité impliquent une large variété dans les contenus et les pratiques d'apprentissage. Cette variété combinée à l'inégalité en termes de représentation des différentes cultures -tant au niveau des utilisateurs que des contenus -impose une nouvelle fois la nécessite d'obtenir un modèle capable de représenter cette diversité. Au-delà de cela, les pratiques d'apprentissage peuvent avoir une influence directe sur la recommandation. Par exemple, dans certains contextes, dû à l'absence d'une large couverture internet sur un territoire ou à un coût d'accès élevé, la pratique peut être de télécharger du contenu lorsque la connexion est disponible et de le consulter hors-ligne. Dans ce cas, il découle que les ressources contenant beaucoup de liens externes sont moins pertinentes que celles plus autonomes. Un autre cas peut concerner les cultures avec une tradition de passage de connaissance orale qui privilégieraient sans doute ce type de contenu. Un dernier exemple peut concerner l'apprentissage des langues qui variera forcément en fonction de la langue natale du locuteur : un Français a besoin d'apprendre l'alphabet cyrillique pour apprendre le russe, mais pas un Bulgare.

0.5.6 Contrainte large-échelle et démarrage à froid global

X5GON a une volonté claire d'indexer un maximum de ressources pédagogiques, nous l'avons mentionné en Section 0.2.4, le nombre de REL dépassent aujourd'hui les 9 millions avec un taux de croissance en augmentation constante. Dans ce contexte, le moteur de recommandation de X5GON doit être capable de répondre à des requêtes sur une très grande quantité de ressources. De plus, l'ambition d'X5GON étant de donner un accès à ces ressources pour le plus grand nombre, il semble pertinent de se focaliser également sur les méthodes permettant une mise à l'échelle. Ces deux raisons font que le moteur de

recommandation X5GON doit être conçu pour pouvoir théoriquement interagir avec des millions d'utilisateurs sur des millions de ressources. Il entre donc complètement dans la catégorie des moteurs de recommandation à large échelle. Cette contrainte s'avère limitante sur le type d'algorithme utilisable et pose également un ensemble de problématiques supplémentaires, notamment lorsqu'elle est combinée avec la contrainte de recommandation pour un contexte d'apprentissage non formel (Section 1.3.3).

Comme nous le verrons dans l'état de l'art (Section 1.3.3), un tel cadre confine généralement à l'apprentissage de modèle de recommandation (typiquement des modèles neuronaux profonds) sur la base des interactions passées des utilisateurs. Dans le cadre du projet, de telles interactions ne sont pas disponibles, car le système reste à développer (on parle de départ à froid global). De manière générale, l'absence de jeu de données, notamment pour la tâche de recommandation, est un problème largement connu dans la littérature (Drachsler et al., 2015).

0.5.7 Spécificité d'un contexte éducatif

Enfin, si le domaine de la recommandation a largement été étudié dans des contextes d'application diverses telles que l'e-commerce ou les plateformes de visionnage de vidéos, son utilisation dans un contexte pédagogique -en particulier pour une application à grande échelle, pour un contexte d'apprentissage non formel- constitue un verrou scientifique d'importance. Dans les applications commerciales, il est suffisant de recommander les ressources une par une à la volée. En reformulant, nous pouvons nous concentrer sur l'objectif à court terme, maximiser localement nos objectifs et procéder d'une ressource à l'autre. Dans notre cas, l'apprentissage n'a pas que des objectifs immédiats et se préoccupe davantage des objectifs à moyen ou long terme. Une personne qui souhaite apprendre la « physique quantique » aura probablement besoin de plusieurs ressources avec une difficulté croissante entre les ressources, mais elle devra également aborder des sujets proches tels que les statistiques afin de progresser en physique quantique. Un défi consiste donc à fournir à l'utilisateur une recommandation de parcours d'apprentissage à travers plusieurs ressources. De la même manière, dans les applications commerciales, l'objectif à maximiser découle directement du modèle commercial. Dans notre cas, l'objectif est d'offrir à l'utilisateur une expérience d'apprentissage satisfaisante : il peut devenir plus difficile de définir un objectif mathématique simple que nous voudrions maximiser. Les paramètres à maximiser peuvent être très différents d'un utilisateur à l'autre ; de fait, les caractéristiques intrinsèques de l'utilisateur se sont avérées être des facteurs très influents dans

la satisfaction et l'efficacité de l'apprentissage. Les caractéristiques démographiques (Guo and Reinecke, 2014), l'intention de l'utilisateur (Li et al., 2015; Seaton et al., 2014), ou simplement le style d'apprentissage sont en pratique des facteurs clé. Cet aspect sera discuté en profondeur en Section 1.4.1.

0.6 Problématique

L'ensemble des verrous scientifiques suscités ne seront pas traités dans ce document, mais ont largement été considérés dans d'autres travaux liés au groupe de projet X5GON. Nous allons dans notre cas nous intéresser à deux problématiques spécifiques que nous retrouvons en filigrane de ce document :

— Comment maximiser l'expérience d'apprentissage et en particulier définir un bon ordre d'apprentissage ?

— Comment capter et comprendre l'intention de l'utilisateur ?

Pour la première, nous proposons une contribution dans le Chapitre 4, pour la deuxième, nous le verrons dans le Chapitre 2, le manque de données utilisateurs sur des traces d'apprentissages complexifie la recherche sur le sujet. Nous proposons donc une contribution indirecte à cette question au Chapitre 3 en proposant une méthode pour publier des jeux de données de traces d'apprentissage à partir de celle recueillie dans le projet X5GON. De plus, nous fournissons dans le chapitre 2 une restitution des travaux réalisés dans le cadre du projet X5GON ainsi qu'un ensemble de retour d'expérience ayant pour but de faciliter la mise en œuvre de projet futur.

0.7 Organisation du document

Le document est organisé comme suit : tout d'abord, nous présenterons un état de l'art conjoint sur les problématiques de représentation des documents et de recommandation (Chapitre 1), deux sujets au cœur de notre problématique. Ensuite, nous nous intéresserons à la problématique spécifique des données et de leurs obtentions dans le cadre de la recommandation à visée pédagogique (Chapitre 2). Ce sera également l'occasion de discuter de nombreux retours d'expérience du projet X5GON et d'un ensemble de contributions techniques et scientifiques que nous avons réalisés au sein du projet. Nous proposerons ensuite deux contributions, l'une portant sur l'anonymisation de données séquentielles dans un objectif de publication de jeux de données de traces utilisateurs (Chapitre 3) ; l'autre

proposant une approche de recommandation adaptée à la tâche pédagogique (Chapitre 4).

Ce document a été rédigé à destination d'un lecteur ayant un niveau minimum de Master dans le domaine de l'informatique. Il traite en général de la question de la recommandation à visée pédagogique et aborde un ensemble de questions périphériques en lien avec cette problématique. L'angle pris dans le discours concerne le point de vue d'un informaticien sur la question. Ainsi, certaines questions d'importance sur le sujet comme la facilité d'adoption de tel algorithme ou la perception par les apprenants ne sont traités que par le média des contributions d'autres chercheurs. Bien entendu, la recommandation à visée pédagogique est un objectif largement multidisciplinaire, d'autres domaines de compétences tels que la sociologie ou la psychologie de l'apprentissage y ont un rôle central à jouer. Parce que le point de vue pris est celui de l'informatique, les contributions ou les restitutions mathématiques en lien avec les algorithmes utilisés ou les méthodes constituent logiquement le cœur du document. D'autres parties sont plus de l'ordre du retour d'expérience pratique quant à la mise en place de ces algorithmes, qui constituent naturellement un niveau de preuve moins élevé qu'une démonstration mathématique. Néanmoins, nous pensons qu'ils sont indispensables pour traiter la problématique. Ces retours d'expériences peuvent être de trois natures :

— être issus des conclusions du projet, X5GON dans ce cas, ils seront précédés dans la marge d'un encadré bleu ;
— être issus de recherches transverses ou provenant d'autres domaines. Dans ce cas, ils seront précédés d'un encadré vert.

Exemple de retour d'expériences provenant des conclusions du projet X5GON.

Exemple de retour d'expériences provenant de recherches transverses.

Enfin, le cadre de travail de la thèse recoupant le cadre du travail du projet X5GON, certaines des contributions présentées dans ce document ont été réalisées au sein du projet. Lorsque nous évoquerons le travail effectué dans le cadre du projet X5GON, nous choisirons une forme neutre pour présenter les travaux du consortium, et réserverons le « nous » à des travaux dans lesquels j'ai été directement associé et où mon apport a été substantiel.

Chapitre 1

État de l'art

Comme discuté dans l'introduction, la problématique principale de la thèse concerne la recommandation à visée éducative dans un contexte non formel et non structuré. Parmi les contraintes accompagnant la problématique, on relève la nécessité de proposer des modèles applicables à large échelle, ainsi que l'ensemble des contraintes liées au contexte non formel et non structuré, et en particulier celles concernant la pluralité de domaine et de format des ressources. Dans un tel contexte, le choix du projet X5gon a été d'utiliser le texte comme modèle de représentation universelle des ressources. Ce choix peut sembler restrictif dans le cadre de notre problématique multimodale : néanmoins, plusieurs arguments, comme la prédominance des méthodes utilisant des représentations textuelles dans le domaine de la recommandation et la qualité des systèmes de traduction/transcription et d'extraction de texte, en particulier sur les ressources éducatives (Miró et al., 2018), plaident en faveur de ce choix (Voir la discussion préalable en Section 0.5).

Une attention particulière sera portée à la présentation des algorithmes et des méthodes que nous emploierons dans la suite du document (Sections 1.1.1.3, 1.1.2.2, 1.2). Cet état de l'art a donc deux objectifs principaux : (i) fournir aux lecteurs un ensemble d'outils théoriques permettant de comprendre les méthodes et problématiques en lien avec la recommandation à large échelle, (ii) fournir un panorama des différentes approches permettant de traiter la recommandation à large échelle et les limitations de ces approches dans le cadre de l'application dans un contexte éducatif.

Pour cela, il sera détaillé sur deux axes. Dans un premier temps, nous nous intéresserons aux approches de représentation sémantique des documents, préalable indispensable aux approches textuelles, en suivant une présentation chronologique. Ensuite, nous nous intéresserons à la recommandation de contenu pédagogique à large échelle et en particulier aux méthodes d'apprentissage d'ordre (*learning to rank*). Les méthodes de représentation sémantique des documents cherchent à représenter les documents dans des espaces parfois tangibles (Rajaraman and Ullman, 2011), parfois non interprétables (Le and Mikolov, 2014) dans lesquels les documents ayant des contenus sémantiquement proches sont éga-

lement proches dans l'espace et à l'inverse les documents sémantiquement éloignés se retrouvent éloignés dans l'espace. Elles constituent donc un préalable intéressant à la recommandation de contenu. Les modèles de recommandation qui exploitent ces représentations dans le but de recommander des contenus pertinent au sens d'une métrique à maximiser représentent aujourd'hui l'état de l'art de la recommandation à large échelle. L'étude de la structure de ces systèmes ainsi que de leurs limitations dans le cadre de la recommandation à visée pédagogique constitueront donc le deuxième axe de cet état de l'art.

1.1 Représentation sémantique des documents

Dans cette section, nous nous intéresserons aux représentations sémantiques des documents. Au contraire des images ou du son qui ont des représentations informatiques vectorielles denses, le texte est encodé dans nos ordinateurs avec une représentation vectorielle éparse et on utilise typiquement une représentation basée sur l'encodage des caractères (ASCII, Unicode...). Cette représentation discrète et éparse a pour défaut principal de ne pas représenter les relations entre les documents. En effet, il n'y a pas de logique sémantique sur la numérotation des caractères ; de ce fait, la simple composition des lettres ne porte aucune signification inhérente aux mots qu'elles forment et a fortiori aux documents que les mots composent.

Pour cette raison, nous nous intéressons ici à des représentations tenant compte de la proximité sémantique. Dans de telles représentations, les documents sont usuellement des vecteurs dans un espace multidimensionnel. Plus particulièrement, nous nous intéressons au cas où ces représentations doivent être apprises de manière non supervisée, c'est-à-dire sans exemple de relation d'ordre, ou d'annotations particulières sur les interactions entre les documents. Cela s'oppose aux méthodes supervisées qui vont chercher à apprendre ces représentations en tirant profit d'exemples d'apprentissage obtenus sur une tâche spécifique, typiquement une classification en domaine (Harish et al., 2010). Le choix de s'intéresser en particulier aux méthodes non supervisées est motivé par la variété des domaines et la quantité de ressources que nous voulons traiter. En effet, plus la quantité de ressources est grande, plus il est difficile d'annoter les ressources,

Par souci de clarté, voici quelques notions et leurs notations que nous utiliserons dans la suite de cette section :

Document : Un document, que nous noterons $\mathcal{D}$, est un multiensemble de mots

notés w_i tel que $\mathcal{D} = \{w_1 \dots w_i \dots w_{|\mathcal{D}|}\}$. En cas d'ambiguïté, nous noterons $w_i^{(\mathcal{D})}$ le mot à la position i dans $\mathcal{D}$.

Corpus : Un corpus, que nous noterons $\mathcal{C}$, est un ensemble de documents $\mathcal{C} = \mathcal{D}_1 \dots \mathcal{D}_i \dots \mathcal{D}_{|\mathcal{C}|}$. $|\mathcal{C}|$ correspond au nombre de documents et $||\mathcal{C}||$ correspond au nombre total de mots dans le corpus.

Vocabulaire : Le vocabulaire que nous noterons V est l'ensemble de mots différents qui occurrent dans un ou plusieurs documents.

Matrice de représentation des documents : Nous noterons D la matrice de représentation des documents, cette matrice servira à représenter l'ensemble des documents du corpus, chaque document sera représenté par un vecteur de taille d_e, la taille du vecteur dépendant de la méthode de représentation choisie. D est de taille $|\mathcal{C}| \times d_e$ avec d_e la dimension des vecteurs de représentation. On remarque que la $i-$ème ligne de la matrice D_i correspond au vecteur de représentation du $i-$ème document $\mathcal{D}_i$. Pour simplifier les notations, nous pourrons utiliser l'abréviation $D_{\mathcal{D}}$ pour nous référer au vecteur de représentation document $\mathcal{D}$ sans avoir à faire référence explicitement à son index.

Matrice de représentation des mots : Nous noterons W la matrice de représentation des mots, cette matrice servira à représenter l'ensemble des mots du vocabulaire, chaque mot sera représenté par un vecteur de taille d_e, la taille du vecteur dépendant de la méthode de représentation choisie. W est de taille $|V| \times d_e$ avec d_e la dimension des vecteurs de représentation (nous ferons l'hypothèse de la même taille pour la représentation des documents et des mots, même si cela n'est pas obligatoire en pratique). Pour simplifier les notations, nous pourrons utiliser l'abréviation W_w pour nous référer au vecteur de représentation du mot w sans avoir à faire référence explicitement à son index.

1.1.1 Approches discrètes

Dans ce premier axe, nous allons nous concentrer sur les approches dites tangibles. Dans de telles approches, les vecteurs de représentation des documents sont plongés dans des espaces aux dimensions connues ou interprétables. On retrouve également ces approches mentionnées sous le terme d'approches discrètes ; cela est un héritage des approches historiques du domaine dîtes *sac de mots* dans lesquelles les dimensions de représentation étaient discrètes (Rajaraman and Ullman, 2011). De la même manière, on

retrouvera aussi le terme d'*approche éparse* : cela est dû au fait que l'interpretabilité des dimensions composant l'espace de représentation engendre souvent des vecteurs de représentation très creux (majoritairement composés de 0). Il existe nombre d'approches permettant de créer de tels vecteurs, mais deux types d'approches émergent dans le domaine. Dans le premier choisit, on utilise comme dimension de représentation des sous-ensembles du vocabulaire usuellement petits ou contenant des mots sémantiquement très liés : ce sont les approches sac de mots. Le deuxième type regroupe les approches qui s'intéressent à représenter les documents dans des espaces dans lesquels les dimensions sont des concepts le plus souvent extraits d'une ontologie (Baziz et al., 2004). Dans cet état de l'art, nous avons fait le choix de présenter brièvement ces deux types d'approches en détaillant pour chacun une méthode particulière. Ce choix d'une description sommaire est motivé par le fait que ce type d'approche engendre des vecteurs de très grandes dimensions qui, nous le verrons plus tard, sont difficilement combinables avec les méthodes d'apprentissage profond utilisées de manière prédominante dans la recommandation à large échelle.

1.1.1.1 Approches *one-hot*

Avant de s'intéresser en détail aux représentations discrètes cherchant à représenter les documents dans des espaces tenant compte de leurs proximités sémantiques, nous allons commencer par introduire l'approche *one-hot* qui n'est pas utilisée comme représentation sémantique en tant que telle, mais qui est souvent utilisée comme représentation d'entrée pour un modèle d'apprentissage de ces proximités sémantiques (Le and Mikolov, 2014; Peters et al., 2018). La méthode *one-hot* se base sur une représentation simplificatrice des documents, dans laquelle le corpus est une matrice identité carrée (même nombre de lignes et de colonnes) de dimensions correspondant au nombre de documents, $D \in |\mathcal{C}| \times |\mathcal{C}|$. Avec cette représentation, chaque document est représenté par un vecteur correspondant à une colonne de la matrice ($D_{\mathcal{D}}$) ; ainsi, toutes les dimensions de ces vecteurs de représentations sont à zéro sauf une qui est à un. Il est important que remarquer que les vecteurs de tous les documents sont alors orthogonaux entre eux ; cette propriété garantie une équidistance entre les documents lorsqu'on utilise des distances telles que la similarité cosinus ou la distance euclidienne. Cela reflète donc très mal leurs proximités sémantiques : en effet, deux documents ne variant que d'un mot sont aussi proches que deux documents n'ayant aucun mot en commun. Par contre, cette représentation ne fait pas d'hypothèse *a priori* sur la proximité des documents : en cela, c'est une représentation numérique intéressante du corpus, notamment pour un modèle d'apprentissage. Bien-sûr, ce type de représenta-

tions ne se limite pas aux ensembles de document, mais peut être utilisé pour tout type d'ensemble. Dans la suite, nous verrons son utilisation notamment pour représenter le vocabulaire (Section 1.2, 1.2 et 1.1.2.2), le corpus ou encore des classes (Section 1.3.2 et 1.3.1) dans le cas d'une classification.

1.1.1.2 Approches sac de mots

Historiquement, les premières approches de représentation sémantique de documents se sont intéressées à utiliser des mots ou des groupes de mots du vocabulaire comme dimensions pour les vecteurs de l'espace. C'est le cas des sacs de mots, dont l'une des plus récentes mentions connues se trouve dans l'ouvrage fondateur de 1954 *Distributionnal stucture* par Zellig S. Harris (Harris, 1954).

Cette représentation suit une approche simplificatrice qui consiste à représenter un document par l'histogramme des occurrences des mots le composant : pour un document donné, chaque mot se voit affecter le nombre de fois qu'il apparaît dans le document. Un document est donc représenté par un vecteur de la même taille que le vocabulaire.

Pour expliquer cela plus formellement, nous allons introduire ici le concept de *matrice de cooccurrences terme-document* souvent abrégé matrice terme-document. Une matrice terme-document décrit la fréquence des termes qui apparaissent dans une collection de documents. Dans une matrice terme-document, les colonnes correspondent aux documents de la collection et les lignes correspondent aux termes (usuellement les mots composant le vocabulaire).

Ainsi pour un corpus $\mathcal{C}$ composé de $|\mathcal{C}|$ documents $\mathcal{D}_1 \dots \mathcal{D}_i \dots \mathcal{D}_{\mathcal{C}}$, la matrice associée $A^{\mathcal{C}}$ est de taille $|\mathcal{C}| \times |V|$ avec $|V|$ la taille du vocabulaire issue de $\mathcal{C}$. Et $A^{(\mathcal{C})}_{i,j}$ correspond au nombre d'occurrences du mot V_j dans le document $\mathcal{D}_i$. Par souci de simplification, l'exposant $\mathcal{C}$ est souvent omis.

De cette matrice, nous pouvons aisément extraire le vecteur sac de mots associé à chaque document comme la colonne $A_{*,i}$ correspondante de la matrice. La Figure 1.1 propose un cas d'illustration de la méthode.

Différentes variantes du sac de mots ont été développées au fil du temps (Rajaraman and Ullman, 2011). Parmi elles, on peut citer sa version binaire où la matrice ne conserve plus le nombre d'occurrences du document, mais l'information de sa présence ou de son absence. D'autres variantes se sont intéressées à normaliser cette matrice afin d'obtenir de meilleures représentations des documents, parmi lesquelles on peut citer entre autre la normalisation PMI (point-wise mutual information) ou la normalisation *TF-IDF* que

$$\mathcal{C} = \begin{cases} D_1 = \text{"Le renard saute sur le dos du chien"} \\ D_2 = \text{"Le chien ne voit pas le renard"} \\ D_3 = \text{"Mon dos me fait un mal de chien"} \end{cases}$$

$$A^{\mathcal{C}} = \begin{array}{c|ccccccc} & \text{renard} & \text{saute} & \text{dos} & \text{chien} & \text{voit} & \text{fait} & \text{mal} \\ D_1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ D_2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ D_3 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

Vocabulaire = {renard, saute, dos, chien, voit, fait, mal}
Mots retirés = {le, sur, du, ne, pas, mon, me, un}

FIGURE 1.1 – Exemple d'une représentation sac de mots.

nous verrons toutes deux plus loin (Bouma, 2009; Luhn, 1957).

Enfin, le sac de mots est souvent vu comme un cas particulier d'un modèle de représentation général par fenêtrage (*w-shingling* dans la littérature) (Broder et al., 1997) ; au lieu de considérer un unique groupe de mots comme cela est fait dans le cas des sacs de mots. Il est possible de considérer des groupes de plusieurs mots ; de tels groupes sont appelés n-grammes. Par exemple, une représentation 2-grammes considérera comme dimension toutes les sous-séquences de deux mots présentes dans le corpus. Conceptuellement, nous pouvons considérer le modèle de sac de mots comme un cas particulier du modèle *n*-grammes, avec n=1. Certaines approches se sont également intéressées à combiner différentes dimensions de *n*-grammes pour construire le vocabulaire (Mikolov et al., 2013b).

1.1.1.2.1 TF-IDF

Le TF-IDF (de l'anglais *term frequency-inverse document frequency*) est une méthode de pondération de la matrice de terme-document souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Contrairement aux approches sac de mots sans pondération, cette méthode permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Intuitivement, un terme est d'autant plus significatif dans un document qu'il est spécifique à ce document. Cette mesure va donc s'intéresser à favoriser les termes présents du document et rares dans le

corpus, car ils sont caractéristiques du contenu du document.

Plus formellement,

$$TF - IDF_{i,j} = A_{i,j} \log \frac{||\mathcal{C}||}{|\{\mathcal{D}_j : w_i \in \mathcal{D}_j\}|}$$

avec $|\{\mathcal{D}_j : w_i \in \mathcal{D}_j\}|$: nombre de documents où le terme w_i apparaît (c'est-à-dire $A_{i,j} \neq 0$). On retrouve ici la fréquence d'apparition du mot dans le document $A_{i,j}$ qui est appelée fréquence brute du terme (term-frequency) et une fraction représentant la rareté du mot dans le corpus (inverse document-frequency). Comme dans le cas du sac de mots, plusieurs variantes sont possibles sur la manière de construire la matrice A. Parmi elles, on peut citer la variante Okapi BM25 encore très utilisé de nos jours en recherche d'information (Robertson and Sparck Jones, 1988).

1.1.1.3 Approches basées sur des ontologies

Les approches discrètes que nous avons vues jusqu'ici n'utilisent pas de base de connaissance extérieure, mais se limitent aux contenus textuels des ressources pour les représenter. Néanmoins, ce n'est pas un cas général dans la littérature ; de nombreuses approches se sont intéressées à représenter les documents en exploitant des connaissances, issues de bases de connaissances extérieures, spécialisées ou généralistes (Harish et al., 2010; Al-Aswadi et al., 2020). De telles bases de connaissances, lorsqu'elles sont organisées et structurées, sont regroupées sous le terme d'ontologie. Une ontologie a pour rôle de représenter un ensemble de concepts relatifs à un ou plusieurs domaines, ainsi que de rendre compte des relations entre ces concepts (Effingham, 2013). Ces approches sont particulièrement pertinentes lorsque le champ d'étude du corpus est bien délimité et que les ressources présentent des finalités similaires.

Dans notre cas, le corpus est mal délimité puisqu'il est susceptible d'évoluer constamment au fil du temps avec la probable avec l'arrivée de nouvelle REL. Cela implique également l'apparition d nouveau domaine de compétence ou de recherche dans l'intérieur du corpus. Par ailleurs, nombre de RELs sont directement issues du monde académique et s'adresse à des publics d'experts, ainsi, il est fréquent que certains concepts abordés soient nouveaux ou très spécifiques et donc potentiellement non ou mal renseignés dans de telles bases de connaissances. De manière plus générale, dans un cas les ressources sont non structurées et généralistes, il est souvent difficile d'obtenir de bonnes représentations sans intervention humaine dans le processus d'annotation des concepts, comme

souligné par Al-Aswadi et al. (2020). Cette annotation est rendue particulièrement complexe lorsque le nombre de ressources dépasse le million comme c'est notre cas et nécessite souvent le travail d'experts.

Pour ces raisons, nous avons choisi de ne pas nous focaliser sur les approches recourant à des ontologies. Néanmoins, nous allons présenter ici une de ces approches en guise d'illustration, car nous l'utiliserons par la suite (Sections 2.3.5.1 et 2.3.5.2).

Wikifier

Wikifier [1] est un service développé par (Brank et al., 2017) permettant d'annoter du contenu textuel avec des concepts Wikipédia relevant de son contenu ; ici chaque article de Wikipédia est un concept et les liens hypertextes entre les différents articles servent de relations de référencement entre ces concepts. Pour la suite de l'explication, nous prendrons garde à appeler une page Wikipédia un *concept* et le texte support de l'hyperlien de référencement entre deux pages une *mention*. Selon leur propre site, Wikipédia est un projet d'encyclopédie collective en ligne, universelle, multilingue, qui vise à offrir un contenu librement réutilisable, objectif et vérifiable, que chacun peut modifier et améliorer. Wikipédia comporte aujourd'hui plusieurs millions d'articles dans différentes langues sur une large variété de sujets, constamment mis à jour et améliorés par une communauté active. Le Wikipédia en langue anglaise comporte par exemple 6,342,268 articles et bénéficie d'une communauté active de plus de 121,664 membres [2]. L'extraction d'une représentation des documents à partir de l'ontologie Wikipédia fait l'objet d'un champ de recherche appelé *Wikification* : un bon point d'entrée pour ce domaine de recherche est Mihalcea and Csomai (2007).

L'algorithme de Wikifier cherche à exploiter la structure du graphe bipartite mention-concept dans le but d'annoter les concepts. Pour cela, il suit un cheminement en 5 étapes :

1. Détection des *mentions*,
2. Création du graphe bipartite *mention-concept*,
3. Enrichissement du graphe avec les liens inter-concepts,
4. Application du PageRank,
5. Extraction des concepts majeurs.

1. Le service est disponible à l'adresse suivante : `wikifier.org`
2. Source : `https://en.wikipedia.org/wiki/Special:Statistics` en date du 23 juillet 2021

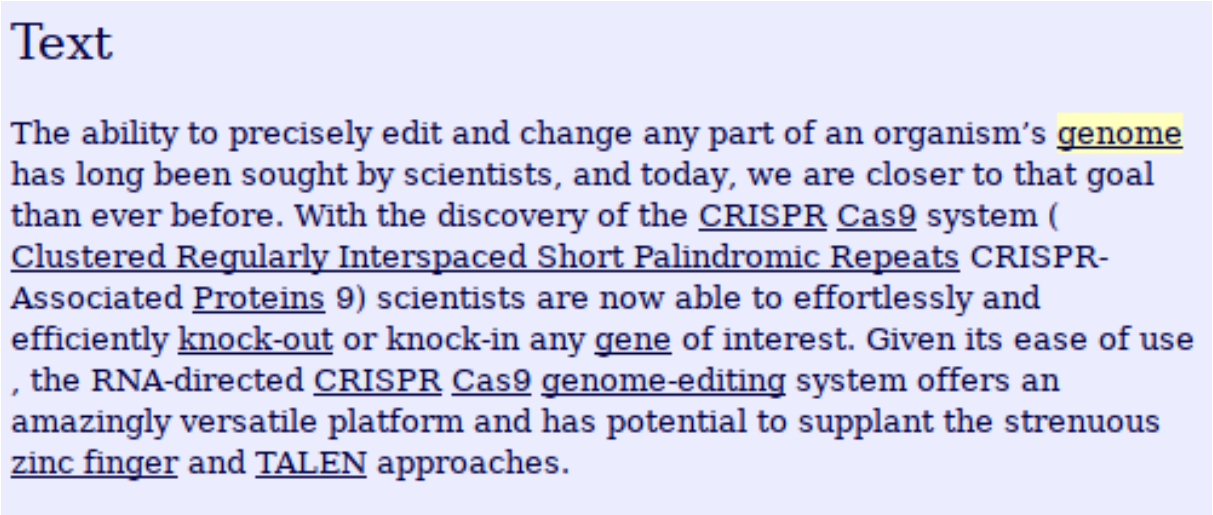


FIGURE 1.2 – Exemple de détection de mention

Détection des mentions et création du graphe La première étape consiste à détecter toutes les sous-chaînes du texte de la ressource ayant également été utilisées comme mention sur Wikipédia ; cela peut se faire en pratique en utilisant un algorithme de recherche de séquence. La figure 1.2 illustre un cas d'exemple de cette détection.

La deuxième étape, elle, s'emploie à reconstruire le graphe *mention-concept*. Pour cela, il faut créer un graphe biparti dans lequel l'ensemble des mentions détectées constitue un premier ensemble de nœuds et l'ensemble des concepts découlant de ces mentions un deuxième ensemble. Une mention a et un concept c sont reliés par un arc orienté $a \to c$ si et seulement si la mention a a été utilisée pour référencer la page de c sur Wikipédia. Dû à la nature de Wikipédia, une même mention peut avoir été utilisée pour référencer différents concepts ; cela ce traduit directement dans le graphe sur lequel un même mention peut référer différents nœuds concepts. Pour tenir compte de la prédominance de chaque concept sur une mention, chaque arc est pondéré par une valeur :

$$\mathbb{P}(a \to c) = \frac{\text{nombre d'hyperliens Wikipedia ayant } a \text{ pour mention et } c \text{ pour cible}}{\text{nombre d'hyperliens Wikipédia ayant pour mention } a}$$

Ajout de lien inter-concepts La troisième étape consiste à enrichir ce graphe en ajoutant des liens concept-concept entre les concepts proches sémantiquement, ici, proche doit se comprendre dans le sens où si l'un d'entre eux est pertinent pour un document d'entrée donné, l'autre est également plus susceptible d'être pertinent pour ce document. Pour cela, pour chaque paire de concepts c et c' dans le graphe, une mesure de similarité

sémantique est calculée de la manière suivante :

$$SR(c, c') = 1 - \frac{log(max(|L_{*,c}|, |L_{*,c'}|)) - log(|L_{*,c} \cap L_{*,c'}|)}{log(N) - log(min(|L_{*,c}|, |L_{*,c'}|))}$$

Avec :

$L_{*,c}$ **:** l'ensemble des hyperliens ayant c comme page de concept cible

N **:** Le nombre de concepts (i.e : page) Wikipédia.

Finalement, un lien $c \rightarrow c'$ est créé pour chaque paire de concepts ayant une $SR(c, c')$ positive, chacun de ces liens est pondéré de la façon suivante :

$$P(c \rightarrow c') = \frac{SR(c, c')}{\sum_{c''} SR(c, c'')}$$

Pour prendre du recul sur cette étape, on peut remarquer que :

$$SR(c, c') > 0 \iff \frac{min(|L_{*,c}|, |L_{*,c'}|)}{N} < \frac{|L_{*,c} \cap L_{*,c'}|}{max(|L_{*,c}|, |L_{*,c'}|)}$$

Avec :

$\frac{|L_{*,c'}|}{N}$ **:** la proportion de concepts Wikipédia qui ont une référence vers c'.

$\frac{|L_{*,c} \cap L_{*,c'}|}{|L_{*,c}|}$ **:** la proportion de concepts référençant c et c' parmi les pages référençant c.

Dans l'étape suivante, ce graphe est ensuite utilisé comme base de calcul d'un vecteur de scores de PageRank pour chaque sommet. Ce calcul est effectué en utilisant l'approche itérative habituelle (c.f algorithme 1) où, à chaque itération, chaque nœud distribue son score de PageRank à ses successeurs immédiats, proportionnellement aux probabilités de transition. Du point de vue des concepts, cette similarité permet de mieux considérer les concepts sémantiquement proches, mais qui serait référencé par des mentions différentes, c'est le cas typique des concepts spécialisés qui ont tendance à avoir des courtes descriptions comprenant peu de mentions. Cela s'explique par les règles qui régissent la rédaction des articles de manière collaborative sur Wikipédia. En pratique, la tâche de rédaction du corps de l'article est réalisée par un nombre relativement restreint de contributeur. Sur cette base, les autres contributeurs peuvent proposer des améliorations tant sur le fond que sur la forme visant à en améliorer la qualité. Ces améliorations peuvent dans certains cas ne pas nécessiter une expertise profonde et permettent à chacun de contribuer en

fonction de ces compétences et de son expertise. Ces améliorations sont regroupées sous forme de tâche, chaque contributeur peut signaler une tâche à réaliser (par exemple ajouter un lien dans cette phrase) ou réaliser une tâche disponible. Beaucoup de tâches sont en lien avec l'édition telle que la correction de fautes d'orthographe, de grammaire, l'ajout de sources ou l'ajout de liens internes. Elles nécessitent donc une expertise moindre sur l'article en question. L'ajout de liens internes est une de ces tâches, elle permet d'avoir une architecture, des liens internes fréquemment améliorés est mis à jour. Évidemment, plus un article est vieux ou populaire (au sens du nombre de lectures et de l'intérêt des contributeurs) plus le nombre de contributions ayant participé à l'améliorer est important, ainsi ces articles bénéficient souvent d'une meilleure représentation dans les liens internes. Du point de vue de notre ontologie, cela signifie qu'un concept ajouté à l'encyclopédie récemment à moins de chances d'avoir des liens internes pertinents qu'un concept ancien. Il en va de même sur les concepts très spécialisés qui sont naturellement moins visibles et donc moins ciblés par les contributeurs et où même l'ajout de liens nécessite une expertise. De cet état de fait, on déduit que la similarité inter-concepts vient contrebalancer ce biais en favorisant les concepts nouveaux ou spécialisés.

PageRank Initialement introduit en (Brin and Page, 1998) comme une méthode de mesure de l'importance des pages web, PageRank est un algorithme d'analyse de liens qui attribue une pondération numérique à chaque élément d'un ensemble de documents hyperliés, tel que le World Wide Web, dans le but de « mesurer » son importance relative au sein de l'ensemble.

Entrées: $\mathcal{V}$: liste de noeuds, d : facteur de propagation, ϵ : critère de convergence
Résultat: Pagerank pour chacun des noeuds
$t \leftarrow 0$
pour chaque $n \in \mathcal{V}$ **faire**
  $PR(n,t) \leftarrow \frac{1}{|\mathcal{V}|}$
fin
pour $\sum_{n\in\mathcal{V}} ||PR(n,t) - PR(n,t-1)|| < \epsilon$ **faire**
  $PR(n,t) \leftarrow \frac{1-d}{|\mathcal{V}|} + d\sum_{n'\in In(n)} \frac{PR(n',t)}{|Out(n')|}$
  `/* Avec In(n) l'ensemble des noeuds prédécesseurs de n et Out(n)`
  `l'ensemble des successeurs de n */`
  $t \leftarrow t+1$
fin
retourner *PR(*, t)*

Algorithme 1: Algorithme du PageRank

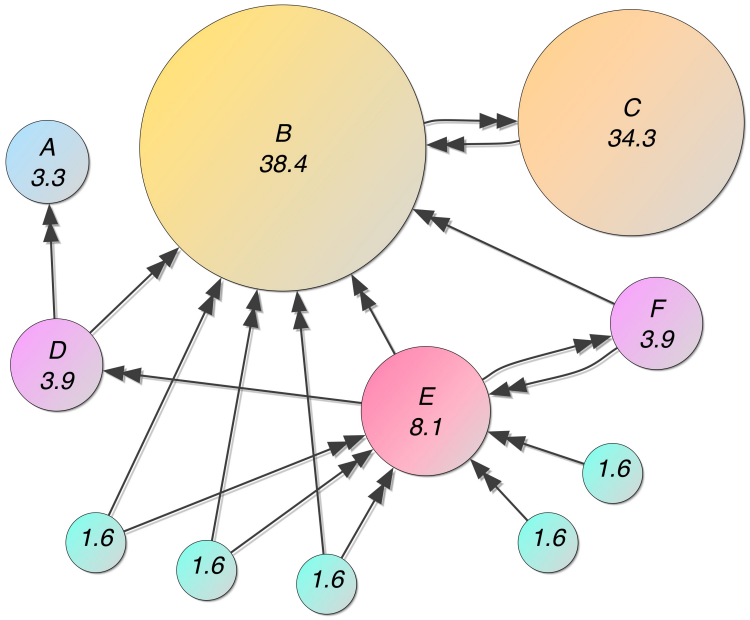


FIGURE 1.3 – Illustration de fonctionnement de l'algorithme de PageRank (Source de l'image : Wikipédia)

L'algorithme peut être appliqué à toute collection d'entités comportant des citations et des références réciproques. Une généralisation de PageRank pour le cas du classement de deux groupes d'objets en interaction a été décrite par (Daugulis, 2011) permettant d'appliquer les algorithmes de PageRank aux graphes bipartites (comme celui qui nous intéresse).

La figure 1.3 illustre le principe de fonctionnement de l'algorithme de PageRank. L'idée générale est celle d'une propagation de réputation : plus un nœud est important (au sens de son degré entrant) plus les références (liens sortants) qu'il possède sont récompensées par l'algorithme. Dans le contexte de Wikifier, l'algorithme de PageRank revient à itérer la formule suivante jusqu'à convergence pour tout nœud n :

$$PR(n, t+1) = dPR(n, 0) + (1-d) \sum_{n'} PR(n', t) P(n' \to n)$$

La formule à itérer suit donc en tous points celle d'un algorithme standard de PageRank comme celui présenté dans l'algorithme 1. Néanmoins, la subtilité ici permettant d'adapter l'algorithme au caractère bipartite du graphe réside dans la valeur d'initialisa-

tion $PR(n, 0)$.

$$PR(n,0) = \begin{cases} 0 & si\ n\ est\ un\ noeud\ concept \\ z\frac{L_{n,*}}{L_n} & si\ n\ est\ un\ noeud\ mention \end{cases}$$

z **:** facteur de normalisation qui assure $\sum_n PR(n, 0) = 1$

L_n **:** le nombre de pages Wikipédia contenant le texte n possiblement sans être *mention*. Défini uniquement pour les nœuds mention.

Ici, la méthode d'initialisation permet de contrebalancer le fait que dans notre graphe les nœuds *mention* n'ont pas de lien entrant, mais aussi de favoriser la convergence de l'algorithme. Pour voir cela plus en détail, il est nécessaire de remarquer que les nœuds mention n'ont par construction pas de lien entrant. Ainsi, il est indispensable de leur attribuer une valeur initiale de PageRank sans quoi ils seraient ignorés dans le processus. De plus, cela permet de mieux représenter des concepts souvent mentionnés avec un vocabulaire très spécifique et donc (à cause d'une mesure SR faible) potentiellement ayant des petits voisins en termes de PageRank, car très spécialisés eux aussi.

Extraction des concepts Le vecteur de PageRank obtenu précédemment permet alors de représenter le document. Dans leur implémentation, les auteurs obtiennent de meilleurs résultats sur les tâches d'annotations par concept (sur le jeu de données AIDA (Hoffart et al., 2011)) en conservant le nombre minimum de concepts couvrant 80% de la somme des PageRank.

Nous avons détaillé plus haut la méthode Wikifier en illustration des approches basées ontologie, le choix de détaillé cette méthode en particulier est motivé par le fait que nous détaillerons une application cette méthode plus tard dans le contexte du projet X5GON (Sections 2.3.5.1 et 2.3.5.2). Néanmoins, même si la littérature regorge d'approches, nous avons fait le choix dans cet état de l'art de ne pas détailler l'ensemble des contributions dans le domaine, le lecteur peut avantageusement se référer à la revue de littérature de (Szymański and Naruszewicz, 2019) pour les approches utilisant l'ontologie de Wikipédia. Une autre ontologie généraliste fréquemment utilisée dans la littérature est WordNet (Baziz et al., 2004).

Conclusion générale sur les approches discrètes Les approches discrètes présentent l'avantage de l'interprétabilité, mais engendrent des vecteurs de très grande dimension. En effet, la taille du vecteur est souvent celle du vocabulaire ou de l'ontologie.

Cette grande dimensionnalité combinée au caractère épars des vecteurs se combine mal avec les méthodes d'apprentissage profond qui sont utilisées de manière prédominante pour des tâches de plus haut niveau (recommandation entre autres). Pour cela des travaux se sont intéressés à représenter les documents dans des espaces denses de dimension ajustable, on retrouve ces approches dans la littérature sous les termes génériques de *représentation continue* ou *représentation dense* : nous présenterons ces approches dans la prochaine section.

1.1.2 Approches continues

Les approches discrètes vues précédemment ont l'avantage de représenter les documents dans des espaces dans lesquels les dimensions de représentation sont facilement interprétables (mots, groupe de mots, concepts). Elles engendrent par contre une très grande dimensionnalité des vecteurs et des représentations très éparses. Cela est problématique, car les données de plus petite dimension peuvent être traitées plus rapidement, notamment par les algorithmes d'apprentissage automatique. Ce phénomène est connu sous le terme de *fléau de la dimensionnalité* : il désigne l'ensemble des phénomènes qui ont lieu lorsque l'on cherche à analyser ou organiser des données dans des espaces de grande dimension alors qu'ils n'ont pas lieu dans des espaces de dimension moindre (Bellman, 1966). En effet, l'augmentation de la dimension implique une croissance combinatoire du nombre de configurations possibles pour les vecteurs et cela complexifie toutes les approches d'échantillonnage, d'apprentissage ou d'optimisation. De plus, l'augmentation de la dimensionnalité augmente mécaniquement la distance entre les éléments de l'espace. De manière plus problématique, cela lisse également la distribution des distances : en guise d'illustration du phénomène, on peut remarquer que dans un monde 2D la probabilité de former un triangle équilatéral ou presque avec trois points aléatoires est nulle ; par contre, dans un espace à 1000D elle est beaucoup plus élevée. D'un point de vue de l'apprentissage automatique, cela signifie que le nombre d'exemples nécessaire pour représenter l'espace croît lui aussi de manière combinatoire avec sa taille, mais aussi qu'il est beaucoup plus complexe d'apprendre des régularités dans des espaces de grandes dimensions. Si avoir une dimensionnalité trop grande n'est donc pas souhaitable, à l'inverse un nombre trop faible de traits a tendance à réduire les performances des modèles : on parle dans la littérature de *phénomène de Hughes* (Hughes, 1968). Comme illustré sur la Figure 1.4 il y a donc un point d'équilibre à trouver entre taille de l'espace et la qualité de la représentation.

Une des solutions consiste à remplacer les données originales par des données dans

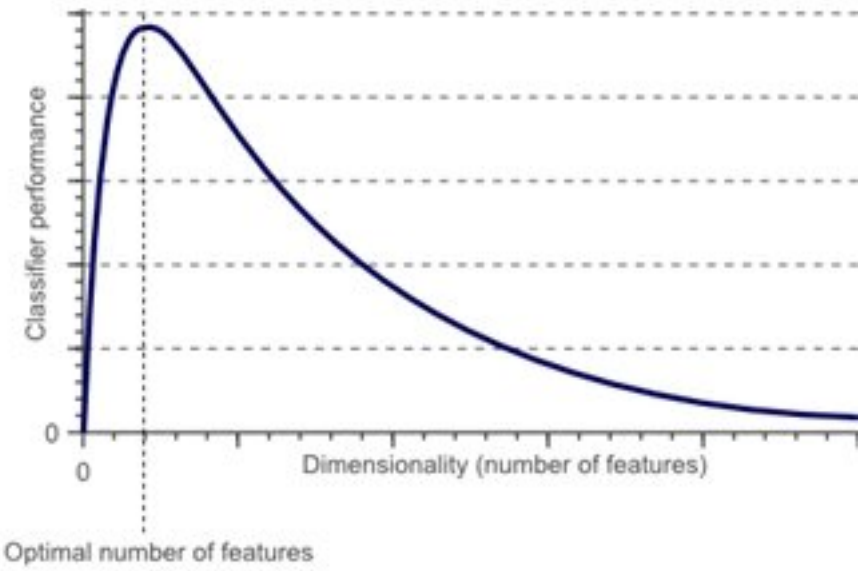


FIGURE 1.4 – Performance d'un algorithme de classification en fonction de la taille des données en entrée (phénomène de Hughes).

un espace de plus petite dimensionnalité. Dans cette sous-section, nous allons nous intéresser aux approches dites continues qui font le choix de représenter les documents dans des espaces denses de petite dimensionnalité, ces approches ont souvent la particularité de pouvoir contrôler la dimensionnalité souhaitée pour les vecteurs. Les vecteurs de représentations denses qui vivent dans ces espaces sont appelés des *plongements lexicaux*, l'attribut « lexicaux » est souvent omis et l'on parle alors simplement de plongements.

Dans cette sous-section, nous commencerons par présenter les approches reposant sur la factorisation de la matrice terme-document qui constituent des méthodes de références dans le domaine, puis nous aborderons les approches s'intéressant à exploiter les représentations vectorielles des mots. Enfin, nous finirons avec les approches basées sur l'apprentissage profond, qui s'emploient à exploiter les représentations internes apprises par de gros modèles neuronaux de langage.

1.1.2.1 Méthodes par réduction de matrice terme-document

L'idée générale des méthodes par réduction de dimensionnalité est de réduire la dimension de la matrice terme-document : pour ce faire, on utilise une approximation de rang inférieur de la matrice (Van Der Maaten et al., 2009). L'idée sous-jacente revient à

compresser l'information de l'espace disparate engendré par la matrice en un espace dense. Bien sûr, cette matrice a pour rôle de capturer le mieux possible les caractéristiques de la matrice de départ, même si en pratique l'approximation n'est jamais parfaite. Cette approximation possède plusieurs avantages :

— Grâce à son rang, la matrice approximée engendre des vecteurs de plus faible dimension pour représenter les documents ;
— La matrice originale terme-document est présumée bruitée : par exemple, les instances anecdotiques des termes doivent être éliminées. De ce point de vue, la matrice approchée est interprétée comme une matrice débruitée (une meilleure matrice que l'originale). Cela vient du fait que le modèle d'acquisition des données textuelle peut être modélisé comme la distribution réelle des données (en particulier les cooccurrences entre mots) plus un bruit additif. Ce bruit est classiquement modélisé par une distribution Gaussienne multivariée dans laquelle les dimensions sont indépendantes et identiquement distribuées sur chaque cooccurrence. Il est le fruit de variations statistiques dépendantes de la taille de l'échantillon, en particulier, il est de moyenne nulle, ainsi, pour un nombre infini de données, la distribution empirique correspond à la distribution réelle. La méthode de réduction de matrice ne conservant que les dimensions les plus pertinentes sur la base des régularités statistiques observées dans la distribution est donc susceptible de supprimer ce bruit.
— La matrice terme-document originale est éparse. En d'autres termes, la matrice originale ne répertorie que les mots effectivement présents dans chaque document, alors que nous pourrions être intéressés par tous les mots liés à chaque document, généralement un ensemble beaucoup plus large en raison de la synonymie.

Une des méthodes les plus utilisées dans le domaine de la représentation de documents est la décomposition en valeurs singulières (SVD pour *Singular Value Decomposition* en anglais).

La SVD est un procédé d'algèbre linéaire permettant la factorisation d'une matrice rectangulaire (Banerjee and Roy, 2014). La SVD opère une projection des éléments de la matrice selon les axes qui capturent le mieux la variance des données (composantes principales ou vecteurs singuliers). C'est une généralisation de la décomposition en valeurs propres (ACP *analyse en composantes principales*, voir figure 1.6) sur des matrices non-carrées. En sélectionnant uniquement les premiers vecteurs singuliers, on peut projeter notre modèle originel dans l'espace engendré par ces vecteurs et construire alors un modèle

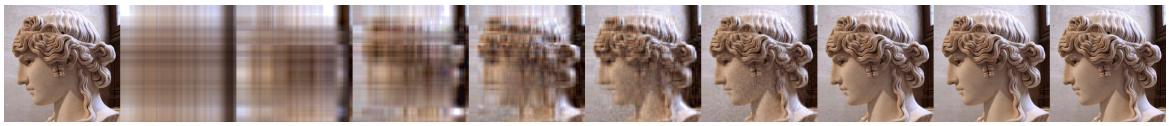


FIGURE 1.5 – Approximations successives d'une image, avec 1, 2, 4, 8, 16, 32, 64, 128, puis toutes les valeurs singulières à partir de la matrice de luminosité des pixels. L'image originale est représentée à gauche. Source : Wikipedia.

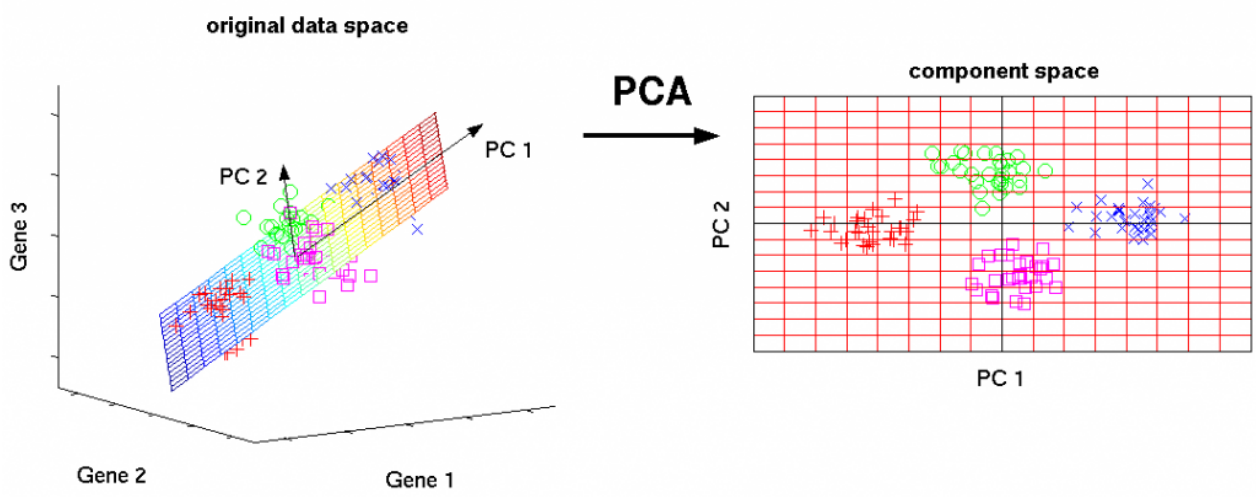


FIGURE 1.6 – Intuition de la SVD par une ACP. Source : http : //www.nlpca.org/ by Matthias Scholz.

simplifié, empirique, décrivant les données. On parle alors de SVD tronquée (*Truncated SVD*). On associe la norme de chaque *vecteur singulier* (appelée *valeur singulière*) à l'importance de la dimension engendrée par le vecteur dans la variance des données. Cette méthode est d'autant plus pertinente lorsque la distribution des valeurs singulières suit une décroissance exponentielle.

La première utilisation reconnue de la SVD dans le cadre de l'analyse du langage naturel est attribuée à Deerwester et al. (1990), et consiste en la factorisation de la matrice de cooccurrences termes-document, on parle de LSA pour Latent Semantic Analysis. Lorsqu'elle est appliquée sur une matrice de cooccurrence termes-document, la SVD permet d'obtenir des projections qui peuvent être utilisées comme vecteur de représentation dense des documents.

La SVD tronquée au rang K possède la propriété d'être la meilleure approximation de la matrice originale de rang K au sens de la norme euclidienne L2. À cause de cela, cette méthode fait implicitement l'hypothèse d'une distribution jointe *a priori* gaussienne

sur les mots et les documents. Cette hypothèse contredit les observations empiriques qui tendent à constater une distribution de Poisson (Jansche, 2003).

Pour cette raison, des méthodes probabilistes ont été utilisées dans le but d'obtenir de « meilleures » réduction de la matrice terme-document : on parle de pLSA pour *probabilistic LSA*.

Parmi elles, on peut citer la méthode LDA (Blei et al., 2003) (de l'anglais *Latent Dirichlet Allocation*) qui est essentiellement une formulation bayésienne de pLSA assumant une distribution *a priori* uniforme suivant une loi de Dirichlet. Cette loi est choisie, car elle peut être vue comme la généralisation la plus directe d'une loi de Poisson (discrète) dans un cas continu.

1.1.2.2 Méthodes par combinaison de plongements de mots

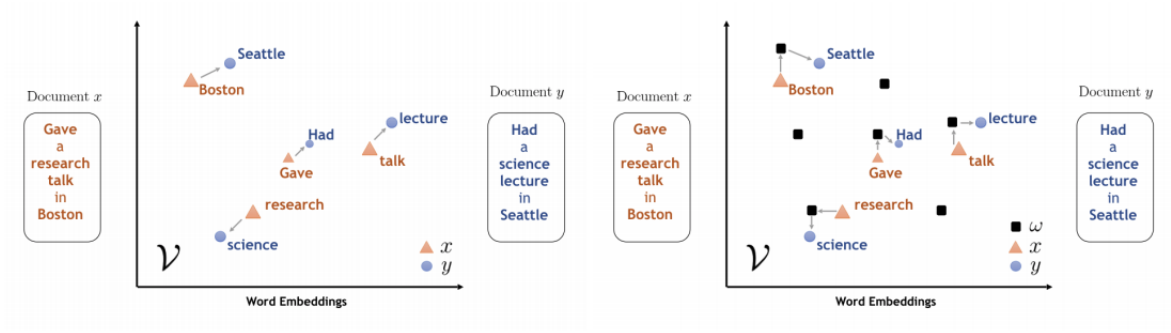
Une autre possibilité envisagée pour représenter les documents est d'exploiter les représentations de mots. En effet, à la manière de ce que l'on a pu observer pour les documents, certaines méthodes s'emploient à exprimer les mots par des vecteurs continus dans des espaces capturant leur similarité sémantique. Dans le cas général, ces méthodes sont souvent utilisées sur de grands corpus de textes généralistes permettant de capturer la plus grande variété possible. Ces modèles reposent sur deux idées sous-jacentes :

— "Linguistic items with similar distributions have similar meanings" (Harris, 1954).

— "You shall know a word by the company it keeps " (Firth, 1957).

La première idée met en avant que deux mots qui occurrent souvent avec les mêmes contextes sont de sens similaires. La deuxième avance que le contexte récurrent d'un mot donne une information sémantique forte sur le sens que l'on peut attribuer à ce mot. Certaines approches se basent sur ces constats pour interpréter la sémantique d'un mot cible à partir de la moyenne, éventuellement pondérée par la distance au mot cible dans le contexte, des mots présents dans le contexte récurrent du mot ciblent, on parle ici de moyenne des mots au sens de moyenne des vecteurs de sémantique les représentant.

Parmi ces méthodes, on cite souvent GloVe (Pennington et al., 2014), word2Vec (Mikolov et al., 2013a), FastText (Bojanowski et al., 2017), SVD (Levy et al., 2015). De manière intéressante, ces premières méthodes sont basées explicitement ou indirectement (dans le cas de word2Vec (Levy and Goldberg, 2014)) sur la *matrice de cooccurrences termes-termes* qui est une matrice similaire à la *matrice de cooccurrences termes-document*. Elle



(a) WMD (b) WME

FIGURE 1.7 – (a) WMD mesure la distance entre deux documents x et y, tandis que (b) WME approxime un noyau dérivé de WMD avec un ensemble de documents aléatoires. Source de l'image : https://towardsdatascience.com/document-embedding-techniques-fed3e7a6a25d#a7aa

se construit en comptabilisant dans le corpus le nombre de cooccurrences de chaque paire de termes dans une taille de contexte donné. En particulier, SVD consiste en une décomposition en valeurs singulières tronquées de cette matrice de cooccurrences dans la même logique que celle de LSA sur la matrice terme-document. Plus récemment, des méthodes réutilisant directement les représentations internes apprises par des modèles de langage basés sur des architectures neuronales ont également été développées. En particulier, (Peters et al., 2018) s'appuie sur l'architecture ELMo et (Devlin et al., 2019) s'appuie sur une architecture Transformer ; nous détaillerons ces deux architectures en section 1.2.

À partir de ces représentations des mots, plusieurs modèles ont été développés pour créer des représentations pour les documents.

En notant W_w le vecteur correspondant au mot w et $D_\mathcal{D}$ le vecteur correspondant au document $\mathcal{D}$, les méthodes les plus simples initialement utilisées opèrent simplement une agrégation sur l'ensemble des mots dans chaque document, les opérateurs d'agrégations les plus utilisés dans la littérature sont la somme $D_i = \sum_{w \in \mathcal{D}_i} W_w$ et la moyenne $D_i = \frac{\sum_{w \in \mathcal{D}_i} W_w}{|\mathcal{D}|}$ (Kusner et al., 2015). D'autres méthodes se sont intéressées à mieux utiliser ces représentations pré-apprises. *Word Mover's Embedding* (WME) (Wu et al., 2018) est une méthode d'apprentissage de représentation de document exploitant la mesure Word Mover's Distance (WMD) introduite par IBM (Kusner et al., 2015). WMD s'exprime comme la distance minimale dans l'espace des plongements que les mots d'un document doivent parcourir pour atteindre les mots de l'autre document.

Des approches plus sophistiquées se sont concentrées sur l'apprentissage conjoint de

représentation pour les mots et les documents (Kenter et al., 2016; Hill et al., 2016; Gupta et al., 2019; Le and Mikolov, 2014). Kenter et al. (2016) utilisent une architecture neuronale siamoise pour apprendre les représentations des mots directement depuis une tâche de prédiction de contexte au niveau des phrases. Pour Gupta et al. (2019), les représentations de mots sont apprises conjointement avec des représentations de n-grammes (plusieurs mots). D'autres approches se sont même intéressées à combiner les approches d'apprentissage conjoint avec des vecteurs de mots pré-appris (Chen, 2017). Pour aller plus loin, l'analyse comparative de Hill et al. (2016) fournit une description détaillée et éclairante des approches.

Parmi les approches conjointes, nous allons présenter l'une des plus efficaces et couramment utilisées : *Doc2Vec* (Le and Mikolov, 2014), Doc2Vec est une extension de Word2Vec. Dans le prochain paragraphe, nous présenterons cette méthode.

Le modèle Doc2Vec

Commençons par une brève introduction à *Word2Vec*, Word2Vec est une méthode basée sur un réseau de neurones artificiels à deux couches entraîné pour reconstruire le contexte linguistique des mots. Ici, on appelle contexte d'un mot, l'ensemble des mots qui entourent un mot dans une fenêtre de taille donnée. Cette méthode se décline selon deux architectures, la première, *Skip-gram*, a pour but de prédire un contexte à partir d'un mot cible ; la seconde, *CBOW* a pour but de prédire un mot cible à partir d'un contexte. Les modèles sont entraînés en utilisant les phrases du corpus comme exemple pour l'apprentissage. La taille m du contexte considéré est un méta-paramètre de ces modèles. Dans la littérature, les meilleurs résultats sont obtenus pour des valeurs proches de $m = 10$. Selon leurs auteurs, l'architecture Skip-gram est plus performante que CBOW sur de petits corpus et permet d'obtenir une meilleure représentation des mots peu fréquents. L'architecture CBOW est plus rapide à entraîner, et plus efficace sur les mots fréquents (Mikolov et al., 2013a).

Doc2Vec reprend le modèle proposé par Word2Vec mais cherche en plus à réaliser cette prédiction contextuellement au document. Comme Word2Vec, Doc2Vec se décline selon deux architectures. L'une, *Distributed Bag of Words version of Paragraph Vector* (Vecteur de Paragraphe en sac de mots distribué) abrégé en PV-DBOW est directement inspirée de Skip-gram et a pour but de prédire le document à partir d'un contexte donné. L'autre *Distributed Memory version of Paragraph Vector* (vecteur de paragraphe en mémoire distribué) abrégé en PV-DM est inspirée de CBOW et a pour but de prédire un

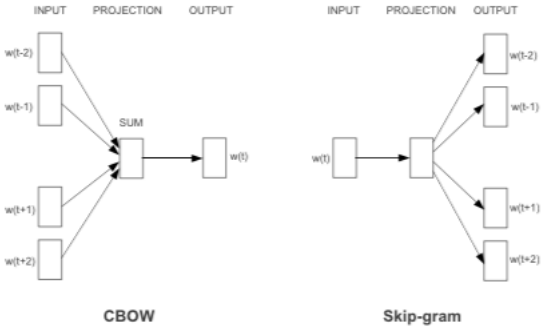


FIGURE 1.8 – Architecture des modèles de CBOW et Skip-gram dans Word2Vec, issue de Mikolov et al. (Mikolov et al., 2013a)

mot cible à partir d'un mot et d'un document donné (Mikolov et al., 2013a).

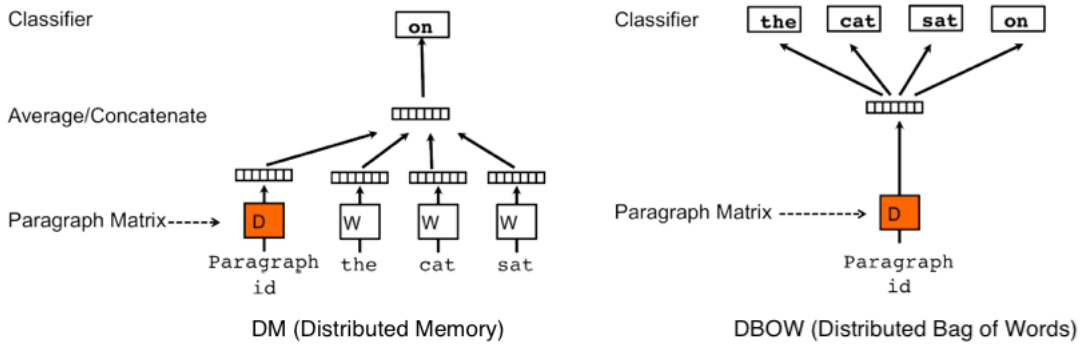


FIGURE 1.9 – Architecture des modèles de PV-DBOW et PV-DM dans Doc2Vec, issu de Mikolov et al. (Le and Mikolov, 2014)

Dans le modèle PV-DBOW, on cherche le vecteur de paramètres θ qui maximise la somme des logs vraisemblances de chaque contexte connaissant le document considéré.

$$\underset{\theta}{\text{ARGMAX}} \left\{ \sum_{\forall(\text{mots contexte,document})} log\mathbb{P}(\text{document}|\text{mots contexte}) \right\}$$

$$\underset{\theta}{\text{ARGMAX}} \left\{ -1/||\mathcal{C}|| \sum_{\mathcal{D}\in\mathcal{C}} \sum_{i=1}^{|\mathcal{D}|} \sum_{j\in[-m,m]} log\mathbb{P}(\mathcal{D}|w_j^{(\mathcal{D})})) \right\} \tag{1.1}$$

$$\mathbb{P}(\mathcal{D}|w_j)) = \frac{\exp(D_{\mathcal{D}}^t W_{w_j})}{\sum_{\mathcal{D}'\in\mathcal{C}} \exp(D_{\mathcal{D}'}^t W_{w_j})} \tag{1.2}$$

Avec,

$\mathcal{C}$ le corpus ;

W_w le vecteur de contexte du mot w ;

$D_\mathcal{D}$ le vecteur cible du document $\mathcal{D}$;

θ Vecteur de paramètres du modèle, composé pour chaque mot des vecteurs contextes (W_w) et pour chaque document des vecteurs cibles ($D_\mathcal{D}$) de taille d.

m La taille du contexte considéré.

Une fois le modèle optimisé, les vecteurs W_w appris pour chaque mot et $D_\mathcal{D}$ appris pour chaque document sont utilisés comme vecteur dense.

Dans le modèle PV-DM, on cherche le vecteur de paramètres θ qui maximise la somme des vraisemblances logarithmiques de chaque mot cible sachant le contexte considéré. Le vecteur de contexte pour le calcul de cette probabilité est le vecteur moyen des vecteurs de contextes de chacun des mots et du document dans le contexte considéré.

$$\underset{\theta}{\text{ARGMAX}} \left\{ \sum_{\forall(\text{cible,mots contexte,document})} log\mathbb{P}(\text{cible}|\text{mots contexte}, \text{document}) \right\}$$

L'optimisation se fait usuellement en utilisant une descente de gradient stochastique. Néanmoins, le calcul des probabilités conditionnelles étant coûteux, il existe plusieurs méthodes pour optimiser ces modèles :

***Hierarchical soft-max* :** Approximation de la probabilité soft-max $p(o|c)$ et utilisation d'un arbre de Huffman pour un accès plus efficace au vecteur correspondant au mot.

***Subsampling* :** Sous échantillonnage des mots ayant de fortes fréquences selon l'hypothèse qu'ils sont moins pertinents en termes de sémantique.

***Negative sampling* :** Minimisation de la vraisemblance logarithmique des cooccurrences qui sont peu ou pas observées. Cette méthode est souvent retrouvée sous l'appellation SGNS (*Skip-Gram with Negative Sampling*) dans la littérature.

En conclusion, dans cette section, nous avons présenté les méthodes permettant d'obtenir des vecteurs de représentations continues des documents ; ces méthodes exploitent les fréquences de cooccurrences des mots et des documents ou des mots entre eux, pour extraire des représentations sémantiques des documents. Dans la grande majorité de ces

méthodes, la matrice de cooccurrence joue un rôle central ; ainsi ces méthodes sont assimilables à des factorisations sous certaines contraintes de normalisations de cette matrice.

La limitation de ces approches réside dans le manque d'interpétabilité ; en effet, au contraire des approches discrètes qui ont des dimensions aisément interprétables (le nombre d'occurrences d'un concept ou d'un mot), les dimensions de l'espace sont ici non interprétables. Des travaux récents tentent de contourner ce problème, néanmoins le compromis entre taille des vecteurs et interprétabilité semble perdurer (Prouteau et al., 2021). Si elles ont le défaut de la non-interprétabilité, ces approches possèdent deux avantages majeurs qui ont fait d'elles les plus populaires pour le cas de la représentation de documents.

Premièrement, ces approches permettent de contrôler la taille des vecteurs de représentation et restent en particulier concurrentielles (au sens des performances sur des tâches de plus haut niveau) même pour des tailles de vecteur relativement petites (les valeurs usuelles sont entre 50 et 300 dimensions). Ces espaces sont souvent des sous-espaces du vocabulaire et respectent en particulier d_e la dimension des vecteurs $<< |V|$. Cela permet de facilement combiner ces méthodes avec des algorithmes d'apprentissage automatique et en particulier d'apprentissage profond. En effet, une petite taille de représentation facilite la convergence des algorithmes d'apprentissages (Shepperd and Cartwright, 2001).

Le deuxième argument en faveur des représentations dites continues est leur efficacité pratique dans de nombreuses tâches dites de haut niveau telles que la catégorisation de documents, l'apprentissage d'ordre, la traduction. Cette efficacité découle majoritairement de la combinaison entre apprentissage profond et représentation continue (Cho et al., 2014; Kiros et al., 2015).

De ces deux arguments, il ressort que l'un des grands avantages des représentations continues sur les représentations discrètes est leur complémentarité avec les réseaux de neurones profonds. Dans la prochaine section, nous allons présenter différente architecture de modèles neuronaux profonds et expliquer comment ces modèles sont devenus incontournables pour la représentation des documents, et plus généralement dans le domaine du traitement automatique du langage.

1.2 Modèles neuronaux profond

Un réseau de neurones artificiels est un système dont la conception est, à l'origine, schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Un réseau de neurones est en général composé

d'une succession de couches de neurones interconnectés. Classiquement, lors de l'inférence (on parle de propagation avant), la couche i prend en entrée la sortie de la couche $i-1$ et opère une transformation non-linéaire sur cette sortie qui sera à son tour utilisée comme entrée de la couche $i+1$. Les poids associés à chaque lien liant deux neurones sont les paramètres du modèle que l'on cherche à apprendre durant l'apprentissage. Le nombre de couches dans le réseau est appelé sa profondeur ; les modèles neuronaux dits profonds ont la propriété d'enchaîner de nombreuses couches résultant en des modèles pouvant atteindre plusieurs milliards de paramètres. Cette transformation non-linéaire qui survient à chaque couche peut être décomposée en deux étapes : la première consiste en une agrégation des valeurs en entrée du neurone. Dans la grande majorité des cas, la fonction somme est utilisée. La deuxième étape est l'activation du neurone qui se traduit par l'application une fonction non-linéaire dite d'activation sur le résultat de cette agrégation, cette étape indispensable permet au modèle d'être un approximateur universel et pas simplement une transformation linéaire de ces entrées. Les modèles sont alors entraînés en utilisant l'algorithme de rétro propagation et une méthode d'optimisation statistique, fréquemment une descente de gradient stochastique.

La remarquable efficacité des modèles neuronaux profonds dans de multiples domaines de l'analyse du signal sonore ou visuel, et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage, ont fait d'eux des modèles incontournables dans le domaine du traitement de l'information. Si la formalisation théorique des réseaux de neurones date des années 1950 (Lettvin et al., 1959), leur efficacité pratique croissante, en particulier avec des architectures dites profondes, est datée à partir de 2012 (Sermanet et al., 2012). Cette efficacité s'explique en partie par la gigantesque masse de données produites, en particulier par le web, qui permet aujourd'hui d'apprendre des modèles très profonds en modérant les risques de sur-apprentissage.

Dans cette section, nous ferons une brève introduction à différent type d'architectures neuronales en utilisant comme cas d'exemple les modèles de langage. Néanmoins, la recherche sur le développement d'architecture neuronale est un domaine foisonnant et nous ne prétendons pas ici présenter l'ensemble de ces travaux, mais simplement quelques travaux clés ayant eu à ce jour des retombées importantes pour les tâches de représentation des documents et de recommandation.

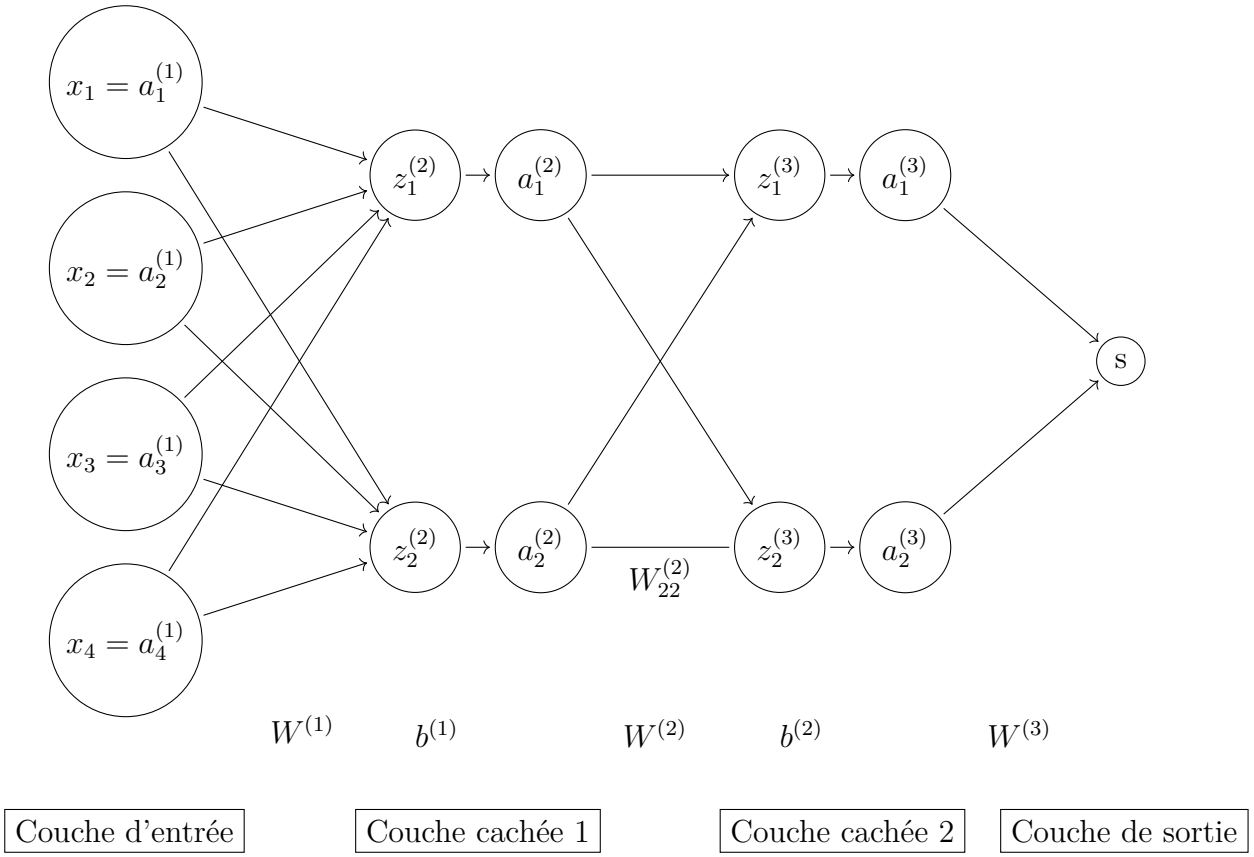


FIGURE 1.10 – Exemple de perceptron multi-couche. Source d'inspiration de l'image : https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

Perceptron multicouche Dans ce paragraphe, nous allons introduire le perceptron multicouche (MLP ou FNN) (Rumelhart et al., 1986) ; cette architecture est historiquement la première introduite, nous l'utiliserons par la suite comme architecture de comparaison mettant en exergue l'intérêt des développements des architectures plus récentes.

La figure 1.10 est une représentation graphique d'un MLP : nous allons utiliser l'exemple de cette figure pour détailler les différentes caractéristiques d'un MLP. À gauche sur la figure, on trouve la couche d'entrée du réseau de neurones, mathématiquement, elle correspond un vecteur x ; dans le cas pratique, ce vecteur est instancié avec les données d'entrée du problème. Dans une tâche de traitements du langage, on aura typiquement des vecteurs one-hot, sacs de mots ou des traits (features) extraits des documents. Cette couche a la particularité d'avoir pour fonction d'activation la fonction identité, on notera ainsi $a^{(1)} = x$ avec $a^{(1)}$ le vecteur représentant la sortie de la 1$^{\text{er}}$ couche, plus généralement,

nous noterons $a^{(i)}$ la sortie de la i-ème couche. Pour obtenir le vecteur $z^{(2)}$ d'entrée de la couche, la sortie de la première $a^{(1)}$ est multipliée par une matrice de passage $W^{(1)}$ qui sert de paramètre au modèle. Finalement, la sortie de la couche $a^{(1)}$ s'obtient en appliquant une fonction d'activation sur l'entrée $z^{(1)}$; les fonctions d'activations couramment utilisées dans la littérature sont la $ReLu(x) = max(0, x)$, la tangente hyperbolique ou la fonction sigmoïde (Sharma and Sharma, 2017). Ces fonctions sont choisies pour leur non-linéarité ainsi que pour d'autres propriétés mathématiques pertinentes au cas par cas, dérivabilité, symétrie, ensemble de définition et d'image. Le même processus se poursuit jusqu'au neurone de sortie s, et peut se résumer par le système d'équation utilisé lors de l'inférence, on parle de *propagation avant* :

$$\begin{aligned} \text{Couche d'entrée (Input layer)} &: x = z^{(1)} = a^{(1)} \\ \text{Couche cachée (Hidden layer)} &: z^{(i)} = W^{(i-1)}a^{(i-1)} + b^{i} \\ \text{Couche de sortie (Output)} &: s = W^{(l)}a^{(l)} \end{aligned} \tag{1.3}$$

L'apprentissage des paramètres se fait en cherchant à minimiser l'erreur entre la prédiction du modèle et la réalité terrain observé dans les données. Cette erreur se mesure par l'intermédiaire d'une fonction de coût que l'on cherchera à minimiser, cette fonction peut varier selon le cas d'application (régression, classification). L'algorithme de rétro propagation (Rumelhart et al., 1986) qui exploite le mécanisme de dérivation des fonctions composites permet de calculer efficacement la dérivée de l'erreur par rapport aux paramètres du modèle. Cette méthode à l'avantage d'être très efficace en temps de calcul. En outre, la rétro propagation est souvent mal comprise comme étant spécifique aux réseaux neuronaux multicouches, mais en principe, elle peut calculer les dérivées de n'importe quelle fonction. Une fois cette dérivée calculée, la recherche du minimum se fait en utilisant un algorithme de minimisation : dans le cas général, le problème n'ayant pas de solution analytique, des méthodes numériques rapides en temps de calcul sont utilisées tel que la descente de gradient stochastique (Ketkar, 2017).

Réseau de neurones récurrent (Recurent Neural Network : RNN)

Le réseau de neurones récurrent (Recurent Neural Network : RNN) est un réseau de neurones artificiel spécialement conçu pour des données séquentielles. Pour cette raison, cette architecture a rapidement été appliquée au langage naturel. Formellement, un réseau de neurones artificiel est dit récurent dès lors qu'il existe au moins un cycle dans sa structure.

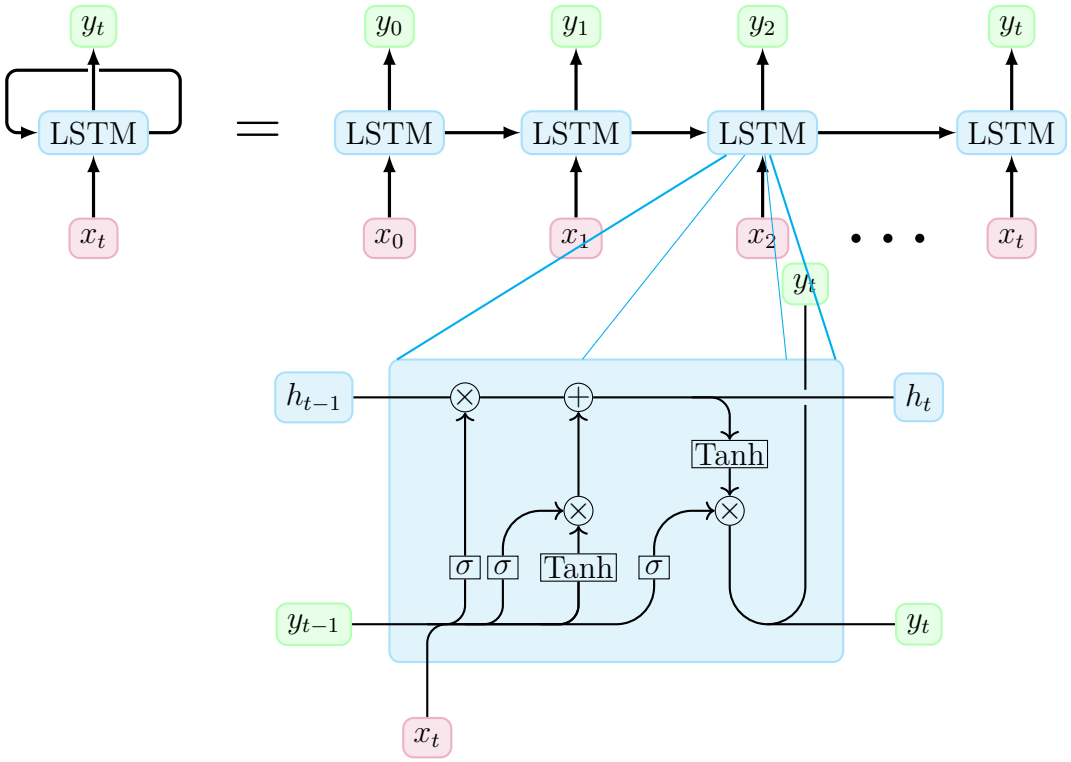


FIGURE 1.11 – Architecture générique d'un LSTM, avec un représentation plié (gauche du signe égale), une représentation équivalente déplié (droite du signe égale) et une magnificence sur la structure interne d'une cellule LSTM.

Il existe de nombreuses architectures de RNNs (Rumelhart et al., 1986). L'une des plus efficaces dans le domaine du traitement du langage naturel est composée d'unités dites *LSTM* (Hochreiter and Schmidhuber, 1997) (Long Short-Term Memory, mémoire à court et long terme en français). Par rapport à d'autres architectures, le LSTM a l'avantage d'être conçu pour résoudre le problème dit d'*évanescence du gradient*. Un problème majeur de la descente de gradient pour les architectures RNN standards est que les gradients d'erreur s'évanouissent (c'est-à-dire qu'ils peuvent tendre vers zéro) ou explosent (c'est-à-dire qu'ils peuvent tendre vers l'infini) de manière exponentielle avec la taille du décalage temporel entre les événements importants. Cette impossibilité pratique vient du fait que les ordinateurs utilisent des nombres à précision finie (Courbariaux et al., 2014; Gupta et al., 2015). Le LSTM tente de surmonter ces problèmes en réduisant le contexte d'un neurone à son propre état passé, offrant la possibilité d'un gradient inchangé et les informations inter-neurones peuvent ensuite être explorées dans les couches suivantes. Les mémoires de différentes portées, y compris la mémoire à long terme, peuvent être apprises en dépit du problème de la disparition et de l'explosion du gradient. En d'autres termes, le LSTM peut théoriquement apprendre des tâches qui nécessitent la mémoire d'événements qui se sont produits des milliers, voire des millions, de pas de temps discrets plus tôt. Le LSTM fonctionne même avec de longs délais entre les événements significatifs et peut traiter des signaux qui mélangent des motifs de basse et de haute fréquence.

La figure 1.11 représente l'architecture générique d'un RNN. Le RNN est représenté de deux manières équivalentes : une dite *compacte* (sur la gauche de la figure 1.11) représentant une seule cellule avec une connexion récurrente ; l'autre dite *développée* (sur la droite de la figure 1.11) représentant plusieurs cellules connectées de manière séquentielle. Contrairement aux réseaux de neurones non récurrents, l'entrée d'un RNN est séquentielle. Sur le schéma, cette entrée est un ensemble ordonné $x = \{x_0 \dots x_t \dots x_n\}$ de taille potentiellement variable entre les exemples. Cette dualité des représentations sert à illustrer le fait que les éléments sont donnés dans l'ordre de la séquence au RNN (représentation développée) et qu'un seul RNN reçoit tour à tour en entrée chacun des éléments(représentation compacte). Au passage de chaque élément une sortie ici y_t est produite ainsi qu'un état caché h_t. Les deux serviront de mémoire au modèle et seront réutilisés comme entrée pour l'élément suivant de la séquence. Grâce à cette architecture, il est en particulier possible de rétro propager le gradient une fois toute la séquence ingérée. On parle de propagation arrière à travers le temps. Cela trouve bien évidemment des applications directes dans le langage naturel comme la détection de sentiment où les données sont naturellement

séquentielles (langage) mais les vérités terrain s'appliquent à la séquence (sentiment relié à la phrase). On trouve également des applications dans le cas inverse dans lequel on souhaite à générer des séquences à partir de données non séquentielles, typiquement dans le cas de la génération de texte. Enfin, d'autres applications cherchent à produire des séquences à partir de séquences, par exemple sur une tâche de traduction, dans ce cas, on parle de modèle *encodeur-décodeur* ou séquence vers séquence (seq2seq (Sutskever et al., 2014)). Cette désignation vient du fait qu'un RNN va chercher à condenser l'information de la séquence entrée (l'encodeur) et un deuxième RNN distinct va chercher à générer la séquence réponse depuis cette information condensée (le décodeur).

Certains modèles comme ELMo (Cho et al., 2014) ou Skip-thought vectors (Kiros et al., 2015) ont exploité cette architecture pour la tâche de représentation des documents. ELMo (Embeddings from Language Models) est basé sur un *LSTM bidirectionnel* (l'ordre naturel et l'ordre inverse du texte en entrée sont considérés) appliqué à une tâche de modélisation du langage pour produire des représentations dites contextualisées des mots et des documents. La particularité des représentations contextualisées est qu'elles ne produisent pas une représentation fixe pour chaque mot, en effet, ELMo examine l'ensemble de la phrase avant d'assigner une représentation à chaque mot qu'elle contient. Ces représentations des mots peuvent ensuite être agglomérées pour obtenir une représentation du document. Skip-thought qui s'appuie également sur une architecture RNN (avec des cellules GRU (Cho et al., 2014)) propose lui de prédire non plus un mot à partir d'un contexte, mot suivant et mot précédent dans le cas de ELMo, mais directement des phrases contextes (suivante et précédente de la phrase cible) à partir d'une phrase cible. C'est donc un modèle encodeur-décodeur, ici un RNN dit encodeur est chargé de créer une représentation dense d'une phrase cible et cette représentation est donnée en entrée à deux RNN distincts dits décodeurs qui ont pour rôle de générer respectivement les phrases suivante et précédente de la phrase cible.

Une des limitations des modèles séquence vers séquence vient du fait que le vecteur d'état caché est transporté le long de la séquence. Ce vecteur agit un peu comme un goulot d'étranglement pour l'information. Ce phénomène a été observé en pratique dans différents contextes applicatifs (Karatzoglou et al., 2018) et étudié en détail par Alon and Yahav (2020). Par exemple, dans le cas d'une traduction, il semble évident que l'information pertinente à conserver n'est pas la même pour chaque mot généré ; de manière assez logique, certains mots dans la phrase d'entrée nécessiteront un maximum d'attention pour générer un mot de la phrase de sortie. De plus, d'un point de vue purement

grammatical en français par exemple, il est nécessaire de focaliser sur le sujet lorsqu'on accorde le verbe, par contre, lorsqu'on accorde l'auxiliaire « avoir » on doit alors s'intéresser au complément d'objet direct dans le cas ou il est placé avant. Ainsi, un algorithme de génération souhaitant respecter la grammaire devrait faire de même. Pour pallier ce problème, un mécanisme dit *mécanisme d'attention* a été introduit par (Bahdanau et al., 2015) ; l'idée de ce mécanisme est d'imiter l'attention cognitive en cherchant à se focaliser sur les parties importantes des données d'entrée et à estomper le reste. La partie des données qui est plus importante que les autres dépend du contexte et est apprise durant l'entraînement à partir les données par descente de gradient.

Nous proposons dans la figure 1.12 un exemple d'utilisation du mécanisme d'attention dans le cas de la traduction. Sur la figure, on peut voir que pour générer le mot « signed » le modèle s'intéresse en particulier aux mots « a été signé », en revanche pour le mot « August » les mots « en août » sont logiquement privilégiés. Mathématiquement, les

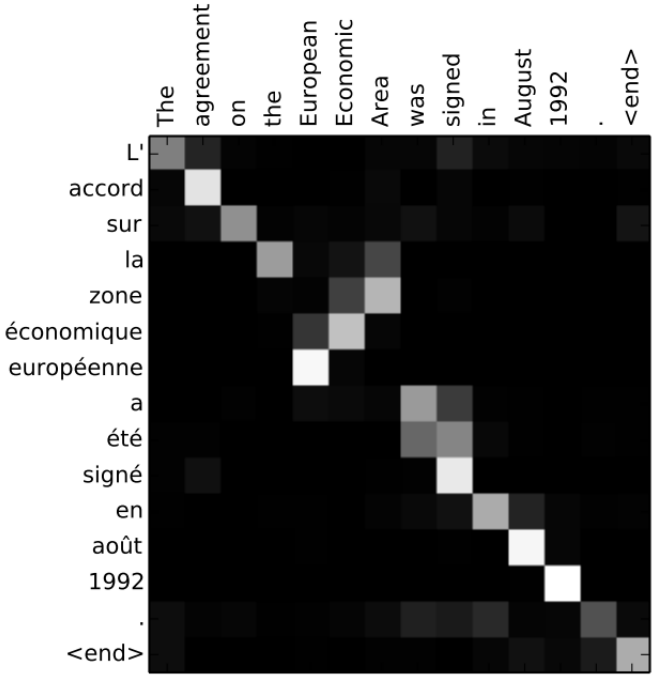


FIGURE 1.12 – Exemple de mécanisme d'attention dans le cas de la traduction. Source de l'image : (Bahdanau et al., 2015)

mécanismes d'attention ont été définis de différentes manières : *additive attention* (Bahdanau et al., 2015), *multiplicative attention* (Luong et al., 2015), *self-attention* (Lin et al.,

2017), *key-value attention* (Daniluk et al., 2019) [3]. Il est important de remarquer que contrairement aux méthodes précédentes, les modèles d'attention permettent naturellement de considérer des paramètres agissant sur un produit des termes en entrée, même si ce produit est calculé dans un (ou plusieurs dans le cas de l'attention à têtes multiples) sous-espace du produit cartésien de ces termes. Dans la suite, nous allons nous intéresser au mécanisme dit de *self-attention* qui sont à la base d'une architecture neuronale appelée Transformers.

Transformers

Les Transformers (on peut les retrouver mentionnés sous le termes transformateurs en français) sont des architectures neuronales profondes. Leur particularité réside dans l'utilisation de blocs dits de *multi-head attention* (ce qui pourrait être traduit par attention à tête multiple). Comme les réseaux neuronaux récurrents (RNN), les transformateurs sont conçus pour traiter des données d'entrée séquentielles. Il est d'ailleurs possible de formaliser les Transformers comme des RNNs (Katharopoulos et al., 2020).

Au contraire des RNNs cités précédemment, le mécanisme d'attention fournit un contexte pour toute position dans la séquence d'entrée. Par exemple, si les données d'entrée sont une phrase en langage naturel, le Transformer n'a pas besoin de traiter le début de la phrase avant la fin. Au contraire, il identifie le contexte qui confère un sens à chaque mot de la phrase. Cette caractéristique permet une plus grande parallélisation que les RNNs et réduit donc les temps d'apprentissage.

Les Transformers sont un modèle de choix pour les problèmes de TALN, remplaçant les modèles RNN tels que LSTM. La parallélisation supplémentaire de l'entraînement permet l'entraînement sur des ensembles de données plus importants, ce qui était impossible auparavant. Cela a conduit au développement de systèmes pré-entraînés tels que BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) et GPT (Generative Pre-trained Transformer) (Radford et al., 2019), qui ont été entraînés avec de très grands ensembles de données linguistiques, tels que le Corpus Wikipédia (Radford et al., 2019; Izsak et al., 2021) et le Common Crawl (Radford et al., 2019; Devlin et al., 2019), et peuvent être raffinés pour des tâches spécifiques.

À l'origine, le modèle Transformers a été développé par Vaswani et al. (2017). La figure 1.13 décrit l'architecture proposée par les auteurs. Même si cette architecture a

3. Le blog de Sebastian Ruder Deep Learning for NLP Best Practices `https://ruder.io/deep-learning-nlp-best-practices/index.html#attention` fournit un bon point d'entrée détaillant succinctement ces différents modèles d'attention.

par la suite pu être adaptée aux différents cas d'application (par exemple l'utilisation d'un décodeur simple) elle n'a jusqu'ici pas été dénaturée et sa compréhension permet d'aborder sereinement la compréhension des architectures qui lui ont succédé. Même si les Transformers servent aujourd'hui pour d'autres types de données que les données textuels, nous allons choisir ces données pour expliquer le fonctionnement du modèle.

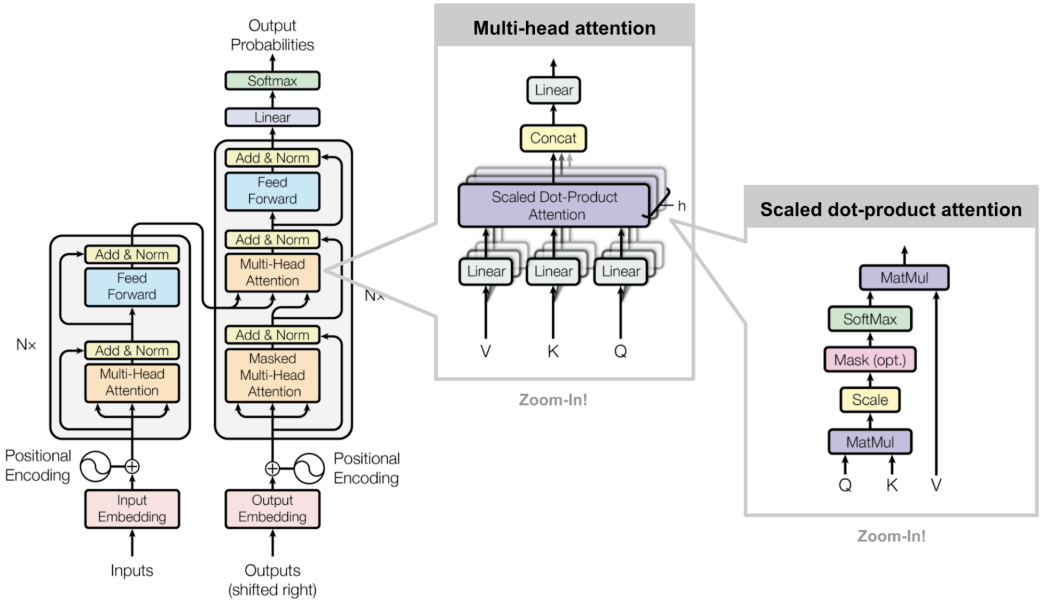


FIGURE 1.13 – Schéma de l'architecture Transformers dans sa version encodeur-décodeur. Source de l'image : (Vaswani et al., 2017)

Dans le cas d'un modèle de langage, l'entrée du modèle prend du texte brut et les mots du texte sont alors représentés avec des vecteurs one-hot (Press and Wolf, 2017). Formellement, l'entrée correspond donc à une matrice de taille *taille du texte* $\times |V|$: cette matrice est directement projetée dans un espace de représentation des mots ($|V| \times d_e$) qui peut être soit pré-apprise, soit apprise durant l'entraînement (Press and Wolf, 2017). Une fois la projection opérée, l'étape suivante consiste à ajouter l'ordre des mots dans notre modèle. En effet, dans le cas d'un RNN cette information est naturellement capturée par le processus de récurrence. Ici, nous allons voir que chacun des mots passent simultanément dans le modèle pour cette raison. Il est nécessaire d'avoir recours à une méthode d'encodage de la position des mots dans la phrase. On parle d'*encodage positionnel* (Positionnal Encoding) : un tel encodage doit satisfaire quelque critères essentiels tels que le déterminisme, l'unicité de l'encodage pour une position donnée, une plage de valeurs

bornée, la généralisation à des phrases plus longues ou plus courtes que celle de l'entraînement, et la cohérence des distances inter-mots pour des phrases de longueurs différentes. La méthode utilisée par les auteurs satisfait tous ces critères et consiste à encoder cette information dans un vecteur de taille d_e qui sera ensuite ajouté au vecteur de chaque mot. Pour cela, en considérant *pos* la position à encoder, le vecteur est engendré $PE_{pos} \in \mathbb{R}^{d_e}$ selon la formule suivante :

$$PE_{pos_i} = \begin{cases} sin(w_k \times pos) \text{ si } i = 2k \\ cos(w_k \times pos) \text{ si } i = 2k+1 \end{cases} \quad \text{Avec } w_k = \frac{1}{10000^{2k \times d_e}}$$

L'un des autres grands avantages de cette méthode est la facilité d'apprentissage des positions relatives. En effet, pour tout décalage *dec* fixe, $PE_{pos+dec}$ peut être représenté comme une fonction linéaire de PE_{pos} [4]. Une fois calculé, le vecteur d'encodage positionnel est directement ajouté au vecteur de représentation de chaque mot. Même si le choix de l'ajout plutôt que de la concaténation n'est pas justifiée par les auteurs, on peut remarquer que la méthode a tendance à n'utiliser que les premières dimensions pour stocker les positions. Dans le cas où la représentation est apprise depuis zéro cela laisse la possibilité au modèle, soit de stocker les représentations sémantiques des mots après les premières dimensions pour éviter d'interférer avec l'encodage positionnel. Soit, à l'inverse, dans le cas où la séparabilité n'apporte pas un avantage, le modèle peut librement apprendre une combinaison de l'encodage sémantique et positionnel sur certaines dimensions. Ce choix semble donc motivé à la fois par la volonté de réduire le nombre de paramètres, mais aussi et surtout par la volonté de laisser une liberté maximale au modèle dans l'apprentissage des représentations apprises.

Une fois l'encodage positionnel réalisé, nous arrivons au cœur de l'architecture Transformers avec un module qui pourra être répété plusieurs fois dans l'architecture ; ce module contient la couche d'attention à tête multiple qui est l'élément central du modèle. L'idée centrale de cette couche est d'exécuter un mécanisme d'attention plusieurs fois en parallèle. Les sorties d'attention indépendantes sont ensuite concaténées et transformées linéairement dans la dimension attendue. Intuitivement, les têtes d'attention multiples permettent de s'occuper différemment de certaines parties de la séquence (par exemple, les dépendances à long terme par rapport aux dépendances à court terme). Au cœur du

4. Le lecteur peut se référer à https://timodenk.com/blog/linear-relationships-in-the-transformers-positional-encoding/ pour une preuve de cette propriété.

processus, on retrouve la matrice d'attention telle que présentée en Figure 1.12. Selon la formulation proposée par (Vaswani et al., 2017) la couche *Attention* reçoit ses données d'entrée sous la forme de trois paramètres, appelés *Requête* (Q pour *Query*), *Clé* (K pour *Key*) et $Valeur$ (V pour *Value*). Ces trois paramètres ont une structure similaire : chaque mot de la séquence étant représenté par un vecteur. La représentation de la séquence d'entrée obtenue précédemment est transmise aux trois paramètres (*Requête*, *Clé* et $Valeur$) du premier encodeur d'auto-attention, qui produit également une représentation codée pour chaque mot de la séquence d'entrée, qui intègre également les scores d'attention pour chaque mot. En passant par tous les encodeurs de la pile, chaque module d'auto-attention ajoute également ses propres scores d'attention dans la représentation de chaque mot. Dans le Transformers, le module *Attention* répète ses calculs plusieurs fois en parallèle. Chacun d'entre eux est appelé une tête d'attention. Le module d'attention divise ses paramètres *Requête*, *Clé* et $Valeur$ en N parties et fait passer chaque partie indépendamment par une tête distincte. Tous ces calculs d'attention similaires sont ensuite combinés pour produire un score d'attention final. Cela donne au Transformers une plus grande puissance pour encoder de multiples relations et nuances pour chaque mot.

$$\begin{aligned} &MultiHead(Q,K,V) = Concat(head_1,\ldots,head_h)W^O \\ &\text{avec } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \\ &\text{et } Attention(Q,K,V) = softmax(\tfrac{QK^t}{\sqrt{d_k}})V \end{aligned} \tag{1.4}$$

Ici les matrices de paramètres $W^Q \in \mathbb{R}^{d_e \times d_k}$, $W_i^K \in \mathbb{R}^{d_e \times d_k}$, $W_i^V \in \mathbb{R}^{d_e \times d_k}$ et $W^O \in \mathbb{R}^{hd_v \times d_e}$ seront apprises durant l'apprentissage. Enfin, d_k et d_v sont des méta-paramètres du modèle, d_k contrôle la taille du sous-espace dans lequel sera calculée la matrice d'attention et d_v contrôle indirectement le nombre de têtes puisque la formule $h = d_e/d_v$ doit être respectée. Selon le cas d'application, les matrices Q, K et V peuvent être instanciées différemment dépendamment de la matrice que l'on cherche à calculer, de la même manière un masquage peut être opéré selon la tâche envisagée.

Une fois cette nouvelle représentation apprise, une couche résiduelle et de normalisation est appliquée, qui a pour rôle de garantir la persistance des informations d'entrées et de conserver une certaine proximité entre la représentation d'entrée et de sortie. Cette dernière propriété est connue pour favoriser l'apprentissage de modèle très profond. Finalement, un perceptron à deux couches dans lequel les vecteurs de représentation enrichis des mots sont donnés un à la fois achève le processus.

À partir de modèles Transformers entraînés pour une tâche de modélisation du lan-

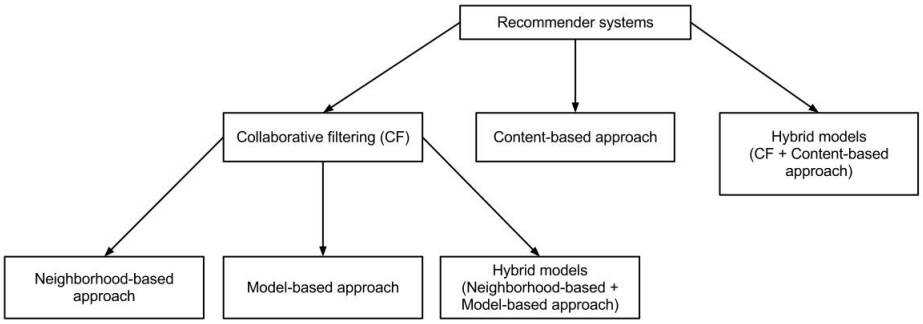


FIGURE 1.14 – Taxonomie des systèmes de recommandations. Figure issue de Wikipédia

gage naturel, des méthodes d'extraction de représentations de document ont été développées (Reimers and Gurevych, 2019; Press and Wolf, 2017), elles constituent aujourd'hui l'état de l'art du domaine.

1.3 Recommandation

Comme nous l'avons présenté en introduction (Section 0.5) la recommandation de contenu à large échelle dans un contexte d'apprentissage non formel constitue aujourd'hui un verrou scientifique au développement d'applications telles que celles envisagées dans X5GON. Pour cette raison, dans cette section, nous allons nous intéresser à la question de la recommandation de contenu à large échelle. En particulier, nous nous intéresserons aux approches applicables à large-échelle et susceptibles d'être utilisées avec comme donnée d'entrée du texte brut non structuré. Un système de recommandation est un système de filtrage de l'information cherchant à prédire la préférence d'un utilisateur parmi une série d'éléments. Les systèmes de recommandation sont utilisés dans une large variété de domaine, depuis la génération de listes d'écoutes de musique, la recommandation de produits sur des sites marchands en ligne jusqu'à la recommandation de contenus ou même d'utilisateurs dans les réseaux sociaux. Historiquement, 3 types d'approches ont été envisagés : les approches basées sur le contenu, les approche basées sur le filtrage collaboratif et les approches hybrides. La Figure 1.14 fournit une taxonomie du domaine. Les systèmes actuels sont pour la très grande majorité hybrides et exploitent les avantages et les inconvénients des différentes approches. Dans les paragraphes qui suivent, nous allons présenter plus en détail ces trois types d'approches.

1.3.1 Les approches basées sur le contenu

Les approches basées sur le contenu exploitent les contenus historiques précédemment consultés par un utilisateur pour lui fournir des recommandations. On parle aussi d' *approches basées sur la personnalité* (Pazzani and Billsus, 2007). Bien que les détails des divers systèmes diffèrent, les systèmes de recommandation basés sur le contenu ont en commun une volonté de décrire les éléments qui peuvent être recommandés, une volonté de créer un profil de l'utilisateur qui décrit les types d'éléments que l'utilisateur aime, et une volonté de comparer les éléments au profil de l'utilisateur pour déterminer ce qu'il faut recommander. En particulier, ces approches ne cherchent pas à exploiter les corrélations entre les préférences de différents utilisateurs et pour cette raison, la recommandation est complètement indépendante des activités des autres utilisateurs. Le profil est souvent créé et mis à jour automatiquement en réponse au retour d'information sur le caractère désirable des articles qui ont été présentés à l'utilisateur. Du fait de leur structure, ces approches utilisent souvent des représentations sémantiques des documents à fin de trouver des ressources proches de l'historique de l'utilisateur (Kowsari et al., 2019). Un exemple simpliste de ces approches serait de recommander à un utilisateur les 10 plus proches voisins dans l'espace de représentation du dernier élément consultés. Dans ce cas, la recommandation serait indifférente vis-à-vis de l'utilisateur, tous les utilisateurs consultant la même ressource obtiendraient le même document. Ce type de recommandation peut aussi tenir compte de métadonnées sur le contenu ou de données contextuelles (le jour de publication du contenu, l'année en cours...).

Nombre de ces approches utilisent un profil utilisateur qui concentre deux types d'informations : (i) un modèle des préférences de l'utilisateur, c'est-à-dire une description des types d'articles qui l'intéressent et (ii) un historique des interactions de l'utilisateur avec le système de recommandation. Il existe de nombreuses représentations alternatives possibles de cette description, mais une représentation commune est une fonction qui, pour tout élément, prédit la probabilité que l'utilisateur soit intéressé par cet élément (Montaner et al., 2003).

Parfois, le système propose à l'utilisateur de créer lui-même son profil. On parle de personnalisation par l'utilisateur (Montaner et al., 2003). Dans ce cas, le système fournit une interface, par exemple un questionnaire composé de liste de choix ou d'intérêts, mais également dans certains cas de zones de texte libre. L'utilisateur doit alors entrer ces informations (par exemple remplir un formulaire). Et un simple processus de mise en

correspondance de la base de données est utilisé pour trouver les éléments qui répondent aux critères spécifiés et les afficher à l'utilisateur. Les systèmes de personnalisation par l'utilisateur présentent plusieurs limites. Tout d'abord, ils exigent un effort de la part de l'utilisateur et il est difficile d'obtenir que de nombreux utilisateurs fassent cet effort. De plus, ce sont des systèmes non évolutifs, lorsque des préférences d'un utilisateur change (par exemple, un utilisateur peut ne pas suivre la natation, mais s'y intéresser pour le temps des jeux olympiques) il doit mettre à jour manuellement son profil. Deuxièmement, les systèmes de personnalisation ne permettent pas de déterminer l'ordre dans lequel les éléments doivent être présentés et peuvent trouver soit trop peu, soit trop d'éléments correspondants à afficher. Enfin, ils sont très difficilement applicables à un contexte multiculturel, multidisciplinaire qui plus est à large échelle avec des contenus non structurés. Nous les laisserons donc de côté pour cet état de l'art.

D'autres systèmes de recommandations sont basés sur des règles : on parle de *système à base de règles* (Pazzani and Billsus, 2007) ; le système de recommandation possède des règles pour recommander d'autres produits en fonction de l'historique de l'utilisateur. Par exemple, un système peut contenir une règle qui recommande la suite d'un livre ou d'un film aux personnes qui ont consommé le premier article de la série. Une autre règle peut recommander le nouveau CD d'un artiste aux utilisateurs qui ont acheté les premiers CD de cet artiste. Les systèmes à base de règles peuvent représenter plusieurs raisons courantes de faire des recommandations, mais ils n'offrent pas les mêmes recommandations détaillées et personnalisées que les autres systèmes de recommandation. En particulier, ils nécessitent souvent des données très structurées (au sens de métadonnées riches exprimant bien les différentes relations entre les ressources) et une connaissance préalable par les développeurs du système de ces contenus et des recommandations susceptibles d'intéresser les utilisateurs interagissant avec. Pour ces raisons, ils sont une fois encore très difficilement applicables à un contexte non formel qui plus est à large échelle avec des contenus non étiquetés.

Enfin, d'autres systèmes cherchent justement à apprendre la fonction de préférences des utilisateurs : ces méthodes vont usuellement utiliser des algorithmes d'apprentissage automatique qui chercheront à prédire les interactions passées (Pazzani and Billsus, 2007; Kowsari et al., 2019). Cela se fait soit par un retour d'information explicite, dans lequel l'utilisateur évalue les ressources via une interface de collecte d'informations, soit implicitement en observant les interactions de l'utilisateur avec les ressources. Par exemple, dans un contexte d'e-commerce, si un utilisateur achète un article, c'est un signe que

l'utilisateur aime l'article, tandis que si l'utilisateur achète et retourne l'article, c'est un signe que l'utilisateur n'aime pas l'article. On parle de retour d'informations implicites, positif ou négatif. En général, il y a un compromis à faire, car les méthodes implicites peuvent collecter une grande quantité de données avec une certaine incertitude quant à savoir si l'utilisateur aime réellement l'article. En revanche, lorsque l'utilisateur évalue explicitement les éléments, il y a peu ou pas de bruit dans les données d'apprentissage, mais les utilisateurs ont tendance à fournir un retour explicite sur un petit pourcentage seulement des éléments avec lesquels ils interagissent.

Dans les paragraphes suivants, nous passons en revue un certain nombre d'algorithmes de classification. Ces algorithmes sont le composant clé des systèmes de recommandation basés sur le contenu, car ils apprennent une fonction qui modélise les intérêts de chaque utilisateur. Étant donné un nouvel élément et le modèle de l'utilisateur, la fonction prédit si l'utilisateur sera intéressé par l'élément. De nombreux algorithmes d'apprentissage de classification créent une fonction qui fournit une estimation de la probabilité qu'un utilisateur aime un élément non vu (Pazzani and Billsus, 2007). Cette probabilité peut être utilisée pour trier une liste de recommandations. Alternativement, un algorithme peut créer une fonction qui prédit directement une valeur numérique telle que le degré d'intérêt. Certains des algorithmes ci-dessous sont des algorithmes d'apprentissage automatique traditionnels conçus pour fonctionner sur des données structurées, typiquement représenté par peu de traits ayant des valeurs numériques ou catégorielles. Lorsqu'ils fonctionnent sur du texte, celui-ci est d'abord converti en vecteurs, représentations sémantiques usuellement discrètes (sac-de-mots, TF-IDF) en sélectionnant un petit sous-ensemble de traits comme attributs. En revanche, d'autres algorithmes sont conçus pour travailler dans des espaces de grande dimension et ne nécessitent pas d'étape de prétraitement des données pour la sélection des traits. Les paragraphes suivants constituent une brève description des algorithmes les plus importants. Un examen approfondi est présenté dans (Kowsari et al., 2019; Montaner et al., 2003; Pazzani and Billsus, 2007; Sebastiani, 2002) qui traitent en particulier d'autres méthodes telles que les régressions linéaires et logistiques, les *machines à vecteurs de support* ainsi que les méthodes par agrégation (bag and boost).

1.3.1.1 Arbres de décision et Forêts aléatoires

L'idée générale d'un arbre de décision consiste à partitionner les données récursivement jusqu'à ce que les groupes engendrés contiennent uniquement des données de la même classe. Le gain d'information ou l'impureté de Gini attendu est un critère couramment

utilisé pour sélectionner les caractéristiques les plus informatives qui dirigent le partitionnement (Hastie et al., 2009). Les arbres de décision fonctionnent particulièrement lorsque le nombre de traits est petit, à l'inverse lorsque le nombre de traits devient trop grand, ils sont très sujets au sur-apprentissage (Ho, 1995). Cela s'explique par une des conséquences théoriques des critères de partitionnement des arbres de décision qui est une préférence pour les petits arbres avec peu de partition. Il peut être démontré expérimentalement que les tâches de classification de textes (comme la recommandation) impliquent fréquemment un grand nombre de caractéristiques pertinentes (Joachims, 1998). Par conséquent, la tendance d'un arbre de décision à baser les classifications sur le moins de tests possible peut conduire à de mauvaises performances en classification de textes. Cependant, lorsqu'il existe un petit nombre de traits pertinent pour représenter les ressources, les performances, la simplicité et la compréhensibilité des arbres de décision pour les modèles basés sur le contenu sont autant d'avantages. L'arbre de décision est un algorithme très rapide à la fois pour l'apprentissage et la prédiction, mais il est également extrêmement sensible aux petites perturbations des données (Hastie et al., 2009). Ces effets peuvent être atténués par des méthodes de validation et d'élagage (Hastie et al., 2009). Ce modèle a également des problèmes avec la prédiction hors échantillon, cas courant de la recommandation, en particulier à large-échelle.

La technique des *forêts aléatoires* ou des forêts de décision aléatoires est une méthode d'apprentissage d'ensembles pour la classification. Cette méthode, qui permet d'utiliser plusieurs arbres de décision en parallèle, a été introduite par Ho (1995). L'idée principale consiste à générer plusieurs arbres de décision aléatoires, les prédictions sont alors attribuées en fonction du vote de chacun des arbres, l'importance attribuée à chaque arbre dépend alors de sa performance sur les données d'entraînements. Les forêts aléatoires sont très rapides à entraîner pour les ensembles de données textuelles par rapport à d'autres techniques telles que l'apprentissage profond, mais assez lentes lors de l'inférence (Bansal et al., 2018). Ainsi, afin d'obtenir une structure plus rapide, le nombre d'arbres dans la forêt doit être réduit, car plus d'arbres dans la forêt augmente la complexité temporelle à l'inférence. Cela est très problématique pour la tâche de recommandation qui doit répondre aux contraintes inverses, à savoir une flexibilité relative sur le temps d'apprentissage fait hors-ligne, mais un temps d'inférence très rapide pour limiter le temps d'attente des utilisateurs. Il faut alors opérer un compromis entre une grande forêt très « sage » et lente, et une petite forêt plus rapide, mais plus sujette aux erreurs et au sur-apprentissage.

1.3.1.2 Plus proches voisins

L'algorithme du *plus proche voisin* (k-PPV) stocke simplement en mémoire toutes ses données d'apprentissage, ici des descriptions textuelles d'éléments étiquetés implicitement ou explicitement. Afin de classer un nouvel élément non étiqueté, l'algorithme le compare à tous les éléments stockés en utilisant une fonction de similarité et détermine le « plus proche voisin » ou les k plus proches voisins. L'étiquette de classe ou le score numérique d'un élément hors échantillon s'obtient alors grâce aux votes de ses k plus proches voisins. k-PPV comme l'ensemble des méthodes par voisinage fait partie de la famille des algorithmes d'apprentissage paresseux. Cette dénomination vient du fait qu'aucun modèle sous-jacent n'est nécessaire dans le but de réaliser les prédictions, la prédiction est en fait complètement dépendante de l'instance à prédire.

La fonction de similarité utilisée par l'algorithme des plus proches voisins dépend du type de données. Une distance euclidienne est préférable lorsque les données sont représentées dans des dimensions aux unités comparables et de pertinence égale, typiquement des dimensions booléennes (présence/absence). Lorsqu'on utilise un espace de représentation sémantique des documents, la mesure de similarité du cosinus est souvent utilisée, car les variations de fréquence ou les dimensions apprise ne garantissent pas d'être comparables entre elles. Par conséquent, elle est appropriée pour le texte lorsque nous voulons que deux documents soient similaires lorsqu'ils traitent du même sujet, mais pas lorsqu'ils ne traitent pas du même sujet. Cette approche, malgré sa simplicité, a largement été utilisée dans le contexte de la recommandation (Kolodner, 2014). Cependant, k-PPV est limité par les contraintes de stockage des données pour les grands problèmes de recherche des plus proches voisins. De plus, la performance de k-PPV dépend de la recherche d'une fonction de distance significative, ce qui fait de cette technique un algorithme très dépendant des données.

1.3.1.3 Rétroaction sur la pertinence

La *rétroaction sur la pertinence* est une technique adoptée dans le domaine de la recherche d'information qui aide les utilisateurs à affiner progressivement leurs requêtes en fonction des résultats de recherche précédents. Elle consiste à faire remonter au système les décisions des utilisateurs sur la pertinence des documents récupérés par rapport à leurs besoins d'information. La rétroaction sur la pertinence a été adaptée à la catégorisation des textes, grâce à la *formule de Rocchio* (Harman, 1992). Le principe général est de per-

mettre aux utilisateurs de noter les documents suggérés par le système de recommandation par rapport à leurs besoins et leurs attentes. Les classifications linéaires peuvent ensuite être utilisées pour affiner progressivement le profil de l'utilisateur ou pour entraîner un algorithme d'apprentissage qui déduira le profil de l'utilisateur.

Cet algorithme représente les documents comme des vecteurs, de sorte que les documents au contenu similaire ont des vecteurs similaires : pour cela, on peut utiliser les méthodes présentées en Section 1.1, le plus couramment le TF-IDF ou un dérivé est choisi. L'apprentissage est réalisé en combinant les vecteurs de documents, d'exemples positifs (par exemple un document récemment consulté) ou négatifs (un document ignoré lors d'une recommandation passée) en un vecteur prototype de chaque classe (positif et négatif). Pour classer un nouveau document, les similarités entre les vecteurs prototypes d'une part et le vecteur de document correspondant d'autre part sont calculées, et la classe dont le vecteur document a la plus grande valeur de similarité est alors attribuée au document. L'approche de classification basée sur Rocchio n'a pas de fondement théorique et il n'y a pas de garanties sur la performance ou la convergence (Pazzani and Billsus, 2007). De plus, l'algorithme de Rocchio pour la classification des textes comporte de nombreuses limites, comme le fait que l'utilisateur ne peut récupérer que quelques documents pertinents à l'aide de ce modèle. Néanmoins, il a été appliqué dans plusieurs systèmes de recommandation, notamment au début des années 2000 (Ahn et al., 2007; Balabanović and Shoham, 1997).

1.3.1.4 Méthodes probabilistes

Parmi les méthodes probabilistes, l'une des plus couramment utilisée est l'approche *bayésienne naïve*. L'approche bayésienne naïve appartient à la classe générale des classificateurs bayésiens. L'idée fondamentale de ces approches est de construire un modèle probabiliste basé sur les données observées précédemment. Le modèle estime la probabilité *a posteriori*, $\mathbb{P}(\mathcal{C}|\mathcal{D})$, du document $\mathcal{D}$ appartenant à la classe $\mathcal{C}$. Cette estimation est basée sur la probabilité *a priori* $\mathbb{P}(\mathcal{C})$ d'observer un document dans la classe $\mathcal{C}$, $\mathbb{P}(\mathcal{D}|\mathcal{C})$, la probabilité d'observer le document $\mathcal{D}$ étant donné $\mathcal{C}$, et $\mathbb{P}(\mathcal{D})$ la probabilité d'observer l'instance $\mathcal{D}$. En utilisant ces probabilités, le théorème de Bayes est appliqué pour calculer $\mathbb{P}(\mathcal{C}|\mathcal{D})$:

$$\mathbb{P}(\mathcal{C}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{C})\mathbb{P}(\mathcal{D}|\mathcal{C})}{\mathbb{P}(\mathcal{D})}$$

Pour classer le document $\mathcal{D}$, la classe avec la plus grande probabilité est choisie :

$$\underset{\mathcal{C}_i}{\text{ARGMAX}} \left\{ \frac{\mathbb{P}(\mathcal{C}_i)\mathbb{P}(\mathcal{D}|\mathcal{C}_i)}{\mathbb{P}(\mathcal{D})} \right\}$$

$\mathbb{P}(\mathcal{D})$ est généralement supprimé, car il est indépendant de la classe. $\mathbb{P}(\mathcal{C})$, $\mathbb{P}(\mathcal{D}|\mathcal{C})$, sont estimés en observant les données d'apprentissage. Cependant, l'estimation de $\mathbb{P}(\mathcal{D}|\mathcal{C})$ de cette manière est problématique, car il est très peu probable de voir le même document plusieurs fois : les données observées ne sont généralement pas suffisantes pour pouvoir générer des probabilités avec une certitude raisonnable. Le classificateur bayésien naïf tente de surmonter ce problème en simplifiant le modèle par l'hypothèse d'indépendance : tous les mots du document observé $\mathcal{D}$ sont conditionnellement indépendants les uns des autres étant donné la classe, cela revient à dire que $\forall(i,j), \mathbb{P}(\mathcal{D}_i|\mathcal{C} \wedge \mathcal{D}_j) = \mathbb{P}(\mathcal{D}_i|\mathcal{C})$. Les probabilités individuelles des mots d'un document sont estimées un par un plutôt que pour le document complet dans son ensemble. Cette hypothèse est très irréaliste, en effet, la langue est largement composée d'expression multimots qui mettent à mal cette hypothèse, elle donne le qualificatif « naïf » au classifieur. L'hypothèse d'indépendance conditionnelle est clairement violée dans les données du monde réel ; cependant, malgré ces violations, empiriquement, le classificateur bayésien naïf a de bonnes performances pour classer les documents textuels (Lewis and Ringuette, 1994; Billsus and Pazzani, 1997). Il reste ensuite à choisir une distribution de probabilités représentant la distribution des mots dans le document.

Usuellement, le modèle d'événement mutinomial (tenant compte du nombre d'occurrences) est préféré à un modèle multivarié de Bernoulli (binaire). Ce choix s'explique par de meilleures performances observées, en particulier sur de grands vocabulaires (McCallum et al., 1998). Il utilise la matrice de cooccurrences termes-documents pour calculer $\mathbb{P}(\mathcal{D}|\mathcal{C})$ de la manière suivante :

$$\mathbb{P}(\mathcal{D}|\mathcal{C}) = \mathbb{P}(\mathcal{C}) \prod_{\forall w \in Vocab(\mathcal{D})} \mathbb{P}(w|\mathcal{C}) A_{\mathcal{D},w}$$

Où est A la matrice de cooccurrence et $Vocab(\mathcal{D})$ le vocabulaire du document $\mathcal{D}$. Pour rendre les estimations de probabilité plus robustes en ce qui concerne les mots peu fréquents, une méthode de lissage est utilisée pour modifier les probabilités qui auraient été obtenues par simple comptage d'événements. Un effet important du lissage est qu'il évite d'attribuer des valeurs de probabilité égales à zéro à des mots qui n'apparaissent

pas dans les données d'apprentissage pour une classe particulière. Une méthode de lissage assez simple consiste à l'ajout de un à tous les comptes de mots pour une classe (on parle d'estimations courantes de Laplace). Une méthode plus intéressante est celle de Witten-Bell (Witten and Bell, 1991). Bien que les performances de la méthode bayésienne naïve ne soient pas aussi bonnes que celles d'autres méthodes d'apprentissage statistique telles que les classifieurs à plus proches voisins, il a été démontré qu'elle peut être étonnamment performante dans les tâches de classification où la probabilité calculée n'est pas importante (Domingos and Pazzani, 1997) ; paradoxalement la méthode bayésienne attribut souvent la bonne classe avec une estimation de probabilité assez mauvaise. Un autre avantage de l'approche bayésienne naïve est qu'elle est très efficace et très facile à mettre en œuvre par rapport aux autres méthodes d'apprentissage. L'approche bayésienne naïve a été utilisée par de nombreux systèmes de recommandation (De Gemmis et al., 2008; Degemmis et al., 2007; Mooney and Roy, 2000).

1.3.1.5 Réseaux de neurones et apprentissage profond

Les réseaux de neurones ont largement été utilisés pour la classification de texte et sont aujourd'hui l'état de l'art du domaine (Devlin et al., 2019; Radford et al., 2019). Dans le cas où l'on se restreint à un contexte dans lequel le nombre de données d'entraînement est suffisant pour envisager des modèles très profonds sans risque de sur-apprentissage, les approches basées sur des modèles d'attention et les réseaux récurrents semblent les architectures les plus performantes. Dans le cas contraire, les approches exploitant des modèles de langage pré-entraînés prédominent dans les usages, néanmoins elles ont été remarquées moins performante que des architectures de type bi-LSTM peu profonde dans ce contexte. En particulier, les approches utilisant des RNN dominées (Sutskever et al., 2011; Mandic and Chambers, 2001) avant l'arrivée des modèles d'attention pure telle que Transformer même dans le cas de gros corpus. Nous n'allons pas ici re-détailler ces architectures largement présentées en Section 1.2 ; cependant il faut garder en tête qu'elles constituent l'état de l'art de la classification de texte (Peng, 2020; Wang et al., 2020; Gampala et al., 2021).

Néanmoins, d'autres architectures ont également été testées comme les CNN (*Convolutionnal Neural Networks*) (Lai et al., 2015; LeCun et al., 2015, 1998). Un CNN est un réseau de neurones comportant des couches de convolutions, ces couches de convolutions peuvent être empilées pour fournir plusieurs filtres sur l'entrée. Pour réduire la complexité de calcul, les CNN utilisent des opérateurs d'agrégations (*pooling*) pour réduire la taille de

la sortie d'une couche à la suivante dans le réseau. Différents opérateurs d'agrégations sont utilisés pour réduire les sorties tout en préservant les caractéristiques importantes (Scherer et al., 2010). L'agrégation la plus courante est le maximum (*max pooling*), qui consiste à sélectionner l'élément maximal dans une fenêtre spatiale pour représenter cette fenêtre. Cela s'explique bien dans le cas des images pour lesquelles tout groupe de quatre pixels adjacents vont alors être représentés par le pixel de plus hautes valeurs, correspondant à une baisse globale de la définition. Afin d'alimenter la couche suivante avec la sortie agrégée, les sorties sont aplaties en une seule colonne. En général, les poids des noyaux de convolutions sont ajustés durant rétro-propagation. Un problème potentiel qui se pose lors de l'utilisation du CNN pour la classification de textes est le nombre de « canaux » (taille de l'espace des caractéristiques). Alors que les applications de classification d'images ont généralement peu de canaux (par exemple, seulement 3 canaux de représentations de l'image Rouge, Vert et Bleu, et par conséquent des vecteurs de taille 3 pour les pixels) résultant en d'excellente performance pour les CNNs, dans le cas du texte, le nombre de canaux peut être très grand (par exemple, 300 dimensions est une valeur usuelle pour un espace de plongements lexical) (Johnson and Zhang, 2015).

Une autre architecture efficace pour la classification de textes et de documents est l'attention hiérarchique (HAN). Cette technique a été introduite par Yang et al. (2016) et Seo et al. (2016) avec la particularité d'apprendre des modèles d'attentions de manière hiérarchique, d'abord sur les mots pour représenter les phrases, puis sur les phrases pour représenter les documents. Cette idée peut même être poussée plus loin en rajoutant des niveaux comme les caractères, les n-grammes, les paragraphes, les chapitres, etc. Elle est donc particulièrement intéressante lorsque les documents sont très segmentés, car elle apprend des représentations pour chaque niveau et de manière jointe.

Enfin, nombre d'approches cherchent à créer des architectures hybrides combinant plusieurs modules différents (CNN, RNN, FNN...). Parmi ces approches, les *réseaux neuronaux convolutions récurrents* (RCNN) cherchent à tirer profit à la fois de la nature séquentielle des RNN et spatiale des CNN (Lai et al., 2015; Wang et al., 2017). Classiquement, les RCNN capturent des informations contextuelles avec la structure récurrente et construisent la représentation du texte en utilisant un CNN (Lai et al., 2015). C-LSTM est une autre technique de classification de textes et de documents introduite par Zhou et al. (2015). C-LSTM combine CNN et LSTM afin d'apprendre des caractéristiques au niveau de la phrase en utilisant des couches convolutions. Cette architecture alimente des séquences de représentations de niveau supérieur dans le LSTM pour apprendre des

dépendances à long terme.

L'apprentissage profond est l'une des techniques les plus en vogue en intelligence artificielle, et de nombreux chercheurs et scientifiques se concentrent sur les architectures d'apprentissage profond pour améliorer la robustesse et la puissance de calcul de cet outil. Cependant, les architectures d'apprentissage profond présentent également certains inconvénients et limites lorsqu'elles sont appliquées à des tâches de classification. L'un des principaux problèmes de l'apprentissage profond est qu'il ne facilite pas une compréhension théorique complète de l'apprentissage (Shwartz-Ziv and Tishby, 2017). Un inconvénient bien connu des méthodes d'apprentissage direct est leur nature de « boîte noire » (Shrikumar et al., 2017). Cet effet boite noire est particulièrement attribué aux FNNs, néanmoins les architectures étant presque toujours hybrides ce problème impacte beaucoup d'approches. En outre, dans le cas du texte, il est également difficile d'interpréter les noyaux de convolutions. De plus, la visualisation des matrices d'attentions est souvent présentée comme permettant une meilleure interprétation du modèle, néanmoins, il a été démontré d'un point de vue théorique qu'une telle garantie n'existe pas (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Enfin, d'un point de vue pratique, les modèles d'attention très profonds en démultipliant le nombre de têtes et de couches d'attention sont rendues très difficilement interprétables, ainsi les matrices d'attention réellement interprétables ne sont plus l'exception que la norme. Enfin, les récentes avancés dans le domaine de l'interprétabilité se concentrent plus sur l'interpretabilité de la décision que du modèle et s'appliquent donc plus difficilement dans le cas de la recommandation (Ribeiro et al., 2016). Ce problème d'effet boite noire sera discuté plus en profondeur dans Section 1.4 lorsque nous traiterons des limitations des approches actuelles dans le cas de la recommandation à visée pédagogique. Une autre limite de l'apprentissage profond est qu'il nécessite généralement beaucoup plus de données que les algorithmes d'apprentissage automatique traditionnels, ce qui signifie que cette technique ne peut pas être appliquée à des tâches de classification sur de très petits ensembles de données (Anthes, 2013). De plus, la quantité massive de données nécessaires aux algorithmes de classification d'apprentissage profond en fond des modèles très gourmands en temps de calcul et en coût énergétique lors de leur entraînement (Severyn and Moschitti, 2015).

1.3.1.6 Conclusion sur la recommandation basée sur le contenu

Dans cette partie, nous avons fait un état des lieux des systèmes de recommandation basés sur le contenu, depuis les approches les plus simplificatrices jusqu'aux modèles

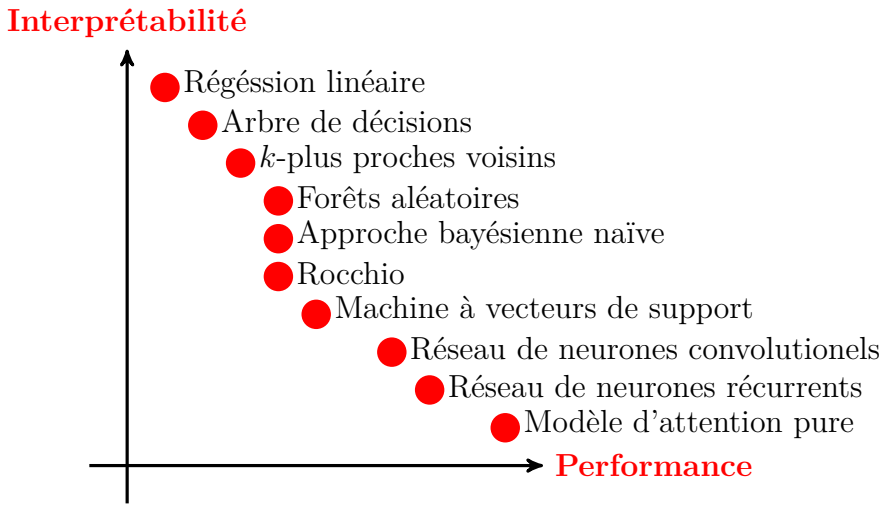


FIGURE 1.15 – Compromis entre performance et interprétabilité dans les modèles de classification de données textuelles. Figure inspirée de (Kowsari et al., 2019)

les plus gourmands en nombre de paramètres. Si les modèles les plus complexes sont aujourd'hui les plus performants, on pense en particulier aux modèles d'apprentissage profonds, ils sont aussi très difficilement interprétables. Cette tendance vers des modèles plus « boite noire » mais plus performants, est résumé dans la Figure 1.15.

Enfin, les systèmes de recommandation basés sur le contenu comportent une limitation importante dans leur usage. En raison de la nature de ce type de systèmes, ils ne peuvent recommander que des éléments qui ont un score élevé par rapport au profil de l'utilisateur. En d'autres termes, des éléments similaires à ceux déjà évalués. Cette lacune, appelée *sur-spécialisation*, empêche ces systèmes d'être utilisés efficacement dans des scénarios du monde réel où la sérendipité est souhaitée.

1.3.2 Approches basées sur le filtrage collaboratif

Les développeurs de l'un des premiers systèmes de recommandation (Goldberg et al., 1992) ont inventé l'expression « filtrage collaboratif », qui a été largement adoptée malgré le fait que les systèmes de recommandations peuvent ne pas collaborer explicitement avec les utilisateurs et que les recommandations peuvent suggérer des éléments particulièrement intéressants, en plus d'indiquer ceux qui devraient être filtrés (Resnick and Varian, 1997). L'hypothèse sous-jacente de l'approche du *filtrage collaboratif* est que si une personne A a la même opinion qu'une personne B sur une question, A est plus susceptible d'avoir l'opinion de B sur une question différente que celle d'une personne choisie au ha-

sard. Cela recoupe l'intuition qui voudrait que les gens obtiennent souvent les meilleures recommandations d'une personne ayant des goûts similaires aux leurs, typiquement leurs amis ou les membres de leurs groupes sociaux. Le filtrage collaboratif englobe des techniques permettant de faire correspondre des personnes ayant des intérêts similaires et de faire des recommandations sur cette base. Par exemple, dans le cas de la recommandation par filtrage collaboratif, les utilisateurs ayant dans le passé fréquemment « aimé » les mêmes ressources seront amenés à le faire encore dans le futur. Les approches de filtrage collaboratif cherchent à fournir des recommandations à partir du comportement passé d'un utilisateur ainsi que des décisions similaires prises par d'autres utilisateurs. Contrairement aux approches basées sur le contenu, la recommandation reçue par un utilisateur n'est donc pas indépendante du comportement des autres utilisateurs.

Un outil très fréquemment utilisé dans ce contexte est la *matrice utilisateur-ressources* qui répertorient les préférences de chaque utilisateur pour chaque ressource. Mathématiquement, c'est une matrice de taille $|utilisateurs| \times |ressources|$, en notant M cette matrice la case M_{ij} correspond à la préférence de l'utilisateur i sur la ressource j. Ainsi, une ligne de la matrice répertorie les préférences d'un utilisateur sur toutes les ressources et une colonne de la matrice les préférences de tous les utilisateurs sur une ressource. Ces préférences peuvent être obtenues de différentes manières avec une note explicite de l'utilisateur, ou de manière implicite en tenant compte d'indications implicites telles que l'engagement ($\frac{\text{durée de visionnage}}{\text{durée de la vidéo}}$), le taux de conversion ($(\frac{\text{nombre d'achats}}{\text{nombre d'accès}}$), le nombre de clics... Ces indications dépendent évidemment de l'application et ne sont pas toujours faciles à recueillir ; de plus, elles n'offrent pas la garantie de vraiment représenter les préférences de l'utilisateur comme le ferait une note explicite. Bien sûr, la plupart des cases de la matrice sont non renseignées. En effet, chaque utilisateur n'interagit qu'avec une fraction des ressources de la plateforme. Il est possible de voir la recommandation comme une généralisation du problème de la classification en considérant les cellules non renseignées de la matrice comme données de test et les cellules renseignés comme les données d'entraînements.

Les systèmes de recommandation par filtrage collaboratif existent sous trois architectures principales. La première est dite *basée sur un modèle*, la seconde, est dite *basée sur le voisinage* (on retrouve également la terminologie *memory-based* en anglais). Enfin, la troisième est dite *hybride* et cherche tout naturellement a combiné les point forts des deux approches précédentes. Dans la suite de cette section, nous allons présenter les grandes lignes des approches de recommandation par filtrage collaboratif. Un lecteur souhaitant

un état de l'art plus détaillé peut se référer à (Aggarwal, 2016b) pour les approches basées sur le voisinage et à (Aggarwal, 2016a) pour approches basées sur un modèle.

1.3.2.1 Approches de filtrage collaboratif basées sur le voisinage

Les approches basées sur le voisinage se décomposent en deux sous-familles d'approches, les *approches centrées sur les utilisateurs* vont chercher à trouver des utilisateurs similaires à l'utilisateur actif (celui à qui l'on veut proposer une recommandation) ; une fois fait, un score de préférence est attribué à chaque ressource sur la base des préférences observées chez les utilisateurs similaires. La deuxième sous-famille concerne les approches centrées contenues (on parle ici des approches de *filtrage collaboratif basées sur le voisinage centrées sur le contenu* et non des approches de recommandation basées contenu). L'idée ici est de déterminer un ensemble d'articles qui sont les plus similaires à l'article que l'on souhaite recommander. Ensuite, afin de prédire la note d'un utilisateur particulier pour cet article, les notes de l'utilisateur courant sur les ressources de cet ensemble sont utilisées. La moyenne pondérée de ces évaluations est utilisée pour calculer l'évaluation prédite de l'utilisateur pour la ressource. Une distinction importante entre les algorithmes de filtrage collaboratif basés sur les utilisateurs et les algorithmes de filtrage collaboratif basés sur le contenu est que les évaluations dans le premier cas sont prédites en utilisant les évaluations des utilisateurs voisins, tandis que les évaluations dans le second cas sont prédites en utilisant les évaluations de l'utilisateur sur les ressources voisines (c'est-à-dire étroitement liés). Dans le premier cas, les voisinages sont définis par les similarités entre les utilisateurs (lignes de la matrice utilisateur-ressources), alors que dans le second cas, les voisinages sont définis par les similarités entre les ressources (colonnes de la matrice utilisateur-ressources). Ainsi, les deux méthodes partagent une relation complémentaire et sont similaires d'un point de vue algorithmique, en effet, cela revient à travailler directement sur la matrice ou sur sa transposée. Les deux approches peuvent donc logiquement être définies dans un cadre unifié. Pour cette raison, les méthodes développées sont donc directement applicables dans les deux situations, pour cette raison dans la suite de cette section, nous présenterons les différentes approches indépendamment qu'elle soit centrée utilisateur ou contenu. Néanmoins, il existe des différences considérables dans les types de recommandations obtenues à l'aide de ces deux méthodes.

D'un point de vue purement algorithmique, le problème du filtrage collaboratif peut être vu comme une généralisation d'un problème de régression (dans le cas de préférences continues) ou de classification (dans le cas de préférences discrètes), pour cette raison les

approches basées voisinage sont vus comme une généralisation des classifieurs de type plus proche voisin dans la littérature de l'apprentissage automatique. En particulier, Bell and Koren (2007b) présente le problème comme une approximation heuristique des moindres carrés. Un tel cadre d'optimisation ouvre également la voie à l'intégration des méthodes de voisinage avec d'autres modèles d'optimisation.

Approches de filtrage collaboratif basées sur le voisinage centrées sur les utilisateurs

Dans les approches centrées utilisateurs, l'objectif est d'identifier les utilisateurs similaires à l'utilisateur cible pour qui l'on cherche à générer des recommandations. Dans le but de déterminer la note de l'utilisateur cible, il faut alors commencer par évaluer sa similarité aux autres utilisateurs. Par conséquent, une fonction de similarité est nécessaire entre les différents utilisateurs (représentés par leurs profils de notes sur les ressources). Le choix de la fonction de similarité est non trivial, car différents utilisateurs peuvent avoir des niveaux de notations différents ; certains utilisateurs ont par exemple consulté plus de ressources, d'autres ont tendance à apprécier de manière plus importante le contenu de la plateforme et ont donc une moyenne de notes plus élevée. Bien évidemment, les utilisateurs ont en plus toutes les chances de ne pas avoir noté les mêmes ressources. La fonction de similarité doit donc être capable de prendre en compte ces différents aspects.

Les fonctions de similarité les plus fréquemment utilisées sont le coefficient de corrélation linéaire de Pearson et la similarité cosinus (Herlocker et al., 2002). Dans le cas du coefficient de Pearson, des normalisations sont envisagées, par exemple les notes peuvent être centré-réduites pour chaque utilisateur (au niveau des lignes de la matrice). Différents mécanismes ont également proposé pour tenir mieux compte du problème de disparité ou ajuster l'impact des ressources les plus populaires (Voir Section 1.3.4.2).

Approches de filtrage collaboratif basées sur le voisinage centrées contenu

Dans les approches centrées sur le contenu, les voisinages sont construits en termes de ressources plutôt que d'utilisateur. Les similarités sont donc calculées entre ressources (colonnes de la matrice). Un des avantages des approches basées sur le contenu est la possibilité de fournir une explication concrète sur la recommandation proposée. Par exemple, Netflix fournit souvent des recommandations sous la forme : Comme vous avez aimé « Snatch » nous vous recommandons aussi « Arnaque crime et botanique »... À l'inverse, ce type d'explication est beaucoup plus dur à réaliser avec des approches centrées sur les

utilisateurs, parce que le groupe des voisins est souvent simplement un ensemble d'utilisateurs anonymes et n'est pas directement utilisable dans le processus de recommandation.

Les méthodes basées sur le voisinage sont toujours divisées en une phase hors ligne et une phase en ligne. Dans la phase hors ligne, les valeurs de similarité utilisateur-utilisateur (ou ressources-ressources) et les groupes de voisins des utilisateurs (ou ressources) sont calculés. Pour chaque utilisateur (ou ressources), le groupe de voisins pertinent est pré-stocké sur la base de ce calcul. Dans la phase en ligne, ces valeurs de similarité et ces groupes de voisins sont exploités pour faire des prédictions en utilisant des relations. Le principal inconvénient de ces méthodes est que la phase hors ligne peut parfois être peu pratique dans des environnements à grande échelle. La phase hors ligne de la méthode centrée sur les utilisateurs nécessite au moins $O(|utilisateurs|^2)$ de temps et d'espace. Cela peut parfois être trop lent lorsque le nombre d'utilisateurs ou de ressources dépasse la dizaine de millions et cela est d'autant plus problématique lorsque le modèle doit être remis à jour fréquemment comme c'est le cas pour la recommandation à large-échelle (souvent avec plusieurs mises à jour quotidiennes des modèles). Néanmoins, la phase en ligne des méthodes de voisinage est toujours efficace. L'autre principal inconvénient de ces méthodes est leur couverture limitée en raison de la disparité. Par exemple, si aucun des voisins les plus proches d'Alice n'a évalué « Vice-Versa », il n'est pas possible de fournir une prédiction d'évaluation de Vice-Versa pour Alice. D'autre part, dans la plupart des cas, nous ne nous intéressons qu'aux éléments les plus importants d'Alice. Si aucun des voisins les plus proches d'Alice n'a évalué « Vice-Versa », cela peut être une preuve que ce film n'est pas une bonne recommandation pour Alice. La disparité crée également des défis pour le calcul robuste de la similarité lorsque le nombre d'éléments mutuellement évalués entre deux utilisateurs est faible.

Méthodes de filtrage collaboratif basées *clustering*

Certaines méthodes se sont intéressées à réduire la complexité de la phase hors-ligne : une des idées consiste à créer des groupes (*clusters*) d'utilisateurs ou de ressources ; on parle de *méthodes basées clustering*. L'idée principale des méthodes basées sur le *clustering* est de remplacer la phase de calcul hors ligne des plus proches voisins par une phase de clustering hors ligne. Tout comme la phase de calcul hors ligne des plus proches voisins crée un grand nombre de groupes de voisins, qui sont centrés sur chaque cible possible, le processus de *clustering* crée un plus petit nombre de groupes de pairs (pairs est ici à prendre au sens d'un ensemble de personnes partageant un même *cluster*) qui ne sont pas

nécessairement centrés sur chaque cible possible. Le processus de *clustering* est beaucoup plus efficace que le temps $O(|utilisateurs|^2)$ nécessaire à la construction des groupes de pairs de chaque cible possible. Une fois les groupes construits, le processus de prédiction des notes est similaire à celui vu précédemment. La principale différence est uniquement que les pairs les plus proches du même groupe que celui de l'utilisateur cible sont utilisées pour effectuer la prédiction, ce qui rend l'approche beaucoup plus efficace. Cette efficacité se traduit par une certaine perte de précision, car l'ensemble des voisins les plus proches de chaque cible au sein d'un *cluster* est de moins bonne qualité que celui de l'ensemble des données. En outre, la finesse du *clustering* régit le compromis entre précision et efficacité. Lorsque les groupes sont à grain fin (comprendre suffisamment petit), la précision s'améliore, mais l'efficacité est réduite. Dans de nombreux cas, des gains d'efficacité très importants peuvent être obtenus pour de faibles réductions de la précision. Lorsque les matrices d'évaluation sont très grandes, cette approche offre une alternative très pratique à faible coût. L'une des difficultés liées à l'utilisation de cette approche est le fait que la matrice d'évaluation est incomplète. Par conséquent, les méthodes de *clustering* doivent être adaptées pour travailler avec des ensembles de données massivement incomplets. Dans ce contexte, l'algorithme des k-moyennes est souvent utilisé, car facilement adapté aux données incomplètes (Chee et al., 2001; Sarwar et al., 2002) parfois adjointe à une méthode de lissage des notes cherchant à combler les vides (Lin et al., 2007). Une autre approche s'est intéressée à un *clustering* aléatoire ou supervisé (O'Connor and Herlocker, 1999). D'autres approches encore ont modélisé le*clustering* sous la forme d'un problème d'optimisation (Xu et al., 2012).

Méthodes par réduction de dimensionalité

Les techniques de réduction de dimensionnalité sont utilisées dans les systèmes de recommandation basé voisinage en améliorant des modèles de voisinage préexistant (Bell et al., 2007; Bell and Koren, 2007b; Koren and Koren, 2008). Comme les méthodes par *clustering*, les méthodes par réduction de dimensionnalité réduisent le temps d'exécution de la phase hors-ligne en accélérant le processus de création des groupes de pairs. En effet, la plus faible dimension des vecteurs réduit le coût de calcul des similarités.

Dans (Bell et al., 2007), les auteurs fournissent un aperçu de la relation entre les méthodes de voisinage et les méthodes de régression en formulant les méthodes de voisinage comme des méthodes basées sur un modèle avec une formulation d'optimisation précise. Cette contribution est particulièrement importante, car de nombreuses autres méthodes

basées sur des modèles, telles que les modèles à variable latentes (nous discuterons de ces méthodes plus en détail en Section 1.3.2.2), peuvent également être exprimées sous forme de formulations d'optimisation. Cette observation ouvre la voie à la combinaison des méthodes de voisinage et des modèles à variables latents dans un cadre unifié (Koren and Koren, 2008) car il est désormais possible de combiner les deux fonctions objectifs. En tirant profit de cela, d'autres modèles basés sur la régression ont été proposés tels que les prédicteurs Slope-One (Lemire and Maclachlan, 2005), les méthodes des moindres carrés (Vucetic and Obradovic, 2005), les méthodes sous-contraintes (Meinshausen, 2013).

Méthodes basés sur des graphes

De nombreuses méthodes basées sur les graphes ont été proposées pour améliorer les algorithmes de filtrage collaboratif. La plupart de ces méthodes sont basées sur les *graphes utilisateur-ressources*, mais quelques-unes sont également basées sur les *graphes utilisateur-utilisateur*. Dans le premier cas, le graphe utilisateur-ressources peut se déduire directement de la matrice utilisateur-ressource, la matrice utilisateur-ressource est en fait la matrice d'adjacence du graphe. On obtient alors un graphe non orienté bipartite avec deux ensembles nœuds : les nœuds utilisateurs et les nœuds ressources. Le graphe est agencé de manière à ce qu'il n'existe pas de lien entre deux nœuds d'un même ensemble, tous les liens sont donc entre un nœud utilisateur et un nœud ressources. Les poids des liens sont attribués grâce aux valeurs de préférences des utilisateurs directement exprimées dans la matrice.

Dans le second cas, le graphe utilisateur-utilisateur est un graphe non orienté ; pour s'abstraire des ressources, une première étape consiste à calculer une matrice de similarité utilisateur-utilisateur. Pour cela, on peut utiliser nombre de mesures de similarité différente comme celle que nous avons vue précédemment. Une fois la matrice calculée, celle-ci sert de matrice d'adjacence et est considérée comme une représentation matricielle du graphe.

Une observation importante du point de vue des méthodes basées sur les graphes est qu'elles montrent une relation intéressante entre les problèmes d' *apprentissage d'ordre*, de recommandation et de *prédiction de liens*. La tâche de prédiction de liens est une tâche classique de la théorie des graphes qui consiste à prédire les liens manquants sur la base des liens préexistants dans le graphe (Liben-Nowell and Kleinberg, 2007). Il est relativement facile de modéliser le problème de recommandation comme un problème de prédictions : en pratique, la probabilité d'apparition d'un lien représenté par l'incertitude

du modèle de prédiction peut directement être interprété comme la probabilité de l'interaction utilisateur-ressources. En particulier, la méthode *ItemRank* proposée par Gori and Pucci (2007) montre comment utiliser directement les méthodes d'apprentissage d'ordre, et la méthode dans (Huang et al., 2005) montre comment utiliser directement les méthodes de prédiction de liens pour le filtrage collaboratif.

D'autres méthodes tirent profit des graphes pour générer les voisinages, en particulier les méthodes par marches aléatoires (Fouss et al., 2007; Yildirim and Krishnamoorthy, 2008), ou basées sur la mesure de Katz (Huang et al., 2004) (nombre de chemins élémentaires entre une paire de nœuds).

1.3.2.2 Approches basées sur des modèles

À l'inverse des méthodes basées voisinage, les méthodes basées sur un modèle résument les données en un modèle prédictif. Par conséquent, la phase d'entraînement est clairement séparée de la phase de prédiction. Les méthodes d'apprentissage automatique traditionnelle comme celle évoquée en Section 1.3.1 ont largement été adaptées au cas du filtrage collaboratif. Comme nous l'avons vu, cela s'explique par le fait que les problèmes traditionnels de classification et de régression sont des cas particuliers du problème de complétion de matrice (ou du filtrage collaboratif). Il est ainsi possible d'appliquer n'importe quel modèle de régression ou de classification sans adaptation spécifique. Nous n'allons pas faire ici une revue exhaustive des différentes approches, une revue plus complète peut être trouvé dans (Aggarwal, 2016a). Néanmoins, nous allons nous focaliser sur le cas particulier des modèles à variables latentes qui sont performants à l'état de l'art et proposent une nouvelle vision du problème.

Modèle à variables latentes

Nous l'avons vu en Section 1.3.2.1, les techniques de réduction de dimensionnalité peuvent être utilisées dans les systèmes de recommandation pour améliorer des modèles de voisinage préexistant. Néanmoins, il est également possible d'utiliser des modèles latents pour prédire les préférences sans s'appuyer sur des modèles de voisinage (Bell et al., 2007; Bell and Koren, 2007b; Hoffart et al., 2011; Koren and Koren, 2008; Koren et al., 2009; Mnih and Salakhutdinov, 2007) on parle alors de *modèle à variables latentes*. L'idée derrière ces méthodes est que la préférence d'un utilisateur à propos d'une ressource n'est pas intrinsèque, mais est le résultat d'une combinaison de préférences internes de l'utilisateur sur différents aspects d'une ressource, et de la présence dans la ressource de

ces différents aspects. Ce sont ces fameux aspects que l'on appelle les variables latentes. En suivant cette hypothèse, il convient donc de connaître la préférence d'un utilisateur sur les différents aspects qui font sa décision pour déduire sa préférence sur n'importe quelle ressource dont on connaît la distribution sur les aspects. Bien sûr, le raisonnement est symétrique : si l'on connaît exactement la distribution sur les aspects d'une ressource, on peut prédire la préférence qu'elle suscitera chez n'importe quel utilisateur dont on connaît la distribution de préférences sur ces aspects.

On peut présenter le problème de manière générique sous la forme d'un problème d'optimisation (Devooght et al., 2015) où l'on cherche à minimiser l'erreur sur les préférences observées du modèle latent :

$$\underset{W,H}{\text{ARGMAX}} \left\{ \sum_{i,j|r_{i,j} \in Observations} E(r_{i,j}, W_i H_j^T) + R(W,H) \right\} \tag{1.5}$$

Dans celui-ci, *Observations* est l'ensemble des observations de préférences d'un utilisateur i sur une ressource j directement extrait de la matrice utilisateur-ressources. Les matrices W et H que l'on cherche à estimer sont respectivement de taille $|utilisateurs| \times k$ et $k \times |ressources|$ avec k un méta-paramètre du modèle qui définit le nombre de variables latentes et donc la taille des vecteurs de représentation des ressources et des utilisateurs. Pour simplifier l'écriture, on note W_i la i-ème ligne de W correspondant à la représentation latente des préférences de l'utilisateur i et H_j^T la j-ème colonne de H correspondant à la représentation latente des préférences sur la ressource j. E est une fonction qui mesure l'erreur que le modèle latent fait sur les préférences observées, E est souvent une fonction d'erreur quadratique. R est une optionnelle fonction de régularisation permettant d'appliquer des contraintes sur les matrices de projections, les normalisations L1 et L2 sont fréquemment utilisées. Le produit WH^T une fois l'estimation réalisée correspond à une version complète approximée de la matrice utilisateur-ressources qui peuvent être utilisée pour la tâche de recommandation.

Ces méthodes sont notamment devenues très populaires eu égard à leurs performances durant le prix Netflix de 2007 (Bell and Koren, 2007a). En effet, en pratique, elles exploitent le fait que des portions significatives de lignes et de colonnes de matrices de données sont fortement corrélées. Par conséquent, les données ont des redondances internes et la matrice de données résultante est souvent très bien approchée par une matrice de bas rang. En raison des redondances inhérentes aux données, cette approximation de bas rang peut être déterminée même avec un petit sous-ensemble des entrées de la matrice

originale. Elle s'avère souvent une estimation robuste des entrées manquantes.

L'approche d'Aggarwal and Parthasarathy (2001) combine par exemple la technique d'espérance-maximisation (EM) avec la réduction de la dimensionnalité pour reconstruire les entrées de la matrice de données incomplètes. Différentes méthodes de réduction de dimensionnalité ont par la suite été envisagées. Chaque méthode propose un compromis différent entre qualité, risque de sur-apprentissage et interprétabilité. De la même manière, différentes contraintes d'optimisations et fonctions objectifs ont également été proposées : ajout de termes de normalisation sur l'optimisation (Paterek, 2007), factorisation à marge maximale (DeCoste, 2006; Weimer et al., 2007) en s'inspirant des machines à vecteurs de support, des formes de factorisation de matrices avec des contraintes de non-négativité (Hofmann, 2004; Zhang et al., 2006) permettant une meilleure interprétabilité de la matrice, des méthodes d'optimisations basées sur l'échantillonnage de Gibbs et permettant de décrire le problème dans un cadre bayésien (Langville et al., 2006), une pénalisation des retours d'informations les plus importants a également été envisagé en appliquant une pénalité proportionnelle à la norme Frobenius de la décomposition LU de la matrice (Devooght et al., 2015).

Approche hybride

La différence majeure entre les systèmes de recommandation basés sur le contenu et les systèmes basés sur le filtrage collaboratif réside dans le fait que les premiers n'utilisent que les traits extraits des utilisateurs ou des contenus pour faire leurs recommandations, tandis que les seconds n'utilisent que les corrélations entre les préférences des différents utilisateurs ou sur les différentes ressources pour faire leurs recommandations. Les deux ont des limitations : l'approche contenu n'exploite pas les corrélations entre les préférences des utilisateurs ou entre ressources ; le filtrage collaboratif ne tient pas compte explicitement du contenu, ni des informations de l'utilisateur. Des techniques hybrides ont donc été développées dans le but d'exploiter les forces des deux approches (De Campos et al., 2010; Godoy and Amandi, 2008). Dans son article, Burke (2002) définit une taxonomie de catégorisation des systèmes hybrides en sept classes. Nous allons ici détailler brièvement ces sept classes (une étude plus détaillée peut être trouvée dans (Çano and Morisio, 2017)) :

Mixage : Les hybrides mixtes représentent la forme la plus simple d'hybridation et sont raisonnables lorsqu'il est possible de réunir simultanément un grand nombre de recommandeurs différents sans nécessité d'ordonnancer les recommandations.

Ici, les listes de ressources générées par différents systèmes sont simplement mélangées pour produire une liste finale recommandée (Smyth and Cotter, 2000; Barragáns-Martínez et al., 2010). Les systèmes de recommandation hybrides mixtes sont simples et favorisent par nature une diversité dans la recommandation.

Pondération : Une autre approche d'hybridation simplificatrice consiste à agréger les scores de sortie des différentes approches grâce à une fonction linéaire. La pondération peut être à la charge des utilisateurs (De Campos et al., 2010), ou apprise au fil du temps (Yu et al., 2013). Elle peut également être la même pour toutes les ressources ou varier en fonction des informations spécifiques relatives à la ressource.

Cascade : Les recommandeurs hybrides en cascade sont des systèmes par niveaux, dans un premier temps un ou plusieurs systèmes sont utilisés pour générer une recommandation, dans un second temps un autre système cherche à raffiner ces recommandations. Un cas particulier de ces systèmes est le système à deux niveaux qui est aujourd'hui prédominant dans le domaine de la recommandation à large-échelle ; pour cette raison, nous étudierons ces systèmes en détail dans la Section 1.3.3. Dans les systèmes à deux niveaux que nous venons de mentionner, la deuxième étape consiste à réordonner les ressources candidates à la recommandation, elle est souvent prise en charge par des architectures neuronales profondes, néanmoins tous les systèmes hybrides en cascade ne sont pas organisés de cette manière (Kunaver et al., 2007; Lampropoulos et al., 2012; Chen et al., 2014).

Commutatif : Dans un hybride commutatif, le système passe d'une technique de recommandation à une autre en fonction de certains critères. La plupart du temps, ce critère est basé sur une mesure de distance ou de similarité (Billsus et al., 2000; Ghazanfar and Prugel-Bennett, 2010; Noguera et al., 2012). Un cas typique consiste à utiliser la commutation vers un système basé contenu lorsque qu'une stratégie de filtrage collaboratif ne retourne pas assez de résultats (Noguera et al., 2012). D'autres approches exploitent un modèle d'apprentissage pour prédire l'évaluation de la préférence, une fois le modèle entraîné, cette estimation est alors utilisée pour évaluer la nécessité d'une commutation (Ghazanfar and Prugel-Bennett, 2010).

Combinaison de traits : Une autre façon de réaliser la fusion contenu/collaboratif est de traiter les informations collaboratives comme de simples données de caractéristiques supplémentaires associées à chaque exemple et d'utiliser des techniques basées sur le contenu sur cet ensemble de données augmenté. Ce type d'hybridation traite la sortie d'un système de recommandation comme un trait pour un autre

système de recommandation (classiquement un système basé contenu qui exploite naturellement les traits) (Bedi et al., 2013; Yang and Hsu, 2010). La combinaison de traits hybrides permet au système de prendre en compte les données collaboratives sans s'y fier exclusivement, ce qui réduit la sensibilité du système au problème de la disparité (Voir Section 1.3.4.2). Inversement, elle permet au système d'exploiter des informations sur la similarité sémantique des ressources qui ne serait pas exploitée dans un système purement basé sur le filtrage collaboratif.

Augmentation de traits : Cette catégorie se rapproche de celle par augmentation des traits, en pratique, elle va chercher à exploiter les recommandations provenant d'autres systèmes pour produire sa propre recommandation, néanmoins, à la différence de la combinaison de traits, les différents systèmes ne nécessitent par de produire le même type de recommandation. À l'inverse, l'idée consistera à augmenter les traits représentant la ressource ou l'utilisateur avec des systèmes de recommandation auxiliaire, par exemple, un système chargé de recommander des auteurs similaires à celui de la ressource en question. Ce type d'hybridation traite la sortie d'un système de recommandation comme un trait pour un autre système de recommandation (classiquement un système basé contenu qui exploite naturellement les traits) (Bedi et al., 2013; Yang and Hsu, 2010). Le premier système a pour rôle de produire une liste de recommandations qui sera ensuite incorporée dans le processus d'un calcul du second système. À l'inverse des hybrides en cascade, ici les scores fournit par le premier système sont conservés dans le second et sont pris en compte dans le calcul de la liste recommandation finale.

Méta niveau : La dernière classe concerne les hybrides dits méta-niveau, une autre façon de combiner deux techniques de recommandation consiste à utiliser le modèle interne appris par l'une d'elles comme entrée pour une autre. Ainsi la représentation interne apprise par le premier système est utilisée comme ensemble de traits pour le second. L'avantage de l'hybride méta-niveau, en particulier pour le cas systèmes basé contenu/ système basé filtrage collaboratif, est que le modèle appris est une représentation comprimée de l'intérêt d'un utilisateur et le mécanisme collaboratif qui suit peut opérer sur cette représentation dense plus facilement que sur les données brutes des retours d'informations.

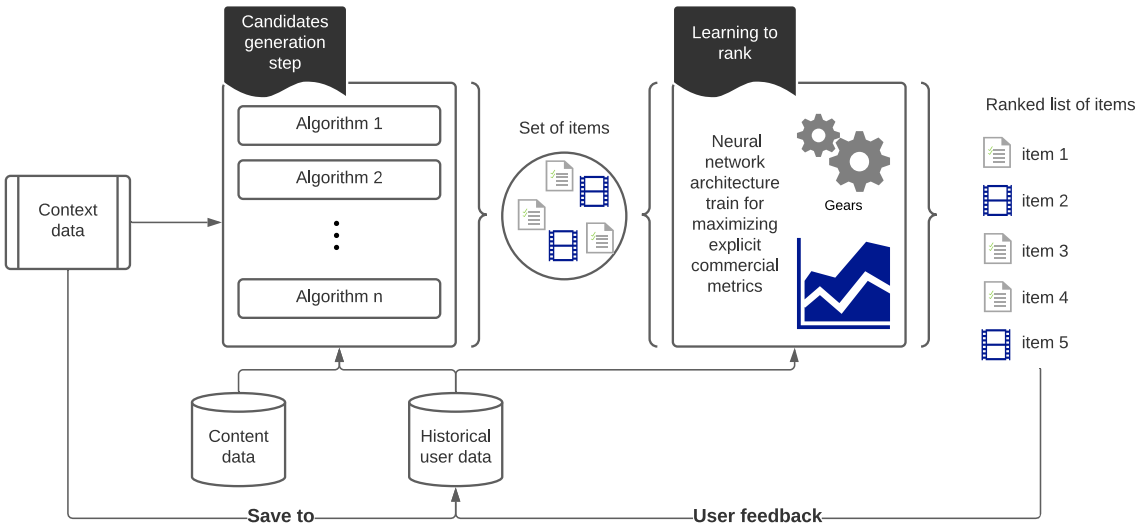


FIGURE 1.16 – Illustrations de l'architecture à deux niveaux des systèmes de recommandation grande échelle

1.3.3 Recommandation dans un contexte à large échelle

Les systèmes de recommandation contemporains ont pour mission de trouver un petit nombre d'éléments pertinents parmi des millions ou des milliards de candidats, personnalisés pour chacun des centaines de milliers ou des millions d'utilisateurs et leurs besoins en constante évolution, le tout devant se dérouler en quelques millisecondes afin de ne pas avoir d'impact négatif sur la vitesse de chargement des pages web (Hron et al., 2020). On parle de *recommandation à large-échelle*. L'une des solutions les plus répandues à ce problème est le système de recommandation à deux niveaux (Borisyuk et al., 2016; Covington et al., 2016; Eksombatchai et al., 2018) dans lequel (i) un ensemble de générateurs de candidats efficaces sur le plan informatique réduit la recherche de millions à seulement quelques centaines d'éléments, et (ii) un algorithme d'ordonnancement, plus lent, mais plus précis, sélectionne et réorganise les quelques éléments qui sont finalement proposés à l'utilisateur.

Les approches de recommandation à très large échelle sont aujourd'hui l'apanage des applications à usage commercial. Dans ce contexte, une architecture à deux niveaux est largement utilisée par la plupart des moteurs de recommandation à grande échelle, comme l'illustre l'examen de trois des plus grands sites webs d'e-business : Pinterest (Zhai et al., 2017), Facebook (He et al., 2014) et YouTube (Zhao et al., 2019).

1.3.3.1 Génération de candidats

Cette architecture à deux niveaux représentée en figure 1.16 est un cas particulier d'hybride en cascade, le premier niveau de cette architecture opèrent une présélection des ressources pertinentes (*candidates génération step* sur la figure). Pour cela, un ensemble non ordonné de candidats potentiels à la recommandation est généré, cette génération se fait en tenant compte à la fois des données contextuelles de la requête de recommandation, des données relatives au contenu actuel, et de l'ensemble des données historiques des utilisateurs sur la plateforme. Les systèmes ont souvent recours à différents algorithmes ; chacun de ces algorithmes capture un aspect de la similarité entre la ressource de la requête et la ressource candidate (Krichene et al., 2019; Covington et al., 2016). Il est important de noter que les générateurs de candidats sont souvent hétérogènes à la fois en termes de taille et d'approche, et de données utilisées pour sélectionner les candidats, allant de simples règles associatives à des réseaux neuronaux récurrents (Chen et al., 2019). Par exemple, un générateur de candidats peut se concentrer sur un ensemble plus restreint de caractéristiques (Yi et al., 2019), tandis que l'ordonnateur peut s'appuyer sur un modèle plus puissant et tenir compte de caractéristiques supplémentaires extraites, par exemple, des évaluations, des attributs spécialisés de l'utilisateur et de l'article, ou du nombre et du type d'interactions passées avec un utilisateur donné (Covington et al., 2016; Ma et al., 2020). Parmi les algorithmes les plus employés, on retrouve les méthodes basées contenu et les méthodes basées sur le filtrage collaboratif. Dans certains cas, il est possible que l'algorithme soit un modèle qui apprend directement de bout en bout depuis les retours qui seront donnés par les choix de l'utilisateur à travers le deuxième niveau du système (Rodriguez et al., 2020).

Le deuxième niveau de l'architecture cherche à ordonner les candidats proposés lors du premier niveau. Usuellement, le nombre de candidats générés dépasse largement l'ordre de la centaine. Pour comprendre à quel point l'étape d'ordonnancement s'avère cruciale, il est intéressant de constater que la visibilité d'une ressource candidat décroît exponentiellement avec son classement. En guise d'illustration de ce phénomène, selon une étude de Cornwell (Granka et al., 2004), les dix premiers résultats du moteur de recherche de Google concentrent plus de 99% des clics des utilisateurs, dont 40% pour le premier résultat. Pour cette raison, cette étape ordonnancement est la plus importante de l'architecture ; nous allons donc la discuter en détail dans la prochaine section.

1.3.3.2 Focus sur l'apprentissage d'ordre

L'apprentissage d'ordre ou *learning to rank* en anglais est un problème central dans le domaine de la recherche d'information. Les méthodes utilisées typiquement sont des méthodes d'apprentissage automatique, généralement supervisées, semi-supervisées ou par renforcement. Les données d'apprentissage consistent en des listes d'éléments avec un certain ordre partiel spécifié entre les éléments de chaque liste. Cet ordre est induit d'une fonction de coût dérivée du modèle commercial de l'application. C'est cette fonction de coût directement calculée à partir des retours d'informations implicites et explicites de l'utilisateur que l'on va chercher à maximiser. Le modèle d'ordonnancement a pour but d'ordonner, c'est-à-dire de produire une permutation des éléments dans les ensembles issue de l'étape de génération des candidats, d'une manière similaire aux ordonnancements dans les données d'apprentissage.

Les fonctions de coût utilisées comme vérité terrain sont la clé de voûte de ce type d'architecture : elles sont directement calculées à partir des retours des utilisateurs sur la plateforme.

Pour les plateformes de visionnage de contenu, la fonction de coût est directement calculée depuis les retours explicites : on parle d'*objectif de satisfaction* (mention, j'aime, commentaire, partage...) ou implicites : on parle d'*objectif d'engagement* des utilisateurs (clic sur la ressource, temps de consommation de la ressource...). Dans le cas de YouTube (Zhao et al., 2019) par exemple, le système de recommandation est une architecture neuronale très profonde [5] cherchant à prédire les différents retours réalisés par l'utilisateur sur la ressource qui va lui être recommandée. Parmi ces retours, on retrouve l'ensemble des interactions possibles d'un utilisateur qui peuvent être discrètes (la présence ou absence d'un « j'aime », d'un commentaire...) ou continues (durée de visionnage...). Logiquement, les retours discrets sont modélisés comme une tâche de classification et les retours continus comme une tâche de régression au cas par cas. De ce point de vue, le modèle neuronal dans son ensemble résout plusieurs tâches à la fois. Pour cette raison, on parle de modèles multitâches. Les différentes prédictions sont ensuite combinées à l'aide d'une couche linéaire définit manuellement et fournissent un score pour chaque ressource (on parlera de pertinence, mais il faut garder à l'esprit que cette pertinence est complètement de la fonction de coût et donc du modèle économique). YouTube ne dévoile pas la nature des pondérations engendrant ce score, mais précise que les poids relatifs aux différents types

5. Aucune information n'est disponible à ma connaissance sur la taille exacte, mais on peut très raisonnablement supposer une architecture dépassant largement le million de paramètres.

de retours. Finalement, il est important de noter que les retours d'informations implicites sont naturellement majoritaires par rapport aux retours d'informations explicites (typiquement, on observe une croissance logarithmique du nombre de « j'aime » par rapport au nombre de vues (Shoufan and Mohamed, 2017)), ils sont donc pour grande partie responsable des ressources recommandées.

Les autres types de plateforme proposent des approches similaires : dans le cas d'applications commerciales par exemple, le nombre de clics et d'achats se retrouve fréquemment au centre du calcul de la fonction de coût.

1.3.4 Les spécificités de la tâche de recommandation

La tâche de recommandation soulève des problématiques spécifiques qui font d'elle un problème singulier en recherche d'information. Dans cette section, nous allons présenter les principales difficultés liées à cette tâche.

1.3.4.1 Démarrage à froid

Nous l'avons vu en Section 1.3.2 les approches basées sur le filtrage collaboratif nécessitent des données historiques des interactions utilisateurs pour fournir une recommandation.

Lorsqu'une ressource est ajoutée au système ou qu'un nouvel utilisateur interagit sur une plateforme, le système de recommandation est confronté à un problème pour donner des recommandations. En effet, en l'absence de donnée historique sur le nouvel utilisateur ou la nouvelle ressource, le modèle se trouve dans l'impossibilité de pouvoir ajuster sa politique de recommandation. Dans la matrice utilisateur-ressources, une ligne ou une colonne est remplie de 0. Ce problème est appelé problème de *démarrage à froid* (*cold start problem*) (He et al., 2016; Lika et al., 2014; Nadimi-Shahraki and Bahadorpour, 2014). Dans la littérature, on distingue deux formes différentes de problème de démarrage à froid : (i) le problème du démarrage à froid d'un nouvel utilisateur (ii) le problème du démarrage à froid d'une nouvelle ressource. Dans le cas d'un problème de démarrage à froid d'un nouvel utilisateur, celui-ci est introduit dans le système, et le système est confronté à un problème de recommandation, car il n'a aucune information sur l'utilisateur. Dans le cas d'un problème de démarrage à froid d'une nouvelle ressource, le système ne dispose d'aucune évaluation pour la nouvelle ressource et il est dans l'impossibilité de déterminer un utilisateur cible pour cette ressource. Parmi les deux types de problèmes de démarrage

à froid, le problème de démarrage à froid d'un nouvel utilisateur est plus difficile et a été largement étudié (Bobadilla et al., 2012; Braunhofer et al., 2014; Lika et al., 2014). On remarque que dans le cas du démarrage à froid dû à une nouvelle ressource, un système de recommandation basé contenu n'est pas impacté, en effet, le choix est fait dans ces systèmes d'utiliser uniquement un profil utilisateur et une représentation sémantique de la ressource qui dans ce cas sont tous deux disponible. En revanche, dans le cas du démarrage à froid dû à un nouvel utilisateur, les systèmes basés contenu sont aussi impactés, en effet, le profil utilisateur utilisé dans ces approches exploite souvent les interactions historiques de l'utilisateur avec la plateforme qui dans ce cas n'existent pas. Pour ces deux types de démarrage à froid, nous parlerons de *démarrage à froid local*, dans le sens ou le problème se focalise autour d'un utilisateur ou d'une ressource.

De manière plus problématique, lors du premier lancement d'un système de recommandation, l'ensemble du système n'a connaissance d'aucune donnée historique des interactions des utilisateurs sur la plateforme. Par conséquent, la recommandation pour l'ensemble des utilisateurs risque d'être grandement impacté ; on parle de *démarrage à froid global* dans le sens où tout le système est impacté. Cela est d'autant plus problématique que les premières interactions des utilisateurs sur une plateforme sont critiques pour fidéliser les utilisateurs et augmenter la visibilité de la plateforme.

La principale difficulté du problème du démarrage à froid est la non-disponibilité des informations nécessaires à la formulation de recommandations. Les solutions proposées définissent des méthodes qui collectent ces informations non disponibles. L'information peut être collectée soit explicitement en interrogeant l'utilisateur, soit implicitement en utilisant des informations existantes. Ainsi, les solutions peuvent être classées en deux catégories en fonction de la manière dont elles collectent ces informations (Gope and Jain, 2017).

Collecte d'informations explicite

L'idée des méthodes explicites consiste à directement demander l'information de ses préférences à l'utilisateur, en suivant l'adage : « Une des meilleures façons de savoir quelque chose sur quelqu'un est de lui demander ». Cet adage fait l'hypothèse que l'utilisateur est à même de reconnaître lui-même le contenu qu'il trouvera le plus pertinent. Dans le cas réel, il est difficile de dire si cela est effectivement le cas en premier lieu, car le cerveau humain comporte de nombreux biais susceptibles d'altérer la collecte d'information, ensuite, car les intérêts de la plateforme et de l'utilisateur peuvent ne pas être complètement

alignés, notamment dans le cas d'applications commerciales enfin, car les utilisateurs ont tendance à ne pas aimer être sollicités sur leurs préférences et peuvent donc saboter la phase de profilage (He et al., 2016; Elahi et al., 2014). Les solutions explicites mettent en œuvre cette idée en interagissant directement avec l'utilisateur pour recueillir les informations manquantes. Dans cette approche, on demande à l'utilisateur de remplir un questionnaire ou d'évaluer des ressources données. Lorsque le système collecte des informations de manière explicite, il peut acquérir des informations plus pertinentes, car il peut contrôler ce qu'il demande. Cependant, l'un des défis de cette approche est que les utilisateurs sont souvent réticents à participer au processus d'interrogation en raison du temps et de l'effort requis (He et al., 2016; Elahi et al., 2014). La sélection d'un ensemble minimal de questions/ressources les plus informatifs est une façon de résoudre ce problème. L'utilisation de la visualisation (Yoo and Gretzel, 2011; Kagie et al., 2011) ou offrir des incitations sont d'autres alternatives. Les solutions explicites visent à recueillir les informations pertinentes adéquates sans accabler l'utilisateur. Deux techniques ont été largement utilisées : *l'apprentissage actif* et *les stratégies basées sur des questionnaires*.

L'objectif de l'apprentissage actif est de sélectionner un sous ensembles de ressources de taille minimal et pertinent pour caractériser l'utilisateur (Elahi et al., 2014). Pour cela, les approches utilisent différentes heuristiques pour la sélection des ressources, qui tentent d'optimiser certains aspects du système tels que l'augmentation de la précision, la réduction de l'incertitude, etc. Contrairement aux méthodes d'apprentissages classiques, le système sélectionne les exemples qui lui sont donnés et les adaptent durant son apprentissage (Rubens et al., 2015), d'où la présence de l'épithète actif en opposition aux apprentissages standards dits passifs. Cette sélection peut se faire de manière personnalisée (Braunhofer et al., 2014) ou être la même pour chaque nouvel utilisateur dans le système (Rubens et al., 2015; Elahi et al., 2014).

Dans les approches basées sur les questionnaires, les utilisateurs reçoivent une ressource parmi un ensemble de ressources et sont invités à donner leur avis. En général, il peut y avoir trois réponses : aimer, ne pas aimer ou ne pas savoir (Sun et al., 2013). Une liste initiale d'éléments de départ et un profil d'utilisateur approximatif sont préparés, qui sont affinés tout au long de l'entretien. Cela a conduit au développement d'un processus de questionnaire adaptatif dans lequel la liste d'éléments de départ est adaptée en fonction de la réponse de l'utilisateur. La plupart des solutions utilisent des arbres de décision pour modéliser les questionnaires adaptatifs (Zhou et al., 2011; Golbandi et al., 2011; Sun et al., 2013). Un bon entretien doit bien sûr être court et concis (Zhou et al., 2011; Sun et al.,

2013). Pour cela, il est nécessaire de sélectionner un minimum d'éléments, mais avec un maximum d'informations. Il existe plusieurs stratégies de sélection des éléments, chacune présentant des compromis différents entre deux aspects, à savoir la précision de l'information et l'effort de l'utilisateur (Rashid et al., 2002). Les stratégies aléatoires basées sur l'entropie pure en sont quelques exemples (Rashid et al., 2002). La combinaison de deux ou plusieurs stratégies est également utilisée. Le questionnaire prend la forme souvent la forme d'une séquence de ressource sur lesquelles il faut exprimer sa préférence (j'aime, je n'aime pas, je ne connais pas), la liste de ressource donnée à un utilisateur dépend alors de ces choix précédents et est déterminé par l'arbre de décision, les ressources sont usuellement données une à la fois chaque ressource correspondant à un nœud de l'arbre. Certains chercheurs suggèrent qu'au lieu de montrer un seul élément à la fois, il est préférable d'en montrer plusieurs chaque nœud correspond alors à une liste de ressource. Cela augmente les chances que l'utilisateur soit familier avec certains des éléments, ce qui lui permet de donner une note plus décisive (Golbandi et al., 2011). Les approches basées sur les entretiens (Sun et al., 2013) sont les meilleures stratégies pour les systèmes de recommandation conversationnel où l'utilisateur et le système interagissent tout au long du processus. Au lieu de demander l'avis des utilisateurs sur des ressources, (Christakopoulou et al., 2016) suggère de poser des questions générales, préparées au moyen d'une enquête, pour comprendre les préférences de l'utilisateur. Plusieurs chercheurs suggèrent que les stratégies d'apprentissage en ligne sont insuffisantes pour cerner les préférences de l'utilisateur (Zhou et al., 2011; Sun et al., 2013). En effet, les critères prédéterminés que l'heuristique apprentissage actif tente d'optimiser sont insuffisants pour sélectionner des questions suffisamment pertinentes lors de l'entretien (Zhou et al., 2011; Sun et al., 2013).

Collecte d'informations implicite

Les solutions favorisent une interaction minimale pour tenter de comprendre les préférences d'un nouvel utilisateur. Pour cela, elles exploitent souvent des informations existantes telles que les données démographiques ou des sources alternatives telles que les médias sociaux. Comme l'interaction est limitée, les utilisateurs se sentent plus à l'aise et sont moins enclins à partir. Parmi les méthodes implicites, beaucoup sont des adaptations des stratégies de filtrage collaboratif que nous avons présentées en 1.3.2. Elles ne sont donc pas utilisables dans le cas d'un départ à froid global. De plus, même dans le cas d'un démarrage à froid local, les stratégies de filtrage traditionnelles ont une portée limitée

en raison du manque d'informations. Une façon de les adapter au scénario du démarrage à froid est de les modifier pour qu'elles fonctionnent sur des données éparses (Bobadilla et al., 2012; Lika et al., 2014; Zhang et al., 2014).

1.3.4.2 Disparité

Le problème de la disparité survient lorsque les interactions des utilisateurs et des ressources en quantité ou en qualité ne sont pas suffisantes pour inférer de manière fiable la similarité entre les entités et pouvoir par conséquent recommander les ressources adéquates (Anand and Bharadwaj, 2011). La disparité est parfois considérée comme la difficulté principale de la tâche de recommandation (Anand and Bharadwaj, 2011). En particulier, les utilisateurs ayant interagi avec des ressources peu consultées (et donc sur lesquelles le système a peu de retour d'informations, typiquement des nouveaux utilisateurs ou une série de nouvelles ressources) se retrouvent isolés ; il est alors nécessaire d'être capable de calculer la similarité entre des utilisateurs qui ne partagent pas de connections directe (ressources similaires consultées). On parle de mesures de *similarités globales* en contraste avec les mesures classiques telles que le coefficient de corrélation de Pearson (que nous avons introduit en Section 1.3.2.1) considérées locales. De cette dénomination sont héritées les notions de *voisins locaux* et *voisins globaux*. Les approches de similarité globale cherchent typiquement à propager la similarité locale entre les utilisateurs.

Le problème de la disparité s'explique bien par la distribution empirique des retours d'informations entre les ressources ; dans un contexte réel, cette distribution a la propriété d'avoir une longue queue (*long-tail property* dans la littérature). Cela veut dire qu'uniquement une faible fraction des ressources est fréquemment consultée. Ces ressources que l'on qualifiera de populaires sont en nombre négligeable en comparaison avec la grande majorité des ressources qui ont elles très peu d'accès. Il en résulte une distribution très inégale des retours d'informations sous-jacents, une majorité des retours d'informations se concentre que quelques ressources, autrement seules quelques colonnes de la matrice utilisateurs-ressource ne sont pas quasi-intégralement rempli de 0. La figure 1.17 illustre cette distribution : dans le cas de YouTube, on observe par exemple que le nombre d'abonnements aux chaînes et donc le nombre d'utilisateurs qui les suivent suit cette distribution. Ainsi, on peut voir que la grande majorité des chaînes ont extrêmement peu d'abonnements quand une faible minorité concentre tous les abonnements. Une telle distribution a un impact important dans la recommandation : par exemple, dans le cas de l'e-commerce, les articles dans la tête de la distribution ont tendance à être des articles relativement

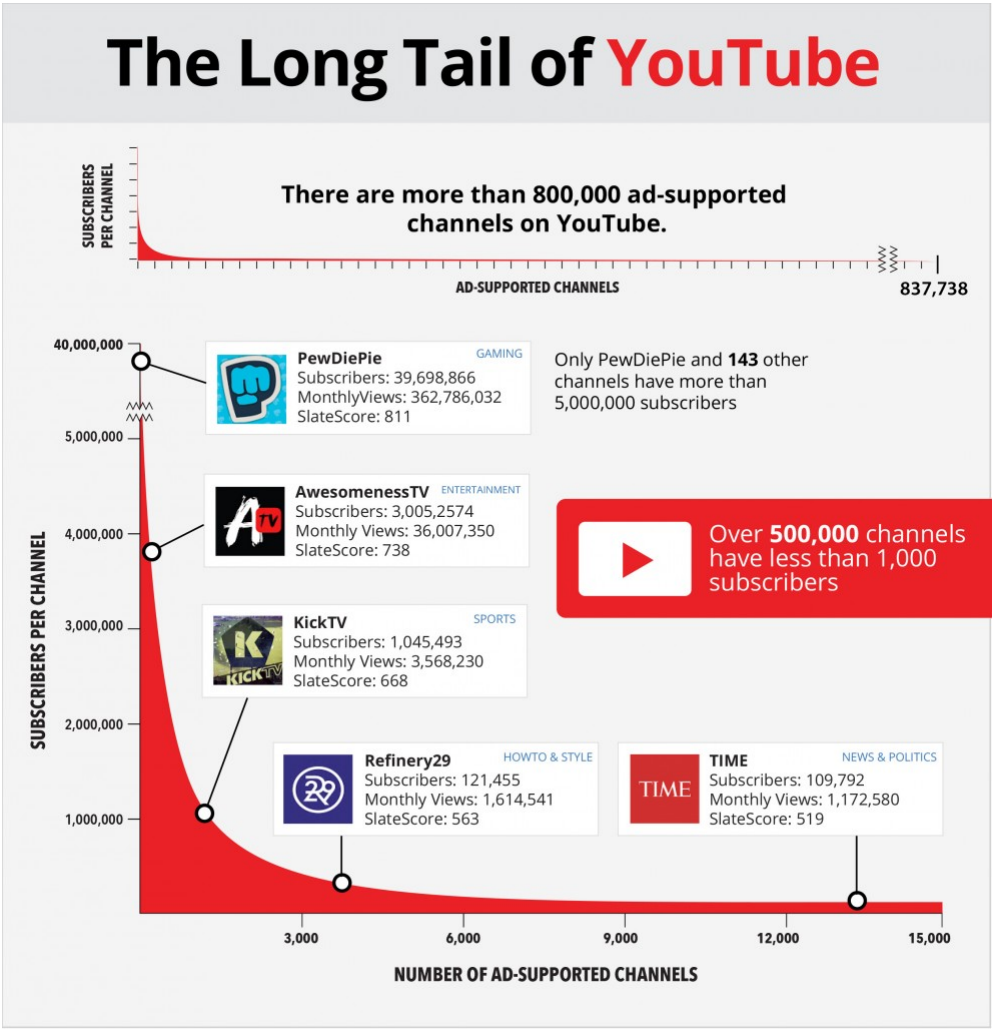


FIGURE 1.17 – Exemple de distribution à longue queue dans les abonnements aux chaînes YouTube. Source de l'image : Open Slate

compétitifs avec peu de profit pour le commerçant. D'un autre côté, les articles à faible fréquence ont des marges bénéficiaires plus importantes. Dans ce cas, il peut être avantageux pour le commerçant de recommander des articles à faible fréquence. En fait, une analyse suggère (Anderson, 2006) que de nombreuses entreprises, telles qu'Amazon.com, réalisent la plupart de leurs bénéfices en vendant des articles de la longue queue. Par ailleurs, en raison de la rareté des évaluations observées dans la longue queue, il est généralement plus difficile de fournir des prédictions d'évaluation robustes. En fait, de nombreux algorithmes de recommandation ont tendance à suggérer des articles populaires plutôt que des articles peu fréquents (Cremonesi et al., 2010). Ce phénomène a également un impact négatif sur la diversité et les utilisateurs peuvent souvent se lasser en recevant le même ensemble de recommandations. Enfin, la distribution à longue queue témoigne du fait que les éléments qui sont fréquemment évalués par les utilisateurs sont moins nombreux. Cela a des implications importantes pour les algorithmes de filtrage collaboratif basées sur le voisinage, car les voisinages sont souvent définis sur la base de ces éléments fréquemment évalués. Dans de nombreux cas, les évaluations de ces éléments à haute fréquence ne sont pas représentatives des éléments à basse fréquence en raison des différences inhérentes aux modèles d'évaluation de ces deux catégories d'éléments. Par conséquent, le processus de prédiction peut donner des résultats trompeurs.

1.3.4.3 Passage à l'échelle

En anglais, le terme *scalability* désigne la mesure de la capacité d'un système à augmenter ou à diminuer ses performances et son coût en réponse à des changements dans les demandes de traitement des applications et du système. Cette capacité est nécessaire pour tout système de recommandation visant une application à large-échelle ; à titre d'exemple, sur YouTube, plus de 300 heures de vidéo sont mises en ligne à chaque minute. Plus d'un milliard d'utilisateurs uniques consultent YouTube chaque mois, pour regarder plus de six milliards d'heures de vidéo, soit en moyenne près d'une heure par personne sur Terre. Chaque seconde, ce sont près de 43 000 vidéos qui sont visionnées, soit 1 460 milliards de vidéos par an[6]. Si YouTube constitue un cas extrême, la croissance tant en nombre d'utilisateurs, de ressources et d'interactions est le marqueur d'une tendance plus globale. Chaque jour, les utilisateurs sont plus nombreux, consultés individuellement plus de ressources et le panel disponible croît également. Cela engendre une croissance exponentielle pour les systèmes de recommandation en nombre de requêtes à traiter. Bien sûr, cette

6. https://www.YouTube.com/yt/press/fr/statistics.html

augmentation ne doit pas se traduire par une baisse des performances coté utilisateur ni en qualité ni en temps de réponse (on considère que le temps de réponse moyen doit être de l'ordre de la milliseconde) (Hron et al., 2020). Ainsi, la capacité de mise à l'échelle est un facteur clé pour déterminer le type de système de recommandation à utiliser. On appelle de tels systèmes des *systèmes évolutifs* et voici quelques éléments clés de la construction de ces systèmes évolutifs.

Une des premières idées consiste à effectuer le calcul des recommandations probables hors-ligne par blocs de données en exploitant la capacité de parallélisation des modèles. Les recommandations sont chargées dans un dictionnaire clé-valeur pour des requêtes à faible latence en temps réel. Cependant, il est pratiquement impossible de stocker toutes les requêtes possibles de tous les utilisateurs sur toutes les ressources ; ainsi, il est nécessaire d'être capable de cibler les requêtes les plus probables des utilisateurs afin de pré-calculer ces recommandations. Pour cette raison, cette méthode se limite généralement aux ressources très populaires. De manière intéressante, en fonction de l'architecture, il est possible de ne pas calculer complètement la recommandation, mais plutôt un résultat intermédiaire indépendant de l'utilisateur (Wu et al., 2014). Par exemple, dans une architecture neuronale, il est possible de pré-calculer les sous-blocs ne dépendant pas de données susceptibles de changer à cause des interactions de l'utilisateur, par exemple les parties de l'architecture n'exploitant que les données contenues.

Une autre idée consiste à optimiser l'entraînement des modèles ; tout d'abord, des méthodes d'échantillonnage opèrent en choisissant au hasard des ressources ou des utilisateurs, ou en supprimant les ressources sans engagement significatif de la part des utilisateurs. Cela provoque deux effets bénéfiques, une réduction du jeu de données et donc du temps d'entraînement. Dans ce contexte, les méthodes tenant compte de la disparité sont particulièrement indiquées, car elles permettent de réduire la complexité des calculs en conservant les performances (Linden et al., 2003).

Les architectures à deux niveaux ont également été développées dans le but d'avoir la meilleure évolutivité possible. En effet, la première phase permettant de sélectionner des centaines de candidats parmi des millions peut souvent être réalisée en amont, notamment quand les modèles de génération de candidats n'utilisent que les données liées à l'utilisateur, ou que les données liées à la ressource. Dans la deuxième phase, il est alors possible d'utiliser davantage d'informations pour faire l'ordonnancement des candidats. Malgré ces informations supplémentaires, le système tire profit du faible nombre de ressources restantes, ce qui permet à cette deuxième phase de s'opérer en ligne en garantissant un

temps de réponse rapide (Covington et al., 2016).

1.3.5 Les méthodes d'évaluations

L'évaluation est importante pour apprécier l'efficacité des algorithmes de recommandation. Pour mesurer l'efficacité des systèmes de recommandation et comparer différentes approches, il existe trois types d'évaluations : *les études d'utilisateurs*, *les évaluations en ligne* (tests A/B) et *les évaluations hors ligne* (Beel et al., 2013a). L'évaluation de la performance d'un algorithme de recommandation sur un ensemble de données de test fixe sera toujours extrêmement difficile, car il est impossible de prédire avec précision les réactions des utilisateurs réels aux recommandations.

Les études utilisateurs sont plutôt à petite échelle. On présente à quelques dizaines ou centaines d'utilisateurs des recommandations créées par différent algorithme de recommandation, puis les utilisateurs jugent quelles recommandations sont les meilleures. Pour cela, ils notent leur satisfaction quant à chacune des recommandations. L'algorithme ayant la meilleure note moyenne est considéré le meilleur. Dans ce type d'évaluation, on mesure uniquement la satisfaction de l'utilisateur au moment de la recommandation, mais pas la qualité moyenne de la recommandation dans le sens ou les utilisateurs ne connaissent pas au moment où il exprime leurs (in) satisfactions les autres ressources qui aurait pu être recommandées. En d'autres termes, on n'évalue plus la qualité de la recommandation que celle du système de recommandation. Ainsi, un système de recommandation, même idéal, pourra être mal évalué s'il doit fournir des réponses à des requêtes dont les résultats attendus par l'utilisateur ne sont pas disponibles dans la banque de ressource.

Dans les tests A/B, les recommandations sont présentées en ligne aux utilisateurs, et le système de recommandation choisit au hasard parmi au moins deux approches de recommandation différentes pour générer des recommandations. L'efficacité est mesurée à l'aveugle à l'aide de mesures implicites d'efficacité telles que le taux de conversion ou le taux de clics (Chopra, 2010). À l'inverse des études utilisateurs, les tests A/B évaluent les systèmes de manière implicite, néanmoins cette évaluation n'est que relative aux autres systèmes en concurrence, et elle ne constitue pas une évaluation intrinsèque d'un système. Il est tout à fait possible d'imaginer des violations d'inégalité triangulaire. Par exemple, en imaginant trois systèmes de recommandations A, B et C confrontés en duel, on pourrait observer que : $A > B$ et $A < C$, alors que $B > C$. Dans ce type de cas, il est très difficile de conclure sur le système à favoriser.

Les évaluations hors ligne sont basées sur des données historiques, par exemple un

ensemble de données qui contient des informations sur la façon dont les utilisateurs ont précédemment interagi. L'efficacité des approches de recommandation est alors mesurée en fonction de la capacité d'une approche de recommandation à prédire les évaluations des utilisateurs dans l'ensemble de données. Contrairement aux deux autres approches, ici, on évalue bien le système de recommandation. Cette évaluation n'est pas toujours une expression explicite de l'appréciation d'un utilisateur dépendamment du domaine. Par exemple, dans le domaine des systèmes de recommandation de ressources pédagogiques, les utilisateurs ne notent généralement pas une ressource recommandée. Dans de tels cas, les évaluations hors ligne peuvent utiliser des mesures implicites de l'efficacité.

La pratique la plus courante pour choisir un système de recommandation consiste à déterminer par des évaluations hors ligne l'algorithme ou les quelques algorithmes les plus performants (au sens de l'évaluation hors-ligne), et de valider ce(s) choix par des études d'utilisateurs ou des évaluations en ligne. Cette pratique est très critiquable Il a été démontré que les résultats des évaluations hors ligne ont une faible corrélation avec les résultats des études d'utilisateurs ou des tests A/B (Beel et al., 2013a; Turpin and Hersh, 2001). Fort souvent, les résultats des évaluations hors ligne ne sont pas en corrélation avec la satisfaction réelle des utilisateurs (Beel et al., 2013a). Les chercheurs ont conclu que les résultats des évaluations hors ligne doivent être considérés de manière très prudente (Roc'io et al., 2020). Et en particulier, car il a été démontré empiriquement que les évaluations hors ligne ne pouvaient pas prédire de manière fiable les performances de clics d'un algorithme en pratique. Il est intéressant de se pencher sur les raisons évoquées pour cette décorrélation : dans (Herlocker et al., 2004) les auteurs avancent que les facteurs humains sont fréquemment une variable explicative ignorée, par exemple, les études hors-ligne ou utilisateurs exposent usuellement les utilisateurs à la recommandation durant une durée relativement courte, elles ne permettent donc pas d'évaluer des phénomènes comme la lassitude d'un utilisateur face à l'absence de variété dans le contenu qui lui est recommandé. L'évaluation peut également largement différer en fonction de facteurs démographiques. Par exemple, les utilisateurs plus âgés ont tendance à être plus satisfaits des recommandations que les jeunes utilisateurs (Beel et al., 2013b). Un autre facteur évoqué est que les évaluations hors-ligne ne tienne pas compte la présentation du site ; ainsi, un moteur de recommandation meilleur, mais plus lent sera parfois perçu comme très mauvais par les utilisateurs agacés par la latence ou le temps d'attente, mais sera valorisé par des études hors ligne pour la qualité de sa recommandation (Herlocker et al., 2004).

Un autre problème vient de l'utilisation des retours d'informations implicites qui restent des indicateurs faillibles de l'appréciation d'un utilisateur En effet, une ressource ayant suscité une fort engagement n'est pas forcément une ressource qu'un utilisateur aurait évalué comme une bonne recommandation. Certains types de contenu ne sont pas exemple naturellement plus addictif pour l'utilisateur et risque donc de susciter plus d'engagement, même s'ils ont un pouvoir prédictif est liée à l'imperfection des ensembles de données hors ligne. Par construction, ces ensembles de données sont sous-optimaux, car ils ne contiennent que des recommandations pertinentes. Ce biais est dû à la méthode de recommandation en ligne en place (Cañamares and Castells, 2018), et aucune garantie ne prouve que ce biais à la même influence sur tous les algorithmes : en pratique, certains auteurs (Beel et al., 2013a) pensent qu'une corrélation entre les évaluations hors ligne et en ligne se produit quand les données sous-optimales ont le même effet sur tous les algorithmes évalués. À l'inverse, si l'ensemble de données sous-optimal a des effets différents sur deux algorithmes, l'évaluation hors ligne donne des résultats différents de ceux de l'évaluation en ligne.

De manière intéressante, le consensus semble se faire sur l'incapacité de prédire à l'avance si l'évaluation hors-ligne est pertinente, et en pratique, les experts de l'évaluation des systèmes de recommandations pensent que cela a des conséquences problématiques qui devraient conduire la communauté à arrêter les évaluations hors-ligne utilisant des retours d'informations indirects (en particulier le taux de clic) (Beel et al., 2013a).

1.3.6 Reproductibilité

Comme nous venons de le voir, les systèmes de recommandation sont notoirement difficiles à évaluer hors ligne. De plus, les évaluations en ligne sont souvent restreintes au contexte d'une plateforme donnée. À très large échelle, les plateformes ayant la capacité de conduire des comparaisons exhaustives des modèles sont des plateformes commerciales, pour lesquelles la publication des résultats peut aller à l'encontre de l'intérêt économique, eu égard à la concurrence économique entre les plateformes. Par ailleurs, les évaluations basées utilisateurs s'appliquent très mal à des contextes non spécifiques tant la variété des ressources, des usages et des utilisateurs rendent complexe la mise au point d'une évaluation utilisateur représentative.

Dans ce contexte, certains chercheurs affirment que cela a conduit à une crise de reproductibilité dans les publications sur les systèmes de recommandation. Plusieurs études (Dacrema et al., 2021; Rendle et al., 2020; Dacrema et al., 2019) récentes in-

terrogent le sens pris par la recherche dans ce domaine. À titre d'exemple (Dacrema et al., 2019) dans une étude de 2019 sur un petit nombre de publications sélectionnées appliquant l'apprentissage profond ou des méthodes neuronales au problème de la recommandation, publiés dans des conférences de premier plan (SIGIR, KDD, WWW, RecSys, IJCAI), a montré qu'en moyenne moins de 40% des articles pouvaient être reproduits par les auteurs de l'étude, avec seulement 14% dans certaines conférences. Dans l'ensemble, les études ont identifié 26 articles, dont 12 seulement ont pu être reproduits par les auteurs, et 11 d'entre eux ont pu être surclassés par des méthodes historiques bien plus anciennes et plus simples, correctement réglées, selon des mesures d'évaluation hors ligne.

Un des seuls points de convergence des différentes études est la supériorité pratique observée des méthodes neuronales, notamment lors de challenges de recommandation tels que WSDM ou RecSys Challenge, mais aussi dans de multiples applications commerciales. Ces challenges sont un outil de reproductibilité ; néanmoins ils sont peu nombreux et restreints à des domaines d'applications spécifiques. L'un des événements qui a dynamisé la recherche sur les systèmes de recommandation a été le prix Netflix. De 2006 à 2009, Netflix a sponsorisé un concours, offrant un grand prix de $1 000 000 à l'équipe capable de prendre un ensemble de données de plus de 100 millions d'évaluations de films et de fournir des recommandations 10% plus précises que celles proposées par le système de recommandation existant de la société. Le projet Netflix a apporté de nombreux avantages en termes d'évaluation et de reproductibilité ; néanmoins, des chercheurs ont pu identifier des utilisateurs individuels en faisant correspondre les ensembles de données avec les évaluations de films sur l'Internet Movie Database (Narayanan and Shmatikov, 2006). Cette brèche en termes de confidentialité a été suivi d'actions en justice et a marqué la fin du prix Netflix. Ce cas est symptomatique de la difficulté de la reproductibilité et de l'évaluation dans le domaine de la recommandation. Nous essaierons de mieux comprendre ces problématiques et de proposer des solutions dans le chapitre 2.

Le problème de reproductibilité dans le domaine est bien connu depuis les années 2010 au moins (Ekstrand et al., 2011) ; parmi les causes principales, on relève souvent l'incohérence dans les méthodes d'évaluation. En 2013, Konstan and Adomavicius (2013) ont utilisé le terme de « crise de reproductibilité » dans le domaine, considérant qu'une une grande partie de la recherche sur les systèmes de recommandation peut être considérée comme non reproductive. Plusieurs analyses comparatives plus récentes, notamment (Said and Bellogin, 2014) en 2014, (Dacrema et al., 2019) en 2019 et (Lv et al., 2021) en 2021, ont montré des incohérences dans les résultats, même lorsque les mêmes algorithmes et

ensembles de données étaient utilisés. En particulier dans leur article de 2019 intitulé : « Are We Really Making Much Progress ? A Worrying Analysis of Recent Neural Recommendation Approaches », Dacrema et al. considèrent que la technologie fait des progrès fantômes depuis 10 ans à cause de l'absence de cadre d'évaluation, de méthodes d'évaluations fiables et de jeu d'évaluations commun (Dacrema et al., 2019). En particulier, nombre d'approches neuronales gourmandes en paramètre et en temps de calcul peuvent être surpassés par des algorithmes conceptuellement et informatiquement plus simple. La chasse aux meilleures valeurs de précision domine les activités de recherches dans ce domaine, même s'il n'est pas évident que des valeurs de précision légèrement supérieures soient pertinentes en termes de valeur ajoutée pour les utilisateurs. En fait, il existe un certain nombre de travaux de recherche qui indiquent qu'une plus grande précision ne se traduit pas nécessairement par de meilleures recommandations reçues (Chen et al., 2017).

1.4 Applications sur des contenus pédagogiques

L'application des systèmes de recommandation dans des applications à visée pédagogique n'est pas une nouvelle idée ; néanmoins, une grande majorité des approches se sont concentrées sur de la recommandation dans les cas où les utilisateurs ou les ressources appartiennent à un groupe très restreint. Par les contraintes caractéristiques, on retrouve le fait de se limiter à un public très spécifique, par exemple les élèves d'une classe, ou uniquement une liste de ressources restreinte, par exemple une liste d'exercice de calcul mental provenant d'un corpus préalablement annoté par des enseignants, ou enfin, un contexte pédagogique très ciblé typiquement une activité pédagogique en présentiel lors d'une heure d'enseignement. Si ce type d'expériences est très utile pour comprendre les spécificités de la recommandation à visée pédagogique, ces expériences souffrent malheureusement de limitations en ce qui concerne le passage à l'échelle. Lorsque le contexte de recommandation devient non formel ; on pense notamment aux plateformes telles que les FLOT, les approches utilisées se rapprochent des méthodes présentées plus haut dans les applications commerciales. Néanmoins, même le contexte des FLOT est assez restrictif par rapport au contexte en jeu dans le projet X5GON. En effet, dans le cadre des FLOT, les cours sont structurés par les enseignants de manière uniforme et sont également typiquement pré-labellisés par catégories, difficulté, objectifs et prérequis. Par ailleurs, les utilisateurs de la plateforme ont un objectif clair qui est de participer à un cours. L'ensemble des utilisateurs sont donc des apprenants pour lesquels on peut suivre et évaluer

la progression grâce à des exercices adéquats fournit par les enseignants du cours. Dans le contexte, X5GON les ressources sont de nature et de finalité différente, de plus, elles sont non structurées entre elles. Enfin, les utilisateurs sont variés : apprenants, enseignants, institutions... Pour cette raison, le contexte X5GON se rapproche plus de celui de plateforme de visionnage de vidéos tel que YouTube que de celui des FLOT.

Le Tableau 1.1 résume les contraintes s'appliquant dans ces différents contextes de la recommandation. On remarque, que les contraintes X5GON sur le plan de la capacité de mise à l'échelle, du public visé et de l'organisation des données se rapproche beaucoup d'utilisations de la recommandation dans des contextes à commerciaux. Pour cette raison, du point de vue purement technique, les modèles utilisés dans les applications à usage commercial semblent tout indiqué dans le cas d'X5GON, par contre, du point de vue de la finalité de la recommandation, le projet X5GON partage avec les FLOT et les utilisations classiques de la recommandation pédagogique la volonté d'améliorer l'expérience d'apprentissage. Pour ces raisons, une question se pose quant à la possibilité d'utiliser les modèles en vigueur dans les applications commerciales pour une application à finalité pédagogique. Dans les sections qui suivent, nous allons présenter les limitations des modèles utilisés dans les contextes à usage commercial, et en particulier de l'architecture d'ordonnancement par maximisation de métrique dans le cas d'application ayant une visée pédagogique.

1.4.1 Limitations des approches par maximisation de métrique dans le cas de la recommandation à visée pédagogique

Comme nous l'avons vu dans la Section 1.3.3, dans les applications à usage commercial, l'objectif de la recommandation est souvent simple et est une conséquence directe du modèle commercial. En conséquence, l'objectif des systèmes de recommandation est de maximiser des paramètres quantitatifs tels que le temps de visionnage ou le taux de conversion. Comme nous allons le voir, la deuxième étape du système qui a pour rôle de réordonnancer les candidats pose un problème pour notre tâche, car elle suggère de pouvoir définir des fonctions objectifs qui reflètent l'expérience positive de l'apprentissage. En pratique, de telles fonctions objectifs sont très difficiles à définir, car des retours d'informations reflétant la qualité de l'expérience d'apprentissage sont très dures à obtenir. D'une part, car les retours d'informations implicites (durée de consultation, taux de

	Données	Public visé	Facteur d'échelle	Finalité de la recommandation
Contexte restreint	Peu de données très structurées (annotations spécifiques sur le contenu de chaque document)	Très ciblé (ex : quelques élèves)	Impossibilité de mise à l'échelle	Améliorer l'activité pédagogique en cours
FLOT	Beaucoup de données très structurées (annotations spécifiques au niveau du document : difficulté...)	Apprenant dans la plus grande diversité	Fonctionnement à large échelle	Enrôler les apprenants dans des cours
Application commerciale	Énormément de données faiblement structurées (annotations se limitant aux méta-données : type, durée...)	Tout le monde	Fonctionnement à très large échelle	Finalité commerciale
Contrainte X5GON	Énormément de données non structurées (possibilité d'absence d'information clés : titre...)	Toutes personnes ayant un intérêt pour une activité d'apprentissage	Fonctionnement à large échelle	Proposer une expérience d'apprentissage à travers des REL

TABLE 1.1 – Comparaisons des contraintes entre différents cas d'application de la recommandation

clics...) ne sont par trivialement corrélaient avec une expérience d'apprentissage positive. D'autres parts, car les retours d'informations explicites susceptibles de mieux capturer la qualité de l'expérience d'apprentissage sont perçus comme fastidieux par les utilisateurs et donc difficile à obtenir en quantité suffisante. Face à ce problème, un choix couramment employé est d'utiliser des fonctions objectifs proches de celle utilisée dans les applications commerciales 1.4.1. Dans cette section, nous allons montrer pourquoi ce choix se heurte à de nombreuses limitations dans le cas des applications à visée pédagogique.

1.4.1.1 Viralité des contenus

Certains effets secondaires néfastes de la maximisation de ces objectifs sont aujourd'hui discutés : en particulier, la maximisation de métrique d'engagement a été démontrée comme favorisant les contenus viraux, violents ou pseudoscientifiques (Allgaier, 2019; Rieder et al., 2018; Shapiro and Park, 2015).

En effet, dans un contexte d'économie de l'attention, ces algorithmes ont tendance à recommander les contenus les plus addictifs au lieu des contenus les plus qualitatifs. Selon Marc Zuckerberg, actionnaire majoritaire de Facebook, possédant un des moteurs de recommandation les plus importants de la planète en termes de requête par jour, les contenus extrêmes et sensationnalistes produisent plus d'engagement, et les systèmes de recommandations ont appris à exploiter ces biais humains bien connus en proposant de manière disproportionnée les contenus clivants ou conspirationnistes[7]. Les travaux de rétro-ingénierie sur l'algorithme de recommandation de YouTube mené par Guillaume Chaslot et l'association Algotransparency montrent que l'algorithme a également tendance à favoriser les contenus à caractère complotistes[8]. Ce constat a été validé par l'équipe de développeurs de YouTube[9].

1.4.1.2 Bulle de filtres

Un deuxième problème est celui des bulles de filtres (de l'anglais : filter bubble), ce concept développé par Pariser (Pariser, 2011) énonce que les algorithmes sélectionnent « discrètement » les contenus visibles par chaque internaute, en s'appuyant sur différentes données collectées sur lui. Chaque internaute accéderait à une version significativement

7. Source : Mark Zuckerberg's Blueprint on Governance, 2018
8. Source : `https://www.algotransparency.org/`
9. Source : Continuing our work to improve recommendations on YouTube By The YouTube Team

différente du web. Il serait installé dans une « bulle » unique, optimisée pour sa personnalité supposée. Cette bulle serait « in fine » construite à la fois par les algorithmes et par les choix de l'internaute (« amis » sur les réseaux sociaux, sources d'informations, etc.). Cet effet existe dans d'autres contextes et pour d'autres médias [10] et semble pour partie inhérente au comportement humain [11] ; à ce jour, la question de l'impact exact des algorithmes reste débattue.

Néanmoins, il reste consensuel que les algorithmes de recommandation actuels ne favorisent pas la diversité des contenus recommandés tant en termes de provenance, de sujet que de point de vue. Ce problème se rapproche de la sur-spécialisation bien connue dans le cas de la recommandation basé contenu, elle est un effet secondaire de la manière dont les systèmes basés sur le contenu recommandent de nouveaux éléments, où la note prédite d'un utilisateur pour un élément est élevée si cet élément est similaire à ceux qu'il aime. Par exemple, dans une application de recommandation de films, le système peut recommander à un utilisateur un film du même genre ou ayant les mêmes acteurs que les films déjà vus par cet utilisateur. De ce fait, le système peut ne pas recommander des éléments qui sont différents, néanmoins intéressants pour l'utilisateur. Les solutions proposées pour ce problème comprennent l'ajout d'un élément aléatoire (Sheth and Maes, 1993) ou le filtrage des éléments trop similaires (Billsus and Pazzani, 2000; Zhang et al., 2002).

Enfin, d'autres problèmes se posent lors de l'étape de génération des candidats. Dans l'état actuel de l'art, comme nous l'avons discuté plus haut, il existe classiquement deux idées principales pour construire la recommandation. La première, les modèles basés sur le contenu, sont basés sur la représentation textuelle des ressources. Les approches de l'état de l'art utilisant des représentations latentes des documents pour capturer les similarités entre leurs représentations textuelles. Cet espace latent de représentation n'est jamais simple à interpréter. Il en résulte qu'il devient très difficile d'expliquer la cause de représentation d'une recommandation.

10. Will Oremus, « Dans leur bulle (de filtres), les internautes ? Pas si simple… »
11. André Gunthert, Et si on arrêtait avec les bulles de filtre ?, « Les "bulles de filtres" : est-ce vraiment la faute d'Internet ? »

1.4.1.3 Reproduction des stéréotypes

Cela est d'autant plus problématique que plusieurs études ont pu observer des biais ethniques ou liés aux stéréotypes de genre directement dans les représentations de ces espaces latents, ou en sortie des algorithmes les exploitant (Caliskan et al., 2017; Bolukbasi et al., 2016; Feldman and Peake, 2021). Dans le cadre des représentations de mots et de documents, il a été démontré que les représentations sémantiques apprises par les algorithmes reproduisent des stéréotypes dérivés de large corpus d'apprentissage dont elles sont issues, indépendamment de la date d'écriture ou des types de textes (articles de journaux, Wikipédia, pages web, etc.) (Bolukbasi et al., 2016; Garg et al., 2018). À l'inverse, il n'y a aujourd'hui pas d'élément qui permette d'avancer que les algorithmes utilisés amplifient ou créent ces biais. Certaines méthodes se sont intéressées à réduire ces biais ou à les supprimer (Feldman and Peake, 2021; Hardt et al., 2016) mais à ce jour aucune méthode ne semble être capable de résoudre le problème, et l'efficacité des méthodes proposées est largement débattu dans la littérature (Gonen and Goldberg, 2019). Une étude parut dans Nature observe que le langage lui-même indépendament de la personne qui l'emploie, contient des empreintes récupérables et exactes de nos préjugés historiques, qu'ils soient moralement neutres comme à l'égard des insectes ou des fleurs, problématiques comme à l'égard de la race ou du sexe, ou même simplement véridiques, reflétant le *statu quo* de la répartition des sexes par rapport aux carrières ou aux prénoms (Caliskan et al., 2017). Le débat scientifique croise le débat social en ce point, plusieurs chercheurs argumentent que le fait de déterminer si nous devons viser la suppression du biais ou plutôt la transparence et la sensibilisation est une question qui dépasse largement le cadre pur de l'informatique (Nissim et al., 2020). Ces biais dans les modèles le langage sous-jacents ont largement été observés sur des tâches de plus haut-niveau [12] (Feldman and Peake, 2021), telles que la traduction avec laquelle des chercheurs ont constaté que Google Translate amplifiait les disparités entre les sexes (Prates et al., 2020). Les moteurs de recherche, où, Datta et al. (2014) ont constaté que le moteur de recherche Google affichait moins d'emplois bien rémunérés pour les femmes que pour les hommes. Dans le domaine de la santé, des algorithmes d'IA ont été déployés pour prévoir des soins préventifs et thérapeutiques personnalisés en fonction de facteurs génétiques et environnementaux. Une étude a montré que ces algorithmes proposaient des recommandations discriminatoires à l'égard des femmes en les diagnostiquant mal ou en ignorant les liens pertinents entre le sexe et les différences de santé (Cirillo et al., 2020). Plus largement, cette problématique semble

12. https://developers.googleblog.com/2018/04/text-embedding-models-contain-bias.html

s'étendre à tous les domaines d'application des technologies de l'apprentissage automatique, de la publicité en ligne (Sweeney, 2013) aux condamnations pénales (Zafar et al., 2017).

1.4.1.4 Méfiance des acteurs de l'éducation

Dans le contexte de l'éducation, ces limitations semblent particulièrement problématiques. Les acteurs de l'éducation étant particulièrement attentifs non seulement à la qualité des contenus qui doivent être proposés, mais aussi à l'enjeu éthique de cette recommandation, un climat de méfiance légitime s'installe chez les acteurs de l'éducation vis-à-vis de ces méthodes. De plus, dans le cadre de l'éducation ouverte, l'accent est mis sur la nécessaire diversité des contenus et des provenances ; contraintes qui, nous l'avons vu, est difficilement satisfaite par les modèles à usage commercial.

Enfin, les méthodes par maximisation de métrique sont fréquemment associés aux géants de l'industrie numérique qui sont parmi les principaux exploitant et développeur de ces modèles. Ils ont historiquement mauvaise presse dans le milieu du libre.

Le propos ici n'est pas de valider ou d'invalider la légitimité de cette méfiance, mais d'observer qu'elle peut être un frein considérable - dont il faut tenir compte- pour l'adoption d'algorithmes de recommandation appliqués aux contenues pédagogiques.

1.4.2 Recommandation dans le domaine de l'éducation

Le domaine de la recommandation de ressources pédagogique s'inscrit dans le domaine de recherche de deux communautés spécifiques : TEL et EDM. TEL provient de l'acronyme anglais « Techonologies to Enhance Learning » et s'intéressent à l'ensemble des apports de la technologie au monde de l'éducation et à l'apprentissage en particulier. EDM provient de l'acronyme anglais « Educational Data Mining » et s'intéresse à l'utilisation des données à des fins éducatives : ce domaine est largement porté par une conférence éponyme dans laquelle sont publiées des revues systématiques des avancées du domaine. La tâche de génération de recommandations à visée éducative est une des sous-tâches reconnue de ces domaines.

L'article (Urdaneta-Ponte et al., 2021) propose une revue systématique des systèmes de recommandation utilisés dans ce contexte. L'objectif de cette revue systématique est

d'obtenir une vue d'ensemble des systèmes de recommandation dans l'éducation, de ses domaines de travail, des éléments de recommandation et des techniques utilisées afin d'identifier les éventuelles lacunes, tout en fournissant un cadre d'orientation approprié pour les futures activités de recherche. Cette étude a été conduite en 2015 et 2020 sur 98 articles publiés en conférences avec revue par les pairs. L'étude constate que les approches utilisées dans les systèmes de recommandation favorisent le filtrage collaboratif, les approches basées sur le contenu et les approches hybrides, avec une tendance à utiliser l'apprentissage automatique depuis 2018. Avec une nette prédominance pour les premières cités, une grande majorité des approches s'intéressent à la recommandation de contenu pour les étudiants au niveau universitaire sur des plateformes d'éducation en ligne. En particulier, l'application à des populations cible différentes (enseignant, élève du primaire ou secondaire) est très peu étudiée. Enfin, et de manière très nette, l'ensemble des papiers évalués se limite à des contextes éducatifs particuliers ou restrictifs ; beaucoup d'entre eux proposent des approches non évolutives (sans garantie de passage à l'échelle), ou spécifiques à un domaine ou à un cas d'utilisation particulier, et souvent une combinaison des trois. Ces constats sont partagés par d'autres revues de littérature traitant le sujet (Cui et al., 2018; Drachsler et al., 2015, 2010).

De manière concomitante, une autre étude comparative de 2019 (Rodrigues et al., 2018) relève qu'un challenge pour le futur est la recommandation de ressources d'apprentissage dans un contexte d'apprentissage non formel. En particulier, dans leur étude sur 82 systèmes d'apprentissage profond dans le domaine EDM, les auteurs ne relèvent que 2 approches relevant de recommandation de contenu. Le premier (Abhinav et al., 2018) recommande des opportunités d'apprentissage aux étudiants en fonction du cours auquel ils se sont inscrits ou qu'ils ont terminé. Et le second (Wong, 2018) aborde le problème de la planification des programmes d'études grâce à un corpus de 10 ans de relevés de notes d'étudiants universitaires. Tous deux sont limités à un cadre éducationnel formel et suggèrent l'utilisation d'une grande quantité de données de retour spécifiques, ce qui rend difficile l'adaptation dans un cadre informel. En conclusion de sa revue, (Hernández-Blanco et al., 2019) souligne également la recommandation de l'apprentissage de ressources dans un cadre informel comme un défi pour l'avenir et souligne le manque d'ensembles de données librement disponibles pour relever ce défi.

Chapitre 2

Développements et retours d'expériences du projet X5gon

Dans ce chapitre, nous allons étudier plus en détail la problématique de la recommandation à visée pédagogique telle qu'envisagée dans le cadre projet X5gon ; en particulier, nous nous concentrerons sur la disponibilité de données contenu, d'utilisateurs et de jeux de données d'évaluation. Pour cela, nous commencerons par décrire l'écosystème des REL au début du projet en prenant garde de détailler les risques et les difficultés majeures identifiées par le projet. En partant de cette étude contextuelle préalable, nous présenterons le plan d'action proposé par X5gon. Nous détaillerons ensuite les travaux réalisés au sein du projet, et finirons par un ensemble de retours d'expérience permettant de mieux comprendre les problématiques en lien avec l'adoption des REL et des technologies numériques dans le cadre de l'éducation. La dernière section sera consacrée à des développements additionnels réalisés par la cellule nantaise après le projet ; ces développements viennent répondre à certains manques identifiés lors des conclusions du projet.

2.1 Contexte du lancement de X5gon

Le projet X5gon est un projet européen de type Horizon2020 lancé en septembre 2018 et terminé en août 2021, qui ambitionnait de créer un réseau de RELs permettant de connecter les RELs, les apprenants et les différents acteurs de l'éducation au sein d'un écosystème promouvant les pratiques de l'éducation ouverte. Dans cette section, nous allons nous intéresser au contexte lors du lancement du projet X5gon.

2.1.1 Objectifs du projet

Dès le début du projet, trois objectifs ont clairement été identifiés :

— connecter les RELs,
— mobiliser les apprenants sur les REL en exploitant des technologies d'intelligence artificielle,
— stimuler la recherche et l'engouement autour des RELs et l'éducation ouverte.

Une des tâches découlant de ces objectifs concerne la recommandation à visée pédagogique dans un contexte d'apprentissage non formel. En dehors des spécificités scientifiques de cette tâche que nous avons étudiée dans l'état de l'art, certaines contraintes pratiques, notamment en termes de disponibilité des données, sont à considérer :

Disponibilités des ressources : La première concerne la disponibilité de contenus à recommander. Dans le cadre du projet, X5GON ces ressources étaient des REL ; être capable d'agglomérer un corpus suffisamment important et varié de RELs constituait un objectif important. Ce corpus devait pouvoir satisfaire les engagements du projet X5GON en exposant une variété d'origines culturelles, de modalités, de langues et de domaines (Atkins et al., 2007). Plus généralement, il était important lors de la création du corpus de suivre les recommandations faites lors du *plan d'action de Ljubljana sur les ressources éducatives libres* [1] en 2017, qui traitaient en détail de cette nécessité de diversité dans l'éducation ouverte. Par ailleurs, Ibrahim et al. (2018) ont montré que l'intégration de données provenant de multiples sources hétérogènes aident les systèmes à fournir de meilleures recommandations. Enfin, le corpus ainsi créé devait pouvoir constituer un ensemble de ressources suffisant pour répondre à deux contraintes important : i) la possibilité de tester le passage à l'échelle des approches, ii) la possibilité d'utiliser des approches gourmandes en données, notamment pour la tâche de recommandation.

Traces d'apprentissage pour la personnalisation : Nous l'avons vu dans l'état de l'art : les modèles de recommandation exploitent des données d'interactions historiques des utilisateurs pour fournir des recommandations personnalisées à chaque utilisateur (Zhao et al., 2019). L'exploitation de ces données en grande quantité constitue un aspect central pour les algorithmes de recommandation à large échelle. Dans le contexte du projet X5GON, la nécessité de personnalisation constituait un enjeu important ; pour cette raison, la disponibilité de données d'interactions permettant de prédire les préférences des utilisateurs constituait une nécessité dans le projet. Un jeu de données permettant de représenter les interactions des utilisateurs sur des RELs et exhibant une diversité tant au niveau des utilisateurs, des

1. https://www.oercongress.org/fr/woerc-actionplan/

ressources et des pratiques semblaient un jeu de donnée idéal pour la tâche qui nous concernait. Un tel jeu de données aurait permis également des évaluations hors ligne des différents systèmes. Pour garantir la qualité de ces évaluations hors-lignes, la prérogative de la littérature et en particulier de (Beel et al., 2013a) (voir Section 1.3.5) est d'utiliser un jeu de données - qui est un sous-ensemble des cas réels - qui soit représentatif de ces dits cas réels. Au vu de la large variété des cas de recommandation réel envisageable, pour atteindre un jeu de données représentatif, il était nécessaire de récolter un grand nombre de données.

Données provenant d'études utilisateurs : Comme nous l'avons vu dans la Section 1.3.5, l'évaluation de la recommandation est une tâche complexe ; les évaluations hors-ligne, bien que largement utilisées, ne constituent pas une évaluation suffisante pour la qualité d'un système de recommandation et peuvent être sujettes à de nombreux biais. Pour évaluer la qualité d'une recommandation, le consensus vise –lorsque cela est possible– à combiner différentes méthodes d'évaluation, en particulier les études utilisateurs et les méthodes d'A/B test (Section 1.3.5). Chacune de ces deux approches présente des avantages et nécessite un environnement d'expérimentation différent. Pour le cas des études utilisateurs, la littérature distingue lorsque que cela est possible deux catégories d'évaluation (Beel et al., 2013a) : une par des utilisateurs provenant d'un panel représentatif des utilisateurs finaux, l'autre par des experts. Dans les deux cas, les utilisateurs/experts sont invités à exprimer leurs préférences entre des listes de recommandations produites par différents systèmes. La disponibilité de données ou la mise en place de telles études fût également un facteur clé dans le contexte d'X5GON ; en effet, dans un contexte éducatif, la satisfaction de l'apprenant est primordiale. De plus, les enseignants possèdent une expertise quant à l'enseignement nécessaire dans une certaine discipline pour un certain niveau. Recueillir la satisfaction des apprenants et l'expertise pédagogique des enseignants peut largement contribuer à améliorer les systèmes. Un système d'apprentissage idéal doit à la fois satisfaire les apprenants et proposer un parcours d'apprentissage avec des connaissances reconnues par des experts.

Environnement d'A/B test : Le principal défaut des études utilisateurs est leur coût. En effet, les études utilisateurs nécessitent du temps humain pour recueillir les préférences. Ce temps est une ressource finie et limitée, en particulier lorsque l'on cherche à agglomérer des avis d'experts. Même lorsque les préférences sont recueillies indirectement, en utilisant des études préalables par exemple, cette li-

mitation temporelle engendre une restriction sur la taille des jeux de données. De tels jeux de données ne suffisent donc pas à valider l'ensemble des domaines, des pratiques et plus largement de la diversité que l'on peut observer dans l'écosystème REL. Ces données sont précieuses et il est important de limiter leur usage pour la validation des systèmes les plus avancés. Même si l'enjeu d'évaluation des systèmes de recommandation est complexe, la pratique actuelle qui semble émerger consiste à effectuer un premier filtrage en tirant profit des données historiques des interactions sur la plate-forme, effectuer un deuxième filtrage entre les systèmes avec des évaluations de type A/B et enfin valider avec des expériences utilisateurs. Nous avons jusqu'ici traité la première et la troisième étape, ces deux étapes ont l'avantage de pouvoir être exprimées en termes de jeu de données. La deuxième étape qui concerne l'A/B test ne se restreint pas à un jeu de données, mais concerne l'évaluation concurrente d'au minimum deux systèmes lors d'une expérimentation en ligne du système de recommandation. Dans notre contexte, cette étape nécessitait donc la mise en place d'un environnement permettant ce type d'évaluation.

Maintenant, que nous avons clairement déterminé les objectifs du projet et en particulier les données critiques à considérer pour la tâche de recommandation, nous allons nous intéresser à différents aspects du contexte du lancement du projet qui ont impacté sur les choix stratégiques et techniques mis en œuvre par la suite.

2.1.2 Les REL : un éco-système éclaté

Une des caractéristiques principales de l'écosystème REL au début du projet concerne l'éparpillement des RELs. En 2018, on constatait plus de 9 millions de sites web hébergeant des RELs ; et malgré cela, il restait difficile d'évaluer le nombre de ressources hébergées sur chaque site. Néanmoins, en se référant à une étude du site https://stateof.creativecommons.org/, pour l'ensemble des contenus sous licence Creative Commons, les trois plus gros hébergeurs de ressources identifiés sont Flickr (415.1 millions d'images), Youtube (49 millions de vidéos) et Wikipédia (46.7 millions d'articles) [2]. Parmi les 15 plus gros hébergeurs, on retrouvait de nombreuses plateformes commerciales (qui représentaient le plus gros de l'effectif $\frac{10}{15}$) sur lesquelles les contenus ouverts et non ouverts cohabitaient. Par exemple : Youtube la plate-forme américaine de partage de vidéos en

2. Le lecteur peut se référer à https://stateof.creativecommons.org/ pour une liste plus exhaustive

ligne, Flickr, la plate-forme américaine spécialisée dans l'hébergement d'images et de vidéos ou encore Medium, la plate-forme web d'hébergement de billets de blog. On retrouvait également des projets collaboratifs avec en tête Wikimedia Commons et Wikipédia, ou institutionnels notamment Europeana, un portail web créé par l'Union européenne qui contient les collections numérisées du patrimoine culturel de plus de 3 000 institutions en Europe. De plus, la plupart de ces plateformes acceptent peu de pluralité dans les formats hébergés : par exemple, Youtube se limite à du contenu vidéo, Wikipédia à des articles ou Flickr à des images. Au niveau des critères de qualité, ils peuvent être très variables de site en site, depuis DOAJ (Directory of Open Access Journals) qui n'accepte que des travaux scientifiques publiés à l'occasion de conférences ou dans des journaux reconnus pratiquant la revue par les pairs, en passant par Wikipédia qui s'appuie sur un système de validation communautaire des informations jusqu'à Youtube qui acceptent plus largement tout type de contenu vidéo et potentiellement même du contenu de désinformation.

Si cette étude est éclairante pour comprendre la diversité des hébergeurs de RELs, il est notable de constater qu'elle s'intéresse à tout type de contenus sous licence Creative Commons, qu'il soit éducatif ou non. Mais aussi qu'elle ne témoigne que du cas des licences Creative Commons et omet donc les autres licences libres. Enfin, elle ne fait aucune distinction entre les types de contenus : ainsi une vidéo a autant de valeur qu'une image, qu'un billet de blog ou qu'un article scientifique dans l'évaluation de l'importance de chacun des hébergeurs.

En plus des hébergeurs mentionnés par l'étude, on peut raisonnablement imaginer parmi les hébergeurs principaux les plateformes de type FLOT (ex : Coursera, Brilliant ...) et les universités (à l'image du MITopenCourseware) (Amiel and Soares, 2016).

Enfin, un très grand nombre de ressources sont hébergées sur de petits sites appartenant à des particuliers : il est difficile d'évaluer le nombre de ressources concernées par ce modèle d'hébergement ; néanmoins, on peut raisonnablement penser qu'il constitue une part importante des ressources tant la pratique semble courante.

Un point important à retenir sur l'éparpillement des ressources est qu'il ne semble pas suffisant de connecter quelques sites pour créer l'ensemble du réseau REL ; néanmoins, connecter quelques sites peut être suffisant pour attirer un grand nombre d'utilisateurs. De la même manière, pour obtenir la plus grande diversité possible, il semblait important de connecter les grands comme les petits acteurs. Cela est d'autant plus marqué que la grande majorité des ressources que l'on peut retrouver sur ces sites sont en anglais ; les autres langues fortement dotées sont l'espagnol, le portugais, l'allemand et le français.

Être capable de connecter également de plus petit hébergeur semble clé pour offrir une meilleure diversité de langues et de cultures (Atkins et al., 2007).

La langue n'est pas le seul aspect de la diversité qui bénéficierait largement d'une connexion plus large des sites ; on remarque également que nombre de sites dédiés à l'apprentissage fournissent des ressources qui conviennent à des apprenants adultes et assez fréquemment avec des pré-requis en termes de niveau équivalent à un cursus universitaire. Cela s'explique par la part importante prise par les FLOT dans cet écosystème ; en effet, la plupart des cours proposés sur les FLOT sont directement rédigés par des universitaires. Un autre facteur concerne les initiatives en faveur de la science ouverte, et en particulier les archives ouvertes, initiative historique des mouvements en faveur du libre accès des connaissances. Ils ont favorisé la création de dépôt d'articles scientifiques en libre accès. À l'inverse, dans l'enseignement pré-universitaire, la tradition confinait à l'usage de manuels et beaucoup moins aux partages des pratiques : ainsi, en l'absence d'initiative collective, on peut penser que les enseignants cherchant tout de même à partager leurs contenus ont dû trouver des alternatives individuelles ou de moindre ampleur comme des sites personnels, des blogs... Là encore, pour obtenir une diversité de niveau dans les REL disponibles, il semble nécessaire d'avoir un réseau mélangeant grands et petits acteurs.

2.1.3 Nouveauté de la problématique de la recommandation pédagogique

Un autre aspect important du lancement du projet concerne l'avancement de la recherche dans le domaine de la recommandation à visée pédagogique. D'après différentes revues de la littérature (Hernández-Blanco et al., 2019; Drachsler et al., 2015) la grande majorité des travaux se sont concentrés : (i) sur l'application de modèles issus de la recommandation commerciale dans un contexte pédagogique, (ii) sur des applications à destination d'un public de niveau universitaire, (iii) sur des approches dans des cadres de recommandations formels. L'application directe des modèles issus de la recommandation dans le cas d'application à visée éducative soulève des questionnements éthiques : comme nous l'avons vu en Section 1.4.1 cet aspect engendre un climat de méfiance chez les acteurs de l'éducation. Le cadre du projet X5GON est lui un contexte de recommandation lors d'activités d'apprentissage non formelle avec des données non structurées ambitionnant un fonctionnement à large-échelle. Le contexte d'apprentissage non formel implique la nécessité de recommander à tout type d'apprenant, par ailleurs, la volonté du projet de co-créer

avec les acteurs de l'éducation confine également à la plus grande transparence dans les méthodes de recommandation. Pour cela, la recommandation dans ce projet constituait donc un cas singulier par rapport aux approches précédentes.

2.1.4 Données utilisateurs et protection de la vie privée

Un aspect important pour le succès du projet réside en la capacité de collecter ou de récupérer des traces d'apprentissages, condition *sine qua non* d'une recommandation plus personnalisée. De plus, la diffusion plus large de jeux de données de traces d'apprentissages des utilisateurs permet le développement d'une recherche plus ouverte basée sur les évidences (Urdaneta-Ponte et al., 2021). C'est-à-dire se référant à des études comparatives des différentes approches pour la prise de décision quant à l'algorithme à utiliser. Néanmoins, la mise à disposition de tels jeux de données ne peut se faire sans garantir l'anonymat des apprenants. Bien qu'une énorme quantité de données ait été saisie dans les environnements d'apprentissage, il est difficile de rendre ces données disponibles à des fins de recherche. Il existe deux possibilités pour permettre la publication de données utilisateurs. Dans la première possibilité : les apprenants doivent être informés et donner la permission de collecter leurs données et de les rendre disponibles à des fins de recherche.

En pratique, lorsque l'on cherche à publier un jeu de données sensible correspondant à de nombreux utilisateurs, cette possibilité est rendue difficile. En effet, il est difficile de recueillir l'accord d'un grand nombre d'apprenants, surtout lorsque les données sont destinées au domaine public et donc accessible à tous. Cela vient du fait que la mise en ligne de données sensibles sur une personne peut avoir des effets néfastes évidents. L'alternative possible est alors *l'anonymisation* : l'idée est de publier un jeu de données préservant la confidentialité des informations personnelles (l'anonymisation sera définie plus formellement au paragraphe suivant). Dans ce cas, la permission des utilisateurs est également nécessaire, mais plus simple à obtenir. L'hébergeur des données doit alors prendre en charge l'ensemble de la procédure technique et législative nécessaire à l'obtention des permissions, l'anonymisation et la mise en ligne du jeu de données. Les acteurs ayant à la fois la volonté, la compétence et les moyens de publier des jeux de données utilisateurs sont par conséquent peu nombreux. De plus, la question de l'anonymisation est complexe et les réponses apportées tant du point de vue juridiques que techniques sont en constante évolution. Cela a poussé les hébergeurs à une certaine prudence concernant la publication de jeux de données anonymisés. Dans les prochains paragraphes, nous allons étudier plus en détail cette question.

2.1.4.1 Comprendre la confidentialité

Commençons par prendre le temps de redéfinir deux termes souvent confondus : *anonymisation* et *pseudonymisation*. Pour cela, nous nous appuierons sur les définitions de la CNIL (Commision Nationale française de l'Informatique et des Libertés). La pseudonymisation est un traitement de données à caractère personnel tel qu'on ne peut pas attribuer les données à une personne physique sans avoir recours à des informations supplémentaires. Pour illustrer, un processus de pseudonymisation remplacera les informations sensibles permettant d'identifier de manière directe une personne (nom, prénom, identifiant de sécurité sociale...) par un identifiant unique. On utilise le terme *identificateur* pour de tels identifiants. Ainsi, la pseudonymisation offre une garantie contre une identification directe ; néanmoins, de nombreuses informations peuvent constituer des *quasi-identificateurs* en présence de connaissances préalables. Par exemple, le genre ou l'âge d'une personne n'est *a priori* pas un identificateur, néanmoins, si l'on a comme connaissance préalable qu'il n'y a qu'une femme de plus de 30 ans dans le jeu de données, ces informations deviennent suffisantes pour identifier la personne. On parle alors de *ré-identification*.

À l'inverse, l'anonymisation est un traitement qui consiste à utiliser un ensemble de techniques de manière à rendre impossible, en pratique, toute identification de la personne par quelque moyen que ce soit et de manière irréversible. Cela signifie en particulier qu'aucun post-traitement sur les données quel qu'il soit ne doit être en mesure de permettre la *ré-identification* des utilisateurs. Il faut insister sur le fait que la ré-identification ne désigne pas simplement la possibilité de retrouver le nom et/ou l'adresse d'une personne, mais inclut aussi la possibilité de l'identifier par un procédé d'individualisation, de corrélation ou d'inférence.

On remarque ici que le processus de pseudonymisation constitue une protection faible, car elle n'offre aucune garantie contre la ré-identification par des attaquants ayant des connaissances extérieures, à l'inverse l'anonymisation constitue une protection forte, car elle garantit l'impossibilité de ré-identifier un individu.

2.1.4.2 Comprendre la position juridique

D'un point de vue juridique, dans la plupart des pays, l'anonymisation est la règle. L'hébergeur qui souhaite publier un jeu de données doit démontrer, via une évaluation approfondie des risques d'identification, que le risque de ré-identification avec des moyens raisonnables est nul. Dans l'union européenne par exemple, dans le cas du RGPD (Rè-

glement Européen de Protection des Données), il est clairement stipulé que : « chaque personne d'un ensemble de données doit être protégée pour que l'ensemble de données soit considéré comme anonyme », de plus, la loi précise que son champ d'application s'étend à tous les jeux de données contenant des informations de nature privée, même « les données qui ne contiennent pas d'identificateurs évidents, mais qui pourraient être ré-identifiables » [3]. Enfin, l'hébergeur doit effectuer une veille régulière pour préserver, dans le temps, le caractère anonyme des données produites. Cette veille doit prendre en compte les moyens techniques disponibles ainsi que les autres sources de données qui peuvent permettre de lever l'anonymat des informations. Dans de nombreux cas, la méthode de publication choisie consiste à mettre directement le jeu de données anonymisé en accès libre : dans ce cas, l'hébergeur doit pouvoir garantir l'anonymat, indépendamment des potentielles avancées scientifiques sur le sujet, mais également de l'apparition de nouvelles données pouvant être mises en lien avec les données publiées.

2.1.4.3 L'impossibilité de garantir la confidentialité

La définition que nous avons proposée jusqu'ici de l'anonymisation est une définition littéraire ; convertir cette définition du point de vue mathématique afin de permettre le traitement des données est un sujet de recherche en lui-même. Nombre de définitions mathématiques de l'anonymisation ont été proposées, mais à ce jour aucune d'elles ne peut garantir une protection totale contre la ré-identification ; ainsi, elles admettent toutes une probabilité faible de ré-identification [4]. Cela est d'autant plus problématique que cette absence de garantie théorique se joint à une absence de garantie pratique. Ainsi, de nombreux experts pensent qu'il est devenu impossible de complétement anonymiser certaines données. Par exemple, Rocher et al. (2019) estiment que 99,98% des résidents des États-Unis pourraient être ré-identifiés dans n'importe quel jeu de données à partir de seulement quinze variables démographiques.

2.1.4.4 Fonctionnement des attaques par ré-identification et précédent Netflix

Si la question de garantir l'anonymisation est rendue aussi difficile, c'est en particulier à cause d'un type bien spécifique d'attaques dites par ré-identification. Dans ce paragraphe,

3. Source et ressources pour aller plus loin : https://www.cnil.fr/fr/reglement-europeen-protection-donnees

4. Avis 05/2014 du groupe de travail « article 29 » du 10 avril 2014 sur les Techniques d'anonymisation

nous allons voir le fonctionnement de ce type d'attaque à travers le cas illustratif de l'affaire Netflix qui constitue un précédent dans le milieu de la recommandation. Comme discuté en Section 1.3.6, entre 2007 et 2009, l'entreprise américaine de Netflix a organisé un concours ouvert pour le meilleur algorithme de filtrage collaboratif permettant de prédire les évaluations des films par les utilisateurs, sur la base des évaluations précédentes, sans aucune autre information sur les utilisateurs ou les films, c'est-à-dire sans que les utilisateurs ou les films soient identifiés, sauf par les numéros attribués pour le concours. Le ou les gagnants du concours pouvaient empocher une somme allant de 50 000 à 1 million de dollars en fonction des performances réalisées par leur algorithme. Ce challenge a créé une émulation autour de la problématique de la recommandation et a été un moteur important des avancées sur la reproductibilité dans le domaine.

En 2007, deux chercheurs, Arvind Narayanan et Vitaly Shmatikov, ont montré qu'il était possible de casser l'anonymat sur le jeu de données (Narayanan and Shmatikov, 2006). En particulier, avec 8 films annotés (dans lesquels jusqu'à deux annotations peuvent être complètement fausses) de manière publique - information qui peut être facilement obtenues lors d'une discussion, grâce à une recherche sommaire en ligne sur les préférences d'une personne ou dans certains cas directement dans des jeux de données publics (MovieLens)- les auteurs ont démontré que 99% des utilisateurs du jeu de données pouvaient être directement ré-identifiés. Pour cela, les auteurs exploitent les corrélations entre les annotations provenant de données publiques et les annotations du jeu de données Netflix. De manière intéressante, il n'est même pas nécessaire que tous les films évalués par l'abonné dans le système Netflix soient connus ; à l'inverse, en exploitant les corrélations internes des jeux de données, un ou deux films en commun sont suffisants dans une très large majorité des cas. Une fois ré-identifiés, les chercheurs connaissent l'historique complet de visionnage de films de l'utilisateur ; en particulier, ils peuvent déduire des informations qui n'auraient pas pu être déduites directement des évaluations publiques. À titre illustratif, les auteurs montrent dans leur article qu'il est possible d'inférer la préférence politique d'une personne, son orientation sexuelle ou encore son orientation religieuse.

Suite à cette publication, le 17 décembre 2019, quatre utilisateurs de Netflix ont intenté un recours collectif contre Netflix, alléguant que Netflix avait violé les lois américaines sur les échanges commerciaux et la loi sur la protection de la confidentialité des vidéos en publiant les ensembles de données. Le 19 mars 2010, Netflix a conclu un accord avec les plaignants, après lequel ceux-ci ont volontairement abandonné la poursuite.

L'affaire Netflix montre combien il est difficile et risqué de publier des jeux de données

d'utilisateurs en respectant l'anonymat. Néanmoins, il ne faut pas croire que ce problème soit spécifique au cas des données historiques de recommandation, en effet, des travaux similaires ont démontré l'efficacité de telles méthodes de ré-identification sur d'autres données telles que des données géographiques (Riederer et al., 2016; Mayer et al., 2016) ou démographiques (Cecaj et al., 2016).

2.1.5 Absence de jeu de données

Nous avons vu en Section 1.4.1 : le domaine de la recommandation à usage pédagogique est un domaine assez nouveau et en pleine expansion.

Un des axes majeurs de progression dans le domaine souvent évoqué concerne la mise à disposition de jeux de données permettant des comparaisons entre les approches (Urdaneta-Ponte et al., 2021; Hernández-Blanco et al., 2019). L'absence de jeu de données de comparaisons est un serpent de mer de la tâche de recommandation : elle est souvent citée comme un facteur nuisant à la reproductibilité dans le domaine (Dacrema et al., 2019). Plusieurs raisons expliquent cette absence, en premier lieu et nous en avons discuté à la Section précédente (Section 2.1.4), les contraintes en matière de protection des données privées rendent complexe la publication de traces d'apprentissages. Les acteurs souhaitant publier leurs données doivent nécessairement disposer d'une expertise dans le domaine de l'anonymisation. De plus, le précédent créé par l'affaire Netflix appelle les acteurs à la plus grande prudence. Néanmoins, cette raison seule n'explique pas complètement cette absence. En effet, la méthode de ré-identification proposée sur les données Netflix utilisait bien un jeu de données de préférences public : MovieLens. Cela démontre qu'il reste possible d'obtenir des jeux de données reflétant les préférences des utilisateurs, soit en leur demandant de les renseigner publiquement (comme c'est le cas de MovieLens) soit en leur demandant leur accord pour publier des données de manière non anonymisée ou pseudo-anonymisée. Malgré cela, peu de jeux de données publics existent dans le domaine de la recommandation. Un autre facteur explicatif est la concurrence existante entre les acteurs.

En particulier, la recherche dans le domaine de la recommandation bénéficie beaucoup de financements privés, en particulier venant des géants du numérique dont les laboratoires de recherche sont à la pointe du domaine depuis une dizaine d'années : (Gupta et al., 2013; He et al., 2014; Zhai et al., 2017; Eksombatchai et al., 2018; Zhao et al., 2019). Les systèmes de recommandation sont aujourd'hui des algorithmes capables de générer beaucoup d'argent dans certains contextes d'utilisation, Amazon ou Facebook par

exemple attribuent une grande partie de leurs chiffres d'affaires aux performances de leurs algorithmes (Anderson, 2006). Dans ce contexte, la publication d'un jeu de données et la totale transparence sur les méthodes utilisées rentre en conflit avec les intérêts commerciaux. Pourtant, les plateformes commerciales sont aujourd'hui sans nul doute les acteurs qui disposent du plus de données utilisateurs.

Les deux facteurs que nous venons de mentionner : la difficulté d'anonymisation et la concurrence entre les acteurs expliquent l'absence de jeu de données de manière globale dans le domaine de la recommandation. La recommandation à usage pédagogique ne fait pas exception à cette règle. D'aucuns pourraient faire la remarque que dans le cas particulier de l'éducation, certains acteurs comme les universités disposent de beaucoup de traces d'apprentissage et d'une expertise théorique permettant leur anonymisation. Cette remarque est, en effet, très pertinente, et les universités en particulier ont sans doute un rôle fort à jouer en faveur de la publication de jeux de données de traces d'apprentissages. Une raison supplémentaire qui explique cette absence dans le cas particulier de l'éducation est la relative nouveauté de la problématique de la recommandation à usage pédagogique. En effet, le virage de la formation en ligne est pris lentement par le système éducatif, même universitaire et reste un nouveau phénomène. Peu d'universités ont aujourd'hui l'infrastructure permettant de récolter et de partager leurs données. De plus, l'argument de la concurrence s'applique également au niveau universitaire. D'autres acteurs susceptibles de disposer de ce type de jeu de données sont les FLOT. Les FLOT sont parmi les acteurs qui disposent du plus de données d'apprentissage, de plus par leurs implantations, ils disposent déjà souvent d'une communauté fidèle et d'une infrastructure en place permettant de stocker et d'exploiter ces traces. Enfin, même d'un point de vue purement commercial, les FLOT tireraient largement profit de l'amélioration des algorithmes de recommandation. Néanmoins, comme nous en avons discuté en introduction (Section 0.1) l'apprentissage en ligne est un domaine nouveau et en plein essor, promis à une forte croissance, ce contexte exacerbe la concurrence entre les différentes plateformes qui, même si elles ont tout intérêt individuellement à améliorer leur propre système de recommandation, ont beaucoup moins d'intérêt à partager les données qui sont à la base de leurs modèles économiques pour une amélioration globale des systèmes.

En résumé, on remarque que les différents acteurs ont, pour des raisons de concurrences, de compétence et d'aversion aux risques, peu d'incitation à rendre leurs données disponibles. Cette absence de jeu de données et d'initiatives visant à leur publication fût un élément clé du contexte de lancement du projet.

2.1.6 Le flou autour des licences ouvertes

Si l'utilisation des licences ouvertes peut sembler simple de prime abord, on remarque que dans leurs utilisations certaines mauvaises pratiques ou méconnaissances peuvent rendre complexe l'identification de la licence ou la publication de contenu sous licence ouverte (Wang and Towey, 2017). Dans cette section, nous allons détailler quelques points qui rendent complexe l'utilisation des licences ouvertes en pratique. Nous verrons que ces points, même s'ils ne sont pas toujours spécifiques aux cas des licences ouvertes, contribuent à un sentiment de complexité dans l'usage des licences ouvertes et rendent difficile l'indexation semi-automatique des RELs. Enfin, il est important de mentionner que les questions liées aux licences émergentes souvent lorsqu'on publie du contenu ouvert. En effet, on peut être plus facilement jugé sur la qualité et le bon respect des règles d'utilisation que lorsque le contenu reste confidentiel.

2.1.6.1 Difficulté d'identification de la licence

Dans un cas idéal, la bonne pratique veut que la licence reliée aux droits d'utilisations d'une ressource soit clairement visible dans la ressource ainsi que dans ses méta-données. Notons que dans le cas des licences CC-0 il n'y a pas d'obligation, et qu'en dehors de la courtoisie, rien n'oblige la personne qui utilise la ressource à citer l'auteur et la licence. Bien sûr, en pratique, la courtoisie ne s'applique pas toujours et il est monnaie courante d'utiliser des images ouvertes (mais pas que) par exemple dans des présentations ou dans des cours en oubliant de citer l'auteur. Cela est dommageable, car le cercle vertueux du partage est ainsi brisé et il devient impossible de distinguer une image CC-0 d'une image piratée (Wang and Towey, 2017). Par ailleurs, la position par défaut en l'absence de toute information de licence est « tout droits réservés », ce qui signifie qu'en l'absence d'information contraire, une personne qui souhaiterait utiliser l'image doit faire l'hypothèse que c'est impossible en raison de la protection des droits d'auteurs. En pratique, et nous l'avons vu notamment en Section 0.5, la variété des usages, des pratiques et des formats de méta-données font que même quand la courtoisie est respectée, l'identification de la licence de manière automatique peut s'avérer complexe et nécessite souvent le développement d'un algorithme de détection spécifique pour chaque site. En effet, l'identification est souvent simple pour un œil humain, mais terriblement difficile pour un algorithme générique.

Un autre point concerne les dépôts de RELs : dans la plupart de ces dépôts, il est

clairement exprimé que l'ensemble des ressources hébergées sont des RELs. Les ressources peuvent donc être utilisées dans la philosophie de l'éducation ouverte. Néanmoins, dans certains cas, la licence de la ressource n'est pas disponible (Amiel and Soares, 2016). Nous faisons alors face à une contradiction : d'un côté le dépôt s'engage à fournir des RELs ce qui nous pousse à déduire que la ressource est ouverte, de l'autre côté, il n'y a pas de licence, la règle par défaut qui s'applique est donc « tout droits réservés ». Les questions pour un utilisateur sont : puis-je faire confiance au dépôt ? quelle sont les conditions de partage de la ressource en question ? La réponse est dans la plupart des cas : oui, il est judicieux de faire confiance au dépôt. Une utilisation convenable ne posera pas de souci ; néanmoins, en cas de contestation, certes extrêmement rare, la responsabilité semble porter sur l'utilisateur qui doit pouvoir justifier de son droit à utiliser le contenu.

2.1.6.2 La variété des licences ouvertes

Lorsque nous avons présenté les licences ouvertes en introduction, nous l'avons fait par le biais des licences « Créative Commons ». Elles constituent un excellent point d'entrée, car ce sont des licences créées pour traiter une variété importante de cas et être le plus pédagogiques possibles. Néanmoins, il existe en pratique une large variété de licences ouvertes. Reconnaître une licence ouverte d'une licence non ouverte n'est pas toujours trivial et nécessite parfois une expertise juridique (Amiel and Soares, 2016). Surtout quand ces licences rentrent en potentiel conflit avec la législation du pays, par exemple, en France le droit d'auteur est inaliénable, les licences CC-0 n'existent donc pas vraiment, elles sont en théorie au minimum CC-BY néanmoins depuis 2006 l'application d'une licence ouverte constitue une exception au droit d'auteur [5].

2.1.6.3 Règles de compatibilité

Un autre point de confusion concerne l'identification de la portée de la licence et les règles de compatibilités entre les licences ouvertes. Supposons un cours ouvert, par nature, il est probable que ce cours soit un matériel composite, lui-même composé de différentes RELs. Cela est même souhaitable, en effet, un des intérêts des REL est de facilement pouvoir réadapter, réagréger ou améliorer du contenu déjà existant (Wiley, 2014). Bien sûr, un cours, même non ouvert, est un matériel composite par nature et

5. Michel Thiollière, Commission des affaires culturelles, Sénat, Rapport sur le projet de loi relatif au droit d'auteur et aux droits voisins dans la société de l'information, avril 2006, p. 52.(France) : `http://www.senat.fr/rap/l05-308/l05-30815.html`

il est rare que chaque illustration, chaque exemple, chaque définition... soit produit *ex nihilo*. Bien souvent, et cela semble indispensable pour assurer la qualité du cours, des définitions de la littérature scientifique, des exemples ou des illustrations particulièrement pertinents sont ré-utilisés. La différence majeure est que dans le cas du contenu ouvert, la démarche de partage favorise et facilite ces pratiques, notamment en permettant de ne pas demander systématiquement le consentement du détenteur des droits et en instaurant un cadre légal autour de ces pratiques.

Néanmoins, une question se pose alors. Puisque que le cours est composite, il est probable qu'il y ait de multiples licences dans les contenus utilisés. Comme mentionné précédemment (Section 0.2.1) chaque licence accorde différents droits, en particulier certaines d'entre elles sont contaminantes (clause NC par exemple, voir la Section 0.2.1 pour un rappel de cette notion). Ainsi, comment savoir quelle licence attribuer au cours dans sa globalité d'un point de vue légal ? Une licence est-elle d'ailleurs applicable sur le cours entier ou faut-il traiter chaque image, chaque paragraphe au cas par cas ? Quelles règles de compatibilité des licences s'appliquent ? Lorsque l'ensemble des ressources utilisées est publié sous licence *Creative Commons*, les compatibilités sont relativement simples à établir (Voir Figure 2.1). Lorsque ce n'est pas le cas, évaluer la compatibilité est complexe et peut nécessiter une expertise juridique.

Une autre question souvent évoquée concerne la clause de non-dérivation : cette clause stipule que l'œuvre doit être utilisée uniquement dans les conditions de son usage initial. En pratique, il peut être difficile de déterminer si cette clause est réellement satisfaite. Par exemple, supposons que « La Persistance de la Mémoire », célèbre tableau de Dali, soit une œuvre disposant d'une licence CC BY-ND. Peut-elle alors être utilisée dans un cours de science cognitive ? ou est-ce un détournement du sens original de l'œuvre ? Pour cette raison notamment une licence non dérivative n'est pas considérée ouverte, néanmoins nombre de ressources ouvertes utilisent ce type de ressources en considérant ne pas se détourner de l'usage initial de l'œuvre. Par ailleurs, où s'arrête la contamination au niveau de l'exercice où l'image est utilisée ? ou contamine-t-elle tout le cours ?

Il en va de même pour la clause NC qui interdit une ré-utilisation commerciale. L'interprétation de la clause est difficile : si je souhaite utiliser la ressource pour un séminaire d'entreprise, vais-je pouvoir le faire ? et pour un congrès scientifique avec des frais d'inscription, est-ce autorisé ? Dans le cas des licences CC-BY-NC il est précisé que seules les utilisations « principalement destinées directement à un avantage commercial ou une compensation financière » sont proscrites. Cette définition volontairement ambiguë est

CHART 3: POSSIBLE COMBINATIONS OF CC CONTENT [101]

	PUBLIC DOMAIN	PUBLIC DOMAIN	BY	BY SA	BY NC	BY ND	BY NC SA	BY NC ND
PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
BY	✓	✓	✓	✓	✓	✗	✓	✗
BY SA	✓	✓	✓	✓	✗	✗	✗	✗
BY NC	✓	✓	✓	✗	✓	✗	✓	✗
BY ND	✗	✗	✗	✗	✗	✗	✗	✗
BY NC SA	✓	✓	✓	✗	✓	✗	✓	✗
BY NC ND	✗	✗	✗	✗	✗	✗	✗	✗

FIGURE 2.1 – Compatibilité entre les licences ouvertes Creative Commons. Source : https://commons.wikimedia.org/

destinée à capturer l'intention de la communauté des utilisateurs de NC-sans imposer des restrictions détaillées qui sont soit trop larges ou trop étroites. Créative Commons explique que même s'il y a toujours des utilisations qui sont difficiles à catégoriser comme commerciales ou non commerciales, il est souvent relativement facile de déterminer si l'utilisation est autorisée, et les conflits connus sont relativement peu nombreux compte tenu de la popularité des licences [6].

2.1.6.4 Publier du contenu sous licence ouverte

Une dernière limitation souvent exprimée sur l'utilisation des licences ouvertes est la nécessité de déterminer qui est le détenteur des droits lors de la création de la ressource (Belikov and Bodily, 2016). Lorsque la ressource est « tout droits réservés » il n'y pas de difficultés : c'est la position par défaut et il n'y a donc besoin de l'accord de personne pour définir les droits. En revanche, lorsqu'on veut publier une ressource ouverte, il est nécessaire de connaître le détenteur des droits, car lui seul peut céder une partie de ses droits. De prime abord, la question semble simple, l'auteur serait le détenteur des droits.

On remarque néanmoins que dans de nombreux cas l'auteur n'est pas le détenteur des droits, par exemple à la sortie d'un *blockbuster* Marvel l'auteur ou le réalisateur n'est pas le détenteur des droits, c'est bien la compagnie Disney qui a commandé l'œuvre qui détient les droits. Dans le cas de l'éducation, lorsqu'un professeur du secondaire rédige son cours est-il détenteur des droits ? Dans le cas français, il semble que dans le cas d'un professeur du secondaire, le détenteur des droits soit l'état. Dès lors, si le professeur souhaite rendre le cours ouvert, il doit obtenir l'aval de sa hiérarchie. Pour un professeur d'université, la règle est différente, en France les enseignants du supérieur sont propriétaires de leur cours. Dans bien des cas, la question ne semble pas si évidente à trancher.

2.1.6.5 Conclusion sur l'utilisation des licences ouvertes

Lorsqu'on analyse l'ensemble de ces difficultés pratiques, on observe deux choses. Premièrement, le fait que le contenu soit publié engage la responsabilité des personnes et oblige donc à être plus pointilleux quant au respect des règles en vigueur. C'est un effet souhaitable de l'ouvert qui oblige au respect des règles et encourage la qualité à tous les

6. Foire aux questions - Creative Commons - Est-ce que mon utilisation viole la clause Non-Commercial des licences ? https://creativecommons.org/faq/fr/#est-ce-que-mon-utilisation-viole-la-clausenon-commercial-des-licences

niveaux. Deuxièmement, le fait que « tout droits réservés » soit la position par défaut facilitent largement ce modèle de répartition des droits. En effet, cela rend plus complexe la création de contenu ouvert que celle de contenu propriétaire, une personne cherchant à minimiser ces efforts est de fait inciter à créer du contenu propriétaire. À la lecture de ces difficultés, on comprend bien que les licences ouvertes peuvent être perçues comme difficiles à appliquer en pratique. En effet, bien des problématiques que nous avons soulevées se posent fréquemment et l'on ne peut pas décemment demander une expertise juridique sur chacun de ces sujets à toute personne souhaitant publier du contenu ouvert. Néanmoins, la majorité des problèmes mentionnés reste valable pour des licences non ouvertes dans la mesure où l'on respecte scrupuleusement les règles en vigueur, le fait que l'ouvert ne soit pas la position par défaut, mais l'exception complexifie grandement la situation. De nombreuses associations, en particulier, Creative Commons travaillent pour rendre l'ouvert facile d'accès et simple d'utilisation, en fournissant à la fois une expertise juridique et un travail de médiation précieux sur le sujet. De plus, il est important d'observer que lorsque l'utilisation est raisonnable, il ne semble pas avoir de problème quant à l'utilisation des ressources ouvertes. Cela vient du fait que les personnes ayant créé du contenu ouvert sont favorables à sa diffusion, en particulier à des fins éducatives.

2.2 Stratégie mise en place par X5gon

L'objectif de la section est de définir les grands axes de la stratégie d'X5GON déterminés lors de la première année du projet (2018-2019). Cette stratégie se décomposait en trois axes majeurs qui guidaient les restitutions et les contributions proposées par les partenaires du projet (Section 2.3).

2.2.1 Indexation automatique

Dans le contexte de lancement du projet apparaissait clairement la nécessité de disposer d'un jeu de données de REL ouvert. Pour être représentatif de la plus grande diversité, il était nécessaire que ce jeu de données soit construit à partir d'une agglomération de contenus hébergés sur différents sites.

Ce fût la problématique du premier axe du projet que de collecter et de rendre disponible un corpus de ressources éducatives libre. Dans un contexte où le nombre de ressources était pléthorique, l'annotation manuelle semblait extrêmement difficile autant

en tant qu'en coût. Deux options restaient alors possibles : l'*indexation participative* et l'*indexation automatique*. L'indexation participative repose sur l'idée d'un référencement collectif des ressources dans lequel tous un chacun peut référencer une ressource (et ces méta-données) qui lui semble pertinente auprès d'une entité centrale en charge de collecter l'ensemble des liens. L'indexation automatique repose sur des algorithmes qui de manière automatique doivent fouiller le web à la recherche de ressources pertinentes, en pratique, la tâche de détection étant très complexe, on utilise des méthodes d'*indexation semi-automatique* qui limiteront leurs recherches à l'intérieur d'un nom de domaine donné. Dans le projet, le choix fût fait d'opter pour une indexation semi-automatique des ressources. Cette indexation était un défi important, car des tâches relativement simples pour un humain comme la détection de la licence s'avèrent difficile à automatiser (Amiel and Soares, 2016). Pour ce faire, les partenaires slovènes du projet ont eu pour rôle de développer un *crawler* généraliste paramétrable pour s'adapter à chacun des sites cibles de RELs.

Dans le but de pouvoir conserver une homogénéité de représentation entre les ressources provenant des différents sites, le choix a été fait de conserver uniquement un ensemble minimal d'informations pour chaque ressource (comme la licence, le titre, l'auteur, l'url ...). Néanmoins, un enjeu important concernait la possibilité de récupérer des informations relatives à la structuration des RELs sur les différents sites. En particulier, les informations permettant de définir l'ordre de consultation des REL défini par les enseignants devaient être récupérées afin d'être utilisées comme jeu de données de validation. Cela devait permettre de répondre ainsi à la nécessité d'études sur des utilisateurs experts tout en évitant les difficultés organisationnelles inhérentes à la mise en place de telles études dans un contexte non formel.

Dans le but de palier au faible nombre d'informations relatives à chaque ressource récupérée par le *crawler*, un ensemble de services développés au sein du projet a eu pour rôle d'enrichir les ressources. Parmi ces services, nous retrouvions notamment des services de traduction du contenu et de représentation sémantique du contenu de la ressource (voir Section 2.3.1.2). Une fois l'ensemble du processus achevé, la ressource était finalement enregistrée dans une base de données (voir Section 2.3.1.4) qui servirait ensuite de corpus de référence pour l'ensemble des outils développés au sein du projet, notamment les moteurs de recherche et de recommandation.

L'ensemble du processus appelé « pipeline d'ingestion » sera détaillé en Section 2.3.1. Par l'intermédiaire d'une API (Section 2.3.5.1) la base de données ainsi constituée a

été rendue disponible de manière publique permettant son utilisation libre par d'autres acteurs.

2.2.2 Création d'une communauté de partenaires et d'apprenants

Le manque de jeux de données de traces d'apprentissage a clairement été identifié comme un frein aux développements de systèmes de recommandation (Urdaneta-Ponte et al., 2021). D'autre part, les utilisateurs de REL identifient souvent la difficulté à trouver une ressource ouverte pertinente comme une barrière à l'utilisation de ressource ouvertes (Belikov and Bodily, 2016). Cette difficulté découle principalement de l'éclatement des ressources entre les différents sites. La stratégie d'X5GON visait à fournir des outils clé en main pour les utilisateurs de RELs à travers des modules de recherche et de recommandation dédiés et facilement intégrable sur les sites hébergeant les ressources. Le terme *plug-in* a été introduit par les membres du projet pour désigner ces modules que nous détaillerons en Section 2.3.3. L'intérêt de ces modules était triple :

Faciliter l'accès aux RELs : et ainsi répondre à la difficulté d'accès aux ressources pertinentes,

Connecter les sites entre eux et favoriser les navigations inter-sites : et ainsi bénéficier des avantages du décentralisé (variétés des pratiques) en conservant une expérience d'apprentissage fluide à travers les sites,

Collecter des traces d'apprentissages des utilisateurs : et ainsi pouvoir fournir des recommandations personnalisées permettant d'aller plus loin dans l'expérience d'apprentissage en ligne.

Plus généralement, X5GON avait la volonté de connecter les sites entre eux et d'encourager les initiatives en faveur de la mutualisation des ressources. Dans cette politique de mutualisation, un autre point important concernait la mutualisation des traces d'apprentissages. En effet, les parcours d'apprentissage collectés à travers le *plug-in* ne témoignent pas de l'ensemble des interactions des utilisateurs et étaient naturellement biaisés par les algorithmes de recommandation et de recherche implémentés.

Pour recueillir une plus large variété de traces d'apprentissage, un outil appelé *connect service* a été développé. Il devait s'intégrer à une plate-forme d'apprentissage et permettre de collecter l'ensemble des interactions utilisateurs. En pratique, cela consistait à collecter

et à mutualiser différentes interactions chez différent partenaire pour mieux appréhender les ressources et le comportement des apprenants. Dans un premier temps, cet outil a été utilisé sur les ENA (Environnement Numérique d'apprentissage) des universités partenaires de Nantes et de Valence ainsi que sur le site partenaire Videolectures.NET. L'ambition des partenaires du projet était de pouvoir collecter une large variété d'informations supplémentaires grâce à cet outil : les interactions d'un utilisateur sur une vidéo (pause, retour arrière, vitesse accélérée, etc), les interactions sur un document (temps de lecture) ou encore sur un quiz (réponses essayées, note obtenue...). Avec ces retours d'informations plus riches, il devrait alors être possible d'inférer des informations supplémentaires sur la ressource (par exemple ses points de difficulté) et sur l'apprenant (sa modalité préférée, les points de blocage dans sa compréhension). Pour les plateformes partenaires, l'intérêt était double : bénéficier d'une recommandation personnalisée pour leurs apprenants et participer à un effort collectif dans le sens d'une éducation plus ouverte. Dans ce sens, un service nommé X5MOODLE spécifiquement dédié à l'ENA Moodle a été développé dans le but de connecter facilement une instance de cours sur Moodle avec le réseau de RELs crée par X5GON. Nous détaillerons cet outil en Section 2.3.4.2.

Enfin, dans le but de toucher un maximum de personnes, notamment au-delà des frontières traditionnelles de l'éducation formelle, une plate-forme de ressources éducatives nommée X5LEARN a été développée. Elle se présente comme un répertoire de RELs disponible et consultable directement en ligne. Et se voulait un point d'entrée pour toute personne à la recherche d'une expérience d'apprentissage ouverte.

2.2.3 Stimulation de la recherche

Le troisième axe visait à créer une émulation autour de la problématique de l'éducation en ligne grâce aux RELs. En particulier dans la communauté scientifique afin de promouvoir la recherche et donc les avancées dans le domaine. Mais aussi au niveau des différents acteurs de l'éducation afin de promouvoir les pratiques de l'éducation ouverte et de sensibiliser un maximum d'acteurs à ses avantages. Pour cela, la stratégie mise en place visait à l'organisation de divers évènements dont des congrès et des hackathons.

Enfin, un facteur important pour faciliter la recherche, plus généralement l'innovation dans le domaine concerne la disponibilité d'outil spécifique clé en main. Cela peut être des modèles de représentation de document spécifiquement dédiés aux RELs, des corpus de données disponibles, etc. Par exemple, la littérature dans le domaine du traitement du langage naturel a démontré que l'utilisation de corpus spécifiques à un domaine améliorait

la tâche de représentation des documents. En effet, comme présenté dans l'état de l'art (Section 1.1) ces approches tirant profit des régularités statistiques dans les données, elles apprennent de meilleures représentations lorsque le domaine y est largement représenté.

Pour donner accès à ces outils clés en main, le choix a été fait de déployer une API spécifique nommée LAM API : nous discuterons de cette API en détail en Section 2.3.5.2.

2.2.4 Développement par API

Des trois axes présentés précédemment, on peut extraire deux contraintes majeures : i) la nécessité d'interopérabilité entre les acteurs et les plateformes, ii) la nécessité de mise à disposition de technologie spécifique à l'éducation ouverte, libre et facile d'utilisation. Pour répondre à ces contraintes, le choix technique fait durant le projet concernait le développement par API. Dans ce document, nous utiliserons le terme « API » comme un abus de langage pour désigner une API web même si d'autres types d'API existent. API est l'acronyme de Application Programming Interface (interface de programmation d'applications), qui est un intermédiaire logiciel permettant à deux applications de communiquer entre elles. Contrairement à une interface utilisateur, qui relie un ordinateur à une personne, une interface de programmation d'applications relie des ordinateurs ou des logiciels entre eux. Elle n'est pas destinée à être utilisée directement par une personne (l'utilisateur final) autre qu'un programmeur informatique qui l'incorpore dans un logiciel. Une API est souvent composée de différentes parties qui agissent comme des outils ou des services à la disposition du programmeur. Ici, on remarque un des premiers avantages de l'API qui est l'automatisation des processus, cela a du sens par rapport à ce que nous avons vu précédemment. En effet, la dispersion des ressources, la difficulté d'automatisation de l'indexation et de la récupération des données qui en découle ont été identifiés comme des freins majeurs. Un autre avantage du modèle API est la possibilité de mise à jour et de constantes améliorations des outils et des données proposées. En effet, contrairement à la publication de jeu de données en dur, l'API permet de faire grossir au fil du temps la taille des jeux de données et d'améliorer les modèles proposés pour les représenter sans surcoût de développement ou de mises à jour pour les applications utilisatrices. Un autre principe central des API est le principe de dissimulation de l'information qui décrit le rôle des interfaces de programmation comme permettant la programmation modulaire en dissimulant les détails de mise en œuvre des modules afin que les utilisateurs des modules n'aient pas besoin de comprendre les complexités à l'intérieur des modules. Cela permet l'utilisation pour une large variété d'utilisateurs : des développeurs d'interfaces utilisa-

teurs, des institutions, des chercheurs... Notamment, cela permet à des chercheurs ayant des spécialités différentes de pouvoir tirer profit des outils/données sans nécessité d'une connaissance globale de l'ensemble des domaines engagés.

2.3 Travaux réalisés

L'objectif de la section est de présenter les différents travaux réalisés au sein du projet X5GON, en particulier, ceux en lien avec la problématique de la recommandation à visée pédagogique. Dans cette section, nous prendrons bien soin de préciser le rôle de la cellule nantaise et en particulier le mien (utilisation du « nous ») dans les développements.

Le schéma de la Figure 2.2 détaille le fonctionnement global et les interactions entre les différents développements réalisés durant le projet : il nous servira de fil conducteur.

2.3.1 Pipeline d'intégration et d'ingestion

La *pipeline* d'intégration et d'ingestion constituait le point d'entrée d'une ressource dans le projet X5GON, c'est une chaîne de traitement complètement automatisée qui a pour rôle d'identifier des RELs sur différents sites et de stocker une représentation enrichie de ces ressources dans la base de données du projet.

2.3.1.1 Organisation du pipeline

L'organisation de la *pipeline* est décomposée en trois grandes parties : i) un *crawler* chargé de repérer la ressource, ii) un service de traduction automatique chargé de fournir une version textuelle de la ressource en différentes langues et iii) un dernier service d'enrichissement cherchant à produire une représentation riche de la ressource.

Le *crawler* a été développé par le partenaire de l'université slovène de Ljubljana à partir de *crawler*s initialement développés pour le site web Videolectures.net. En raison de la variété des pratiques, le développement de ce *crawler* est un sujet de recherche stimulant et un véritable challenge. Nous n'allons pas ici renter dans les détails d'implémentations et les problématiques spécifiques liées à ce *crawler*. Néanmoins, le lecteur curieux peut avantageusement se référer au délivrable « D2.1 Requirements & Architecture Report (M6) » du projet et en particulier à la Section 4.2.1 pour un traitement plus détaillé [7]. Ce qu'il faut retenir pour la suite et que le *crawler* permet d'identifier la ressource et

7. lien vers le délivrable : https://www.x5gon.org/science/deliverables/

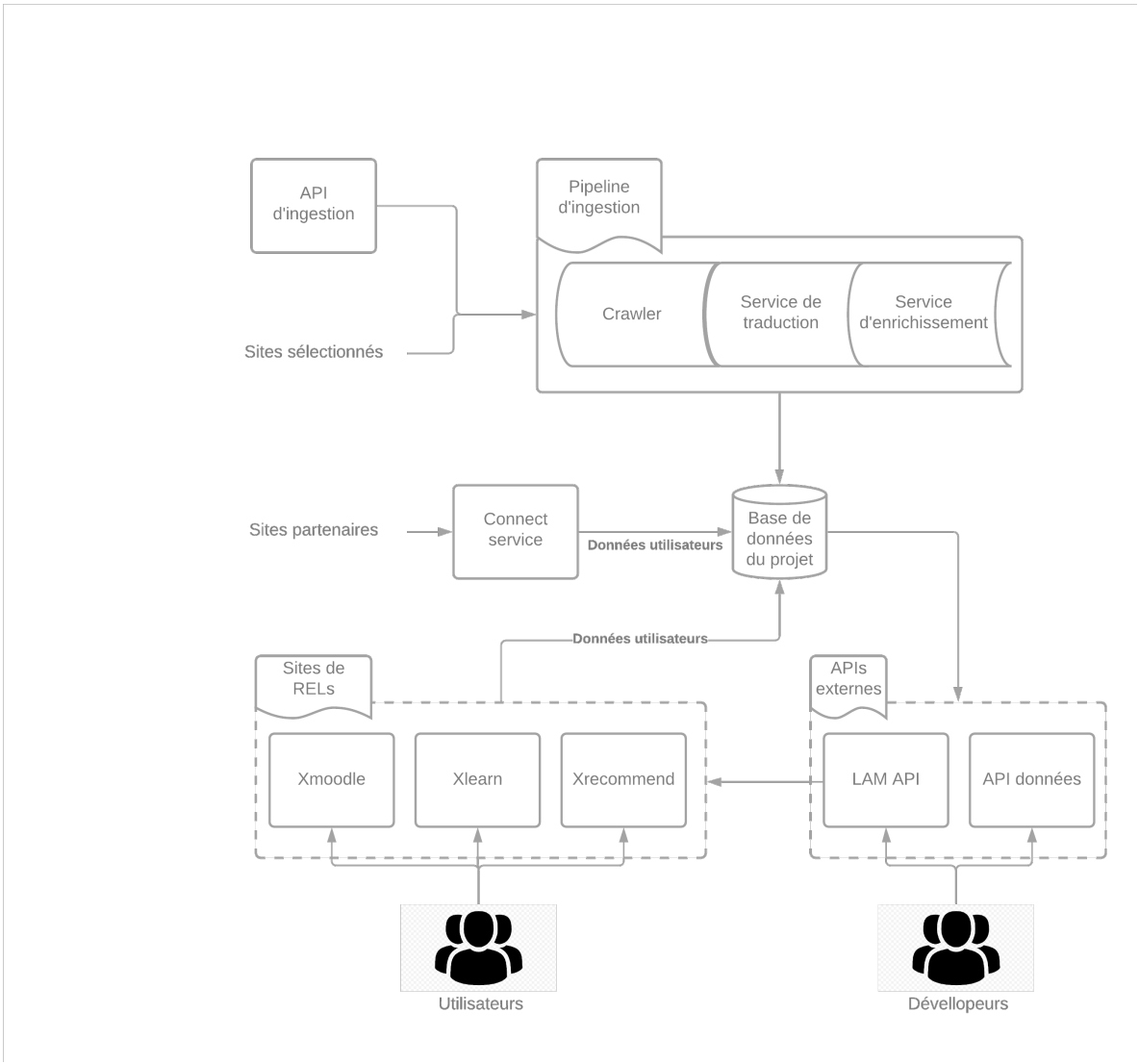


FIGURE 2.2 – Architecture globale des développements du projet.

de stocker un ensemble de métadonnées variables selon les sites en lien avec la ressource (titre, auteur, url d'accès, etc.). À ce jour, le *crawler* n'est en particulier pas capable d'interpréter les liens éventuels entre les ressources (exercices liés, partie d'un même cour, etc.). En simplifiant, le *crawler* considère le lien d'accès comme constituant unitaire de la ressource. Ainsi pour une présentation en conférence par exemple, la vidéo est considérée comme une ressource, le pdf contenant les diapositives aussi, mais aucun lien spécifique n'est fait entre les deux ressources. De la même manière, si sur un site les quatre parties d'un même cours sont représentées par quatre liens différents vers les pdf de chacune des parties, l'information de l'appartenance au même cours sera perdue.

La deuxième étape de la *pipeline* est le service de transcription et de traduction de l'université de Valence (Espagne). À ce stade, le service de l'université de Valence reçoit un lien téléchargeable vers le contenu de la ressource, naturellement ce lien peut mener à une vidéo au format mp4, comme à un document pptx ou encore à un pdf. Le rôle du service est alors de fournir dans chacun des cas une représentation textuelle multilingue de la ressource. Cette représentation sera ensuite utilisée pour créer une représentation sémantique de la ressource ainsi que pour proposer la ressource en différentes langues. Pour réaliser cette tâche, les partenaires de Valence utilisent un algorithme de transcription automatique dans le cas des contenus audio ou vidéo et un algorithme d'extraction de texte dans le cas des contenus textuels. Une fois cette représentation textuelle obtenue, un algorithme de traduction est utilisé pour fournir une version en différentes langues. Les langues actuellement supportées sont : le français, l'anglais, l'espagnol, l'allemand, le slovène, le catalan, le mandarin et le russe. L'ensemble des algorithmes utilisés sont des développements des partenaires valencians connus pour être à la pointe dans le domaine, pour plus d'informations voir : le « délivrable D3.5 Final support for Cross-lingual OER (M30) » section 2 et 3 [8].

La troisième étape concerne la représentation sémantique de la ressource : une représentation textuelle simple est souvent insuffisante pour bien représenter la ressource et en particulier pouvoir la mettre en lien avec les autres ressources. C'est un élément clé pour pouvoir comparer les ressources entre elles, les connecter et produire la recommandation, pour cette raison, c'est une étape majeure de notre problématique que nous allons prendre le temps de détailler dans la section suivante.

8. lien vers le délivrable : https://www.x5gon.org/science/deliverables/

2.3.1.2 Systèmes d'enrichissement de la ressource

Le système d'enrichissement a entièrement été développé par la cellule nantaise, en particulier Walid Bem Rhomdane et moi-même pour la phase de développement, avec l'apport de Colin de la Higuera pour la conception. Il comprend plusieurs méthodes de représentation des documents implémentés sous formes de routines qui sont déclenchées dès qu'une nouvelle ressource est indexée. Certaines de ces méthodes nécessitent des calculs sur l'ensemble du corpus, pour se faire des scripts ont été mis en place permettant de mettre à jour les représentations de chacune des ressources de manière automatique tout en maintenant un environnement de production.

Le système déploie six méthodes de représentations des documents : trois méthodes classiques et trois méthodes dérivées pour prendre en considération la temporalité interne des ressources. Commençons par les trois méthodes classiques.

2.3.1.2.1 Méthodes implémentées de représentation sémantique

Pour représenter nos ressources à partir des textes bruts, nous avons utilisé trois méthodes de représentation des documents :

TF-IDF : La méthode TF-IDF présentée en Section 1.1.1.2.1 représente les documents par une distribution sur les mots-clés du corpus. Nous avons retenu cette méthode, car c'est une méthode de référence fréquemment utilisée dans le domaine qui a l'avantage d'avoir des dimensions facilement interprétables. Dans le contexte du projet, les dimensions les plus importantes ont également été utilisées pour permettre aux utilisateurs finaux d'avoir une idée d'ensemble du contenu de la ressource grâce à ces quelques mots-clés.

Doc2Vec : La méthode Doc2Vec (Le and Mikolov, 2014) présentée en Section 1.1.2.2 représente les documents par un vecteur dense non interprétables. Nous avons retenu cette méthode, car elle fournit une représentation dense idéale pour l'utilisation combinée avec des modèles d'apprentissage automatique, notamment pour la tâche de recommandation.

Wikifier : La méthode Wikifier (Brank et al., 2017) présentée en Section 1.1.1.3, représente les documents par une distribution sur les concepts Wikipédia. C'est donc une représentation basée ontologie qui présente - comme les approches sac-de-mots - l'avantage de l'interprétabilité des dimensions. Elle offre également la possibilité de connecter les ressources au web sémantique.

Les trois approches retenues sont complémentaires et offrent différentes opportunités. Dans l'idée d'un partage des ressources et de leurs représentations avec d'autres chercheurs par l'intermédiaire d'API le choix fait ici est de laisser un maximum de liberté quant aux méthodes utilisées pour les tâches de plus haut niveau.

Néanmoins, ces trois approches ont un défaut commun qui nous a semblé problématique dans le cas de ressources éducatives et en particulier pour des ressources mutimodales et de taille variable : elles ne tiennent pas compte de la temporalité interne des notions dans le document et représentent à l'inverse le document comme un bloc indivisible. Cela semble problématique d'un point de vue utilisateur, en effet, imaginons un utilisateur s'intéressant à la question spécifique des émissions de CO2 des centrales nucléaires, il vient de consulter une première vidéo de vulgarisation sur le sujet et souhaite approfondir ces connaissances. Supposons que lui recommander un chapitre du rapport du GIEC soit pertinent. Avec une représentation monolithique, il est peu probable que les deux ressources se retrouve liés : certes le rapport du GIEC traite de la question et est une source très pertinente, mais il traite aussi de nombreux autres sujets (à titre d'exemple le résumé pour les décideurs de 2013 et un document de 222 pages). Une recommandation pertinente devrait alors recommander le chapitre particulier traitant du sujet dans le rapport, pour réaliser de telle recommandation, nous pensons qu'une représentation à grains plus fins est nécessaire. De la même manière, avec une méthode de représentation monolithique, il semble difficile de prendre en compte la continuité pédagogique, la construction du discours et l'enchaînement des idées. Ici encore, avoir une représentation plus fine tenant compte de la temporalité du discours semblent un prérequis important.

2.3.1.3 Méthode de représentation sémantique à grains fins conservant la chronologie

Pour palier à cette problématique, nous avons étendu avec une approche simplificatrice les trois méthodes présentées plus haut. Nous avons nommé ces trois nouvelles méthodes : *continuous TF-IDF*, *continuous Doc2Vec* et *continuous Wikifier*.

Les trois méthodes utilisent une phase de pré-traitement du texte commune. Elle consiste à découper le texte brut en un nombre choisi de tronçons de taille *chunk_size*, chaque tronçon ayant un chevauchement de *overlap* mots avec le tronçon précédent. Ensuite, l'algorithme de création de la représentation (TF-IDF, Doc2Vec, Wikifier) doit être appliqué de manière indépendante sur tous les tronçons comme s'il s'agissait de documents distincts. Une fois calculé, l'ensemble des vecteurs de représentations pour chaque tronçon

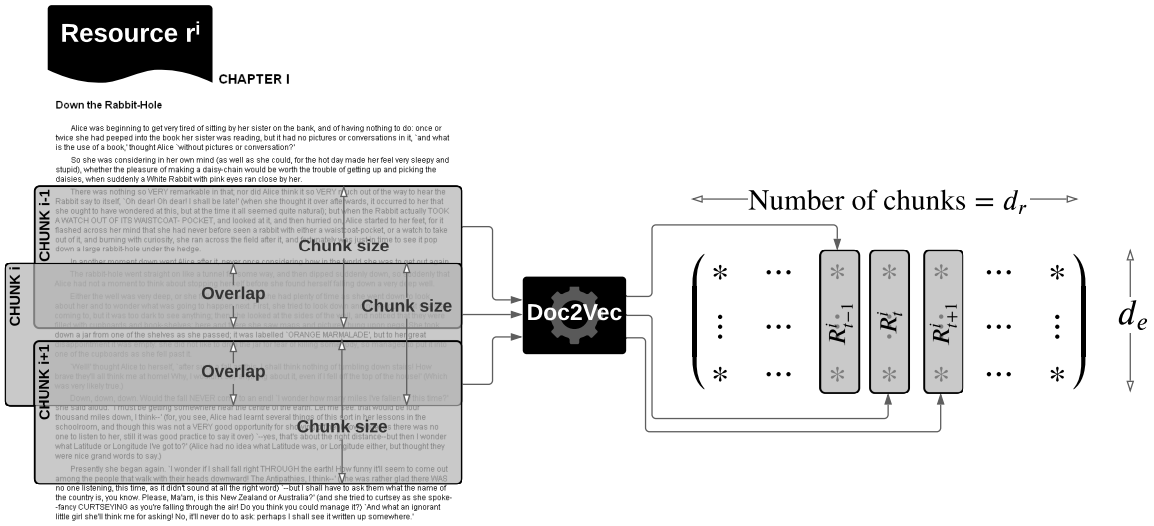


FIGURE 2.3 – Procédure de construction d'une représentation sémantique des ressources à grains fins préservant la chronologie

peut être agrégés dans une matrice de taille $nb\_chunk \times d_e$ dans laquelle la $i^{ème}$ colonne est le vecteur de représentation sémantique du $i^{ème}$ tronçon. L'ensemble de la construction de ces représentations sémantiques des ressources préservant la chronologie est résumé dans la Fig. 2.3. Les paramètres *chunk_size* et *overlap* peuvent être calibrés de différentes manières en fonction de l'utilisation finale, par exemple, un $chunk\_size = 1000$ mots correspond approximativement à 8-10 min de discours pour une présentation à débit de parole moyen avec diapositives. Dans le cas du projet, il s'avère que la méthode est suffisamment peu coûteuse pour permettre d'être recalculée à la volée, ainsi le choix des valeurs de ces paramètres est à la discrétion de l'utilisateur lors d'un appel via l'API (LAM API voir Section 2.3.5.2).

2.3.1.4 Base de données

La base de données du projet est le lieu de stockage final des ressources une fois indexées et enrichies : c'est un élément constitutif central du projet avec lequel la plupart des fonctionnalités et des outils développés interagissent. La base de données a été développée par la cellule nantaise en collaboration avec la cellule Slovène, nous avons pris en charge la conception et le schéma de la base de données à Nantes. La cellule Slovène a eu la charge de l'administration, de l'hébergement et de l'implémentation de la base de données.

Dans cette section, nous n'allons pas détailler l'ensemble des tables de la base, mais nous concentrer sur quelques points essentiels qui nous semblent importants pour la suite. Le schéma relationnel de la base de données (Figure 2.4) servira de fil directeur à notre présentation.

Nous allons d'abord nous intéresser à la partie publique de la base de données (voir Figure 2.4). Cette partie est dite publique, car elle ne contient que des données relatives au contenu et toutes les informations qu'elle stocke sont directement accessibles via la MAIN API (voir Section 2.3.5.1). La table *oer_materials* est la table alimentée par le *crawler* : elle contient les métadonnées relatives à la ressource (titre, description, etc.), chaque ressource est également identifiée par une url de téléchargement qui lui est unique (clé étrangère de la table url). De plus, chaque ressource étant également liée à travers une clé étrangère à un ensemble d'enregistrement de la table *material_contents* contenant les versions muti-langues de la représentation textuelle de la ressource.

Dans le but de pouvoir exprimer des relations de séquentialité entre les ressources, les tables *episodes* et *series* ont été créées. En pratique, le *crawler* à échouer à récupérer ce type d'information. Nous discuterons des raisons de cet échec et de méthodes alternatives permettant de récupérer ce type d'information en Section 2.5.2. Les modèles de représentation de la ressource sont eux stockés dans la table *features_public*. Sans rentrer dans les détails, il faut retenir que dans l'architecture les tables *features_public*, *tools*, *experiments* et *experiments_results* stockent l'ensemble des modèles représentant les ressource. Elles ont été pensées dans le but d'être modulaires, c'est-à-dire de pouvoir rajouter ou supprimer des méthodes de représentation au fil du temps.

Avant de nous intéresser à la partie stockant les données utilisateurs, nous allons parler des tables : *urls*, *contains* et *providers*. L'objectif de ces tables est de stocker des informations telles que : « ces 3 ressources ont été accédées par le crawler depuis la même page ». Ce type d'information peut être intéressant notamment pour inférer des relations entre les ressources ; en effet, les urls identifiant les ressources étant des urls de téléchargement, il est fréquent qu'une même page référence plusieurs de ces urls. C'est le cas par exemple du site Videolectures.NET. Sur le site Videolectures.NET chaque page liée à une présentation contient par exemple un lien téléchargeable vers le diaporama, un autre vers la vidéo et parfois d'autres vers du contenu additionnel.

Enfin, les tables *users*, *cookies*, *user_activities* et *rec_user_transitions* ont pour rôle de stocker l'ensemble des interactions des utilisateurs sur les RELs. Ces interactions peuvent venir de deux sources distinctes, les sites partenaires implémentant le *connect service* (voir

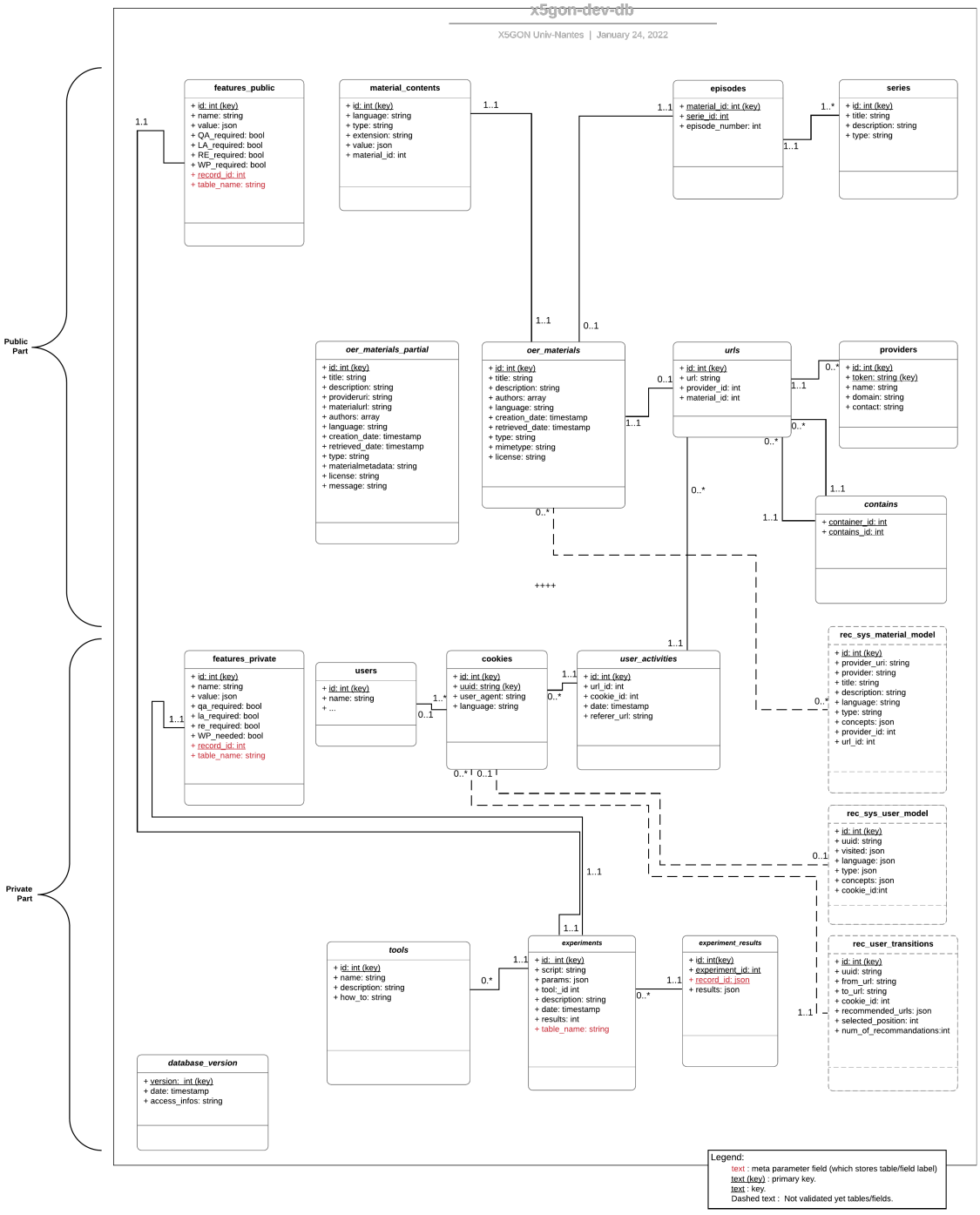


FIGURE 2.4 – Schéma d'organisation de la base de données du projet.

Section 2.3.2) et le *plug-in* de recommandation : X5RECOMMAND. Dans le cas des données provenant des sites partenaires, la seule information stockée concerne le passage d'un utilisateur sur une url spécifique à une date spécifique. Dans le cas du *plug-in*, l'information stockée est plus riche avec notamment l'ensemble de la liste de recommandation proposée et le choix fait par l'utilisateur. Les utilisateurs sont toujours identifiés par un cookie, en pratique sur les données stockées aucune information supplémentaire n'a pu être récupéré, ce qui conduit la table *user* à être vide. On remarque que les données utilisateurs récupérées ne sont pas aussi riches qu'attendues. C'est un problème important qui illustre la difficulté à récupérer des données utilisateurs externes au projet, ainsi que la difficulté à agglomérer des traces d'apprentissage composites. Nous discuterons de ces problématiques en Section 2.5.1.

Nous allons ignorer ici les tables *rec_sys_material_model* et *rec_user_model* qui n'ont pas d'intérêt sémantique et ont été utilisées afin de fluidifier les requêtes des systèmes de recherche et de recommandation.

2.3.2 *Connect service*

Le *Connect service* est un des points d'entrée des données utilisateurs dans le projet. Il a été développé par les partenaires Slovènes avec la contribution des différents partenaires pour s'adapter aux spécificités locales de chaque site. L'objectif de service est de connecter les sites contenant des RELs entre eux. Il permet aussi de répondre au manque récurrent de données de traces d'apprentissages des utilisateurs en fournissant un service d'agrégation des traces utilisateurs entre les sites. En pratique, le *Connect service* collecte à chaque nouvelle actualisation d'une page le cookie de l'utilisateur et l'url de la page de consultée. Les informations une fois collectés sont transmises à la base de données.

En particulier, ces données, une fois anonymisées, peuvent permettre des tâches d'évaluations hors-lignes des algorithmes de recommandation sur des traces d'apprentissage. Et ainsi stimuler une recherche basée sur les évidences dans le domaine de la recommandation à usage pédagogique. Il y a deux principaux points forts à cette approche agrégeant des données de différents sites. Premièrement, cela permet une robustesse dans les données récupérées, en effet, le caractère multisite rend les traces récupérées moins dépendantes de l'architecture du site, ou au moteur de recommandation implémenté. Deuxièmement, l'architecture multisite promet une plus grande variété d'apprenant et de pratique d'apprentissage permettant de mieux répondre à une tâche de recommandation non formelle sur des données non structurées.

Le *Connect service* permet donc à n'importe quel hébergeur de RELs, au prix d'une modification minime de son site web (littéralement deux lignes de JavaScipt), de connecter ses RELs au *Global OER Network* et de permettre la récupération des traces d'apprentissage de ces utilisateurs. Le détail de la procédure est disponible à l'adresse : https://www.x5gon.org/about/connect/.

Le *connect service* a été implémenté sur 4 sites, les 4 sites offrent une variété intéressante tant en termes de langues, d'utilisateurs que de pratiques :

VideoLectures.NET : VideoLectures.NET est le plus grand dépôt de vidéos universitaires en ligne au monde. Il est hébergé par l'Institut Jozef Stefan en Slovénie. La grande majorité du contenu concerne l'informatique, et plus particulièrement la science des données, le web sémantique, l'apprentissage automatique et les systèmes complexes. Les ressources proviennent de conférences avec revue par les pairs du monde entier telles que ICML, NIPS, ECML PKDD, SIGKDD. Le contenu s'adresse à des experts du domaine. Enfin, si l'on observe une diversité des formats et des langues, la grande majorité concerne du contenu vidéo en langue anglaise.

Učbeniki : Est un répertoire de ressource en langue slovène destinée aux élèves du secondaire. Parmi les matières proposées, on trouve l'algèbre, le slovène, la mathématique, l'anglais, l'allemand, la géographie, les sciences de la vie ou encore la physique et la chimie. Les ressources proposées sont des exercices sous forme de page web et permettant aux étudiants d'avoir des retours directs sur leurs réponses.

Université de Nantes : L'université de Nantes met à disposition un ensemble de cours en libre accès via son environnement numérique de travail Madoc (instance de Moodle). Les cours présents concernent l'apprentissage de l'informatique pour des élèves de niveau universitaire (licence et master) et sont en langue française. Malgré le faible nombre de ressources disponible, l'intégration de ce site est rendu particulièrement intéressante, car elle démontre la capacité du *connect service* à s'intégrer sur une ENA et en particulier Moodle.

Universitat politècnica de València : L'université polytechnique de Valence propose un service nommé Polymedia hébergeant des ressources provenant sous forme de vidéos réalisées par les enseignants de l'université de Valence. Les cours sont de niveau universitaire (license et master) et sont en langue hispanique (espagnol, catalan). Le service s'adresse directement aux étudiants de l'université de Valence.

La volonté du projet X5GON a été de voir le nombre de sites partenaires augmenté dans le but d'obtenir la plus grande diversité possible. Néanmoins, le projet a rencontré des

difficultés pour convaincre des hébergeurs de RELs d'implémenter le *connect service* sur leur site. Nous discuterons de ces difficultés en Section 2.5.3.

2.3.3 *Plug-in* de recommandation : X5recommand

Le *Plug-in* de recommandation : X5RECOMMAND, se présente sous la forme d'une liste de recommandation de REL provenant de multiples sites. Il est pensé pour être facilement implémentable sur tout site souhaitant connecter son contenu à l'écosystème des REL. Une fois implémenté, le *plug-in* fournit une liste de recommandation basée sur la ressource actuellement consultée par l'utilisateur, chaque site a ensuite la liberté de présenter cette recommandation comme bon lui semble, néanmoins un affichage standard épuré (voir Figure 2.5) est également fourni par le projet.

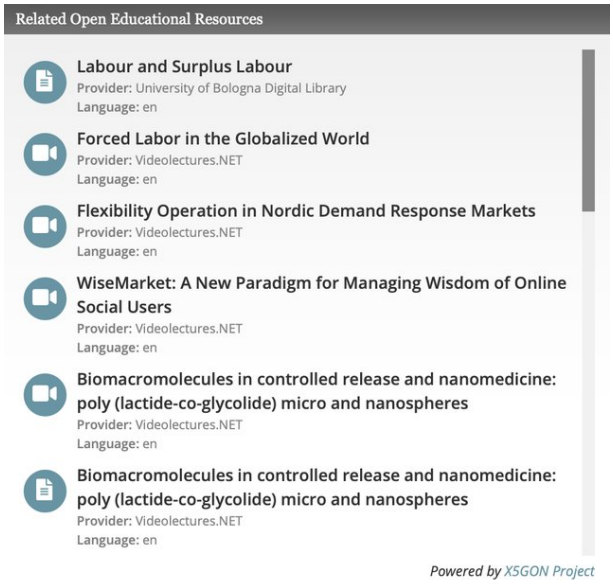


FIGURE 2.5 – Capture d'écran du *plug-in* de recommandation.

L'algorithme de recommandation utilisé est un algorithme entièrement basé sur les contenus, qui sélectionne les plus proches voisins de la ressource consultée dans l'espace de représentation TF-IDF ; la distance utilisée est une distance cosinus. Nous avons réalisé le développement de la méthode au sein de la cellule nantaise, la méthode a ensuite été évaluée lors d'une étude utilisateur portée par les partenaires de l'université d'Osnabruck. La méthode proposée, bien que simple, a été démontrée comme la plus pertinente par les utilisateurs, elle a en particulier surpassé des méthodes s'appuyant sur d'autres

représentations des documents (Doc2Vec, Wikifier, ...) et d'autres mesures de distance (euclidienne).

2.3.4 plate-forme à destination des utilisateurs

Durant le projet, plusieurs plateformes d'apprentissage web ont été développées, ces plateformes sont venues dans une deuxième étape du projet en réponse aux difficultés rencontrées pour récupérer des données utilisateurs externes. Néanmoins, dû à leurs développements tardifs, nous n'avons pas été à même de récupérer suffisamment de données utilisateurs pour pouvoir les exploiter dans le contexte de cette thèse. Ces plateformes restent néanmoins des pourvoyeurs de données utilisateurs prometteuses pour des travaux futurs.

L'objectif de ces développements est de proposer directement des outils d'apprentissage à des apprenants finaux.

Dans cette optique, deux plateformes ont été développées : X5LEARN et X5MOODLE.

2.3.4.1 plate-forme éducative : X5learn

La plate-forme X5LEARN est une plate-forme destinée à faciliter l'accès à des ressources éducatives en ligne gratuites disponible : `https://x5learn.org/`. La Figure 2.6 fournit une capture d'écran de l'interface présente sur le site. X5LEARN met à la disposition des utilisateurs un certain nombre d'outils pédagogiques permettant d'interagir avec des ressources éducatives ouvertes, ainsi qu'un ensemble d'outils adaptés aux préférences pédagogiques des utilisateurs. Il est destiné à aider aussi bien les enseignants que les étudiants. Le développement de la plate-forme a été réalisé par l'équipe de l'UCL (University College of London) et reprend bon nombre de développements collectifs réalisés au sein du projet, dont le *continuous wikifier* ou le système de recommandation basé contenu développé à Nantes.

Nous n'allons pas ici faire le détail de l'ensemble des technologies mises en place sur la plate-forme X5LEARN, le lecteur curieux peut notamment se référer à (Perez-Ortiz et al., 2021). Nous nous concentrerons sur une application de la méthode d'ordonnancement de ressource pédagogique que nous présenterons au Chapitre 4. La plate-forme offre notamment la possibilité aux enseignants de créer des listes de lectures de RELs partageables avec leurs étudiants, un service permet en particulier de réorganiser de manière automatique cette liste de lecture dans le but d'offrir la meilleure expérience d'apprentissage.

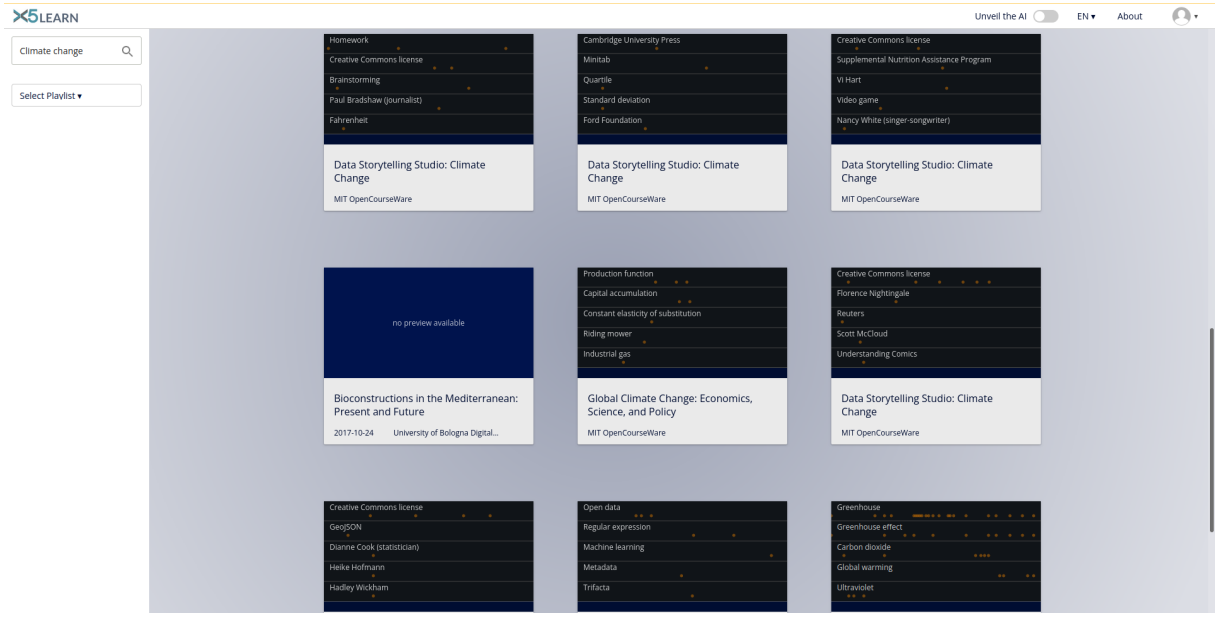


FIGURE 2.6 – Capture d'écran X5LEARN

Ce service que nous avons développé à Nantes lors du stage de Mathieu Vavrille en 2019 s'appuie directement sur la méthode TANN (présentée au Chapitre 4).

Au sein du projet, la plate-forme X5LEARN est un fournisseur potentiel de traces d'apprentissage et de retours d'informations riches en provenances des utilisateurs (pertinence de la ressource, qualité du contenu, qualité de la forme...).

2.3.4.2 Intégration à un environnement numérique d'apprentissage : X5moodle

Le X5-Moodle est un *plug-in* d'activité Moodle entièrement développé à Nantes, mis en œuvre sur la base de 2 idées clés : i) fournir des outils intelligents basés sur l'apprentissage automatique pour diffuser des ressources ouvertes en tant que support pendant la session de cours, et ii) faire évoluer les outils grâce à des approches basées sur l'utilisation collective plutôt que de se concentrer sur les données propres à un utilisateur. L'objectif de ce *plug-in* est de fournir aux enseignants et aux étudiants la meilleure expérience d'apprentissage.

Le *plug-in* reprend nombre de développements du projet, dont le système de recommandation basée contenu. Chaque instance du *plug-in* est liée à un cours publié par un enseignant via Moodle. Il est utilisable sous deux angles différents : celui de l'enseignant et celui des étudiants. Du point de vue des enseignants, le *plug-in* est principalement une interface de configuration dans lequel il peut paramétrer un moteur de recherche de RELs, le

moteur de recommandation ainsi qu'une liste de RELs que les étudiants devront consulter.

Une fois paramétré, les étudiants peuvent bénéficier des fonctionnalités du *plug-in*. Le moteur de recherche X5GON permet aux étudiants de rechercher de nouvelles ressources RELs. En outre, une liste des requêtes de recherche les plus fréquentes effectuées par les participants au cours est proposé aux étudiants. L'idée de cette liste est de favoriser la consultation des mêmes ressources, et ainsi inviter les étudiants à des réflexions collectives sur leurs contenus. La Figure 2.7 fournit une capture d'écran de l'interface du *plug-in*.

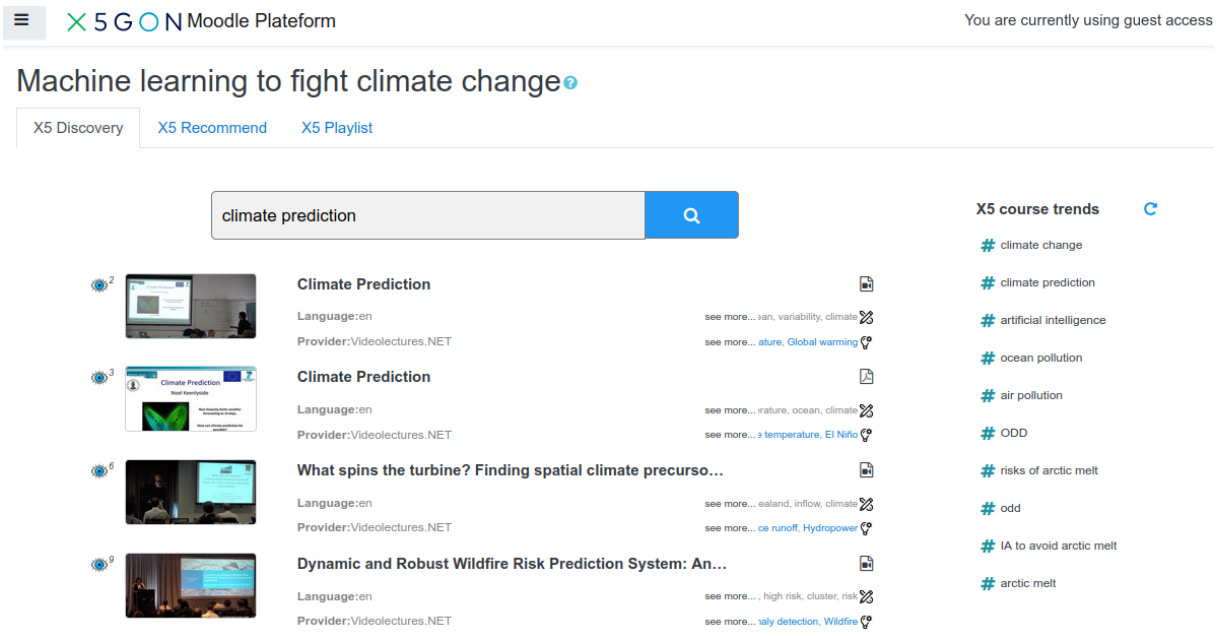


FIGURE 2.7 – Capture d'écran X5MOODLE

Une des particularités importantes du *plug-in* est d'étendre les fonctionnalités du moteur de recommandation pour construire une recommandation basée sur la popularité observée des RELs dans les interactions des participants au cours. Les recommandations évoluent systématiquement (avec les ressources populaires et les nouvelles ressources) en fonction de l'exploration des étudiants et de l'attention portée par la classe aux nouvelles REL. Cette recommandation destinée à un groupe présente l'avantage de favoriser les interactions sur les ressources consultées entre les étudiants et de créer une expérience d'apprentissage commune au sein de la classe. Elle permet notamment d'obtenir des informations sur la pertinence de certaines ressources dans un contexte particulier (niveau d'études, traitements du sujet, public cible). Une vidéo d'illustration du fonction-

nement du *plug-in* est disponible à cette adresse : https://mediaserver.univ-nantes.fr/videos/x5-moodle-ve/.

2.3.5 Dissémination

L'émulation de la recherche sur des algorithmes dédiés à l'exploitation des RELs et la promotion des pratiques de l'éducation ouverte ont été des objectifs importants du projet. En outre, la communication sur les technologies développées au sein du projet est un facteur clé pour motiver de nouveaux acteurs à devenir partenaire du projet et à connecter leurs ressources et leurs utilisateurs au réseau global de RELs.

Dans le but de déployer des outils clés en main stimulant la recherche sur les questions en lien avec l'éducation ouverte, les acteurs du projet ont mis en place deux APIs : i) une permettant un accès facilité au corpus de REL créé par le projet (MAIN API Section 2.3.5.1) ii) une autre donnant accès à des modèles intermédiaires dédiés aux RELs (LAM API Section 2.3.5.2).

Dans le but de communiquer sur les développements et d'impliquer de nouveaux acteurs dans l'écosystème de l'éducation ouverte plusieurs, le consortium a organisé et participé à plusieurs évènements scientifiques (Section 2.3.6) ainsi qu'un hackathon (Section 2.3.7).

2.3.5.1 MAIN API

La Main API est l'interface permettant d'accéder au corpus de REL collectées durant le projet ; l'url https://platform.x5gon.org/products/feed#api présente les différents services de l'API, ainsi que des exemples d'utilisations. Elle a été entièrement développée par la cellule slovène et s'interface directement avec la base de données du projet. Son rôle est de donner accès aux RELs au plus grand nombre et de manière uniformisée, elle vise des utilisateurs ayant volonté de récupérer des ressources éducatives. Pour cela, un premier ensemble de service fournit des ressources éducatives depuis leur identifiant X5GON. Cette récupération peut se faire ressource par ressource ou directement par bloc. Dans l'esprit de répertoire d'X5GON l'url originale, les informations relatives à l'auteur et à la licence sont également fournis, cela permet d'encourager les utilisateurs à donner les liens vers les auteurs originels des ressources pédagogiques même lorsque la licence ne le rend pas obligatoire. Enfin, le contenu textuel est également disponible dans les différentes langues dans le but de toucher une grande variété de fournisseurs et d'utilisateurs.

Le moteur de recherche et le système de recommandation présentés en Section 2.3.3 sont également disponibles via les différents services. Ils permettent notamment de sélectionner des sous-ensembles du corpus relatif à un concept, à un mot-clé spécifique, ou bien similaire à une ressource pré-sélectionnée.

La Main API vient en partie combler le manque observé de corpus de REL en fournissant l'accès à un corpus de ressources multilangue, multisite et multidomaine de taille conséquente ($> 100\,000$ ressources). Si l'on peut émettre quelques limitations sur le corpus proposé par la Main API (notamment la prédominance de l'Anglais que nous discuterons en Section 2.5.4), il reste que ce corpus est aujourd'hui le plus gros corpus de RELs en libre accès. De plus, l'API offre la possibilité rare de pouvoir sélectionner des corpus de RELs pertinentes pour un contexte ou un domaine donné. Pour cette raison, la Main API est une des contributions majeures du projet.

La Main API se cantonne à proposer les RELs dans leur format brut telles que récupérées sur les différents sites, mais ne donne pas accès aux représentations enrichies développées au sein du projet. Pour cela, une deuxième API a été développée : la LAM API.

2.3.5.2 LAM API

La LAM API est une API destinée à des utilisateurs ayant des connaissances préalables dans le domaine de la recommandation, de la représentation de document ou plus généralement de la recherche d'information. C'est une API que nous avons entièrement développée au sein de la cellule nantaise. Elle vise donc en particulier les développeurs de technologies éducatives en ligne. Ici, l'accent est mis sur la mise à disposition de modèles spécifiques pour le traitement des ressources éducatives. Toutes les informations et les liens utiles au sujet de cette API peuvent être trouvée à cette adresse : `https://gitlab.univ-nantes.fr/x5gon/lamapi`. Une version en production de l'API est disponible à : `https://wp3.x5gon.org/lamapidoc`. Pour cette raison, les outils sont pensés pour être appliqués sur des ressources indexées par X5GON mais aussi sur des ressources non indexées alors fournies par l'utilisateur sans nécessité de voir la ressource être ingérée dans le système. De ce point de vue, cette API ne se limite pas aux usages dans le contexte de l'éducation ouverte.

La littérature dans le domaine du traitement du langage naturel a démontré que l'utilisation de corpus spécifiques à un domaine améliore la tâche de représentations des documents. Un des premiers rôles de la LAM API est de tirer profit de la position du

projet X5GON ayant aujourd'hui référencé plusieurs milliers de RELs pour créer des modèles de représentation des ressources spécifiques à l'éducation. Au sein du projet, trois méthodes de représentations des documents sont utilisées : Wikifier, TF-IDF et Doc2Vec. Pour deux d'entre elles (Doc2Vec et TF-IDF) une étape préalable qui s'applique sur l'ensemble du corpus est nécessaire au calcul des représentations sémantiques des ressources. Ces étapes tirent grandement profit de corpus spécialisés comme celui d' X5GON, chacune des méthodes fournit des modèles permettant de représenter des documents hors corpus. Pour Doc2Vec c'est un modèle neuronal, dans le cas de TF-IDF c'est une liste de mots-clés pertinents dans le cas du corpus.

L'API fournit un premier ensemble de services permettant de récupérer une représentation sémantique d'une ressource indexée ou d'un texte libre. Les représentations proposées sont les six représentations présentées dans la Section 2.3.1.2. Pour chacune des représentations, il est également possible de récupérer les plus proches voisins d'une ressource ou d'un texte donné dans l'espace de représentation.

Un autre ensemble de services propose des outils de pré-traitements textuels. De la même manière que la représentation des documents, les méthodes de l'état de l'art sur ces tâches de pré-traitements de données textuelles telles que la lemmatisation tirent profit d'un apprentissage sur un domaine spécifique. Pour cette raison, un service de lemmatisation, un service de suppression des mots vides et un service de phrasage sont proposés (voir (Mikolov et al., 2013b) pour plus de détail).

Un troisième ensemble de services permet de récupérer une mesure de difficulté en lien avec la ressource selon différentes méthodes.

Enfin, un dernier ensemble de services utilise le système d'ordonnancement de REL que nous présenterons au Chapitre 4. Un premier service permet d'ordonner dans un ordre cohérent un ensemble de REL donnée en entrée. Un deuxième service propose d'identifier une ressource susceptible d'être rajoutée dans une séquence de REL donnée. Et un troisième service recommande une ressource à supprimer dans un ensemble de REL donnée.

2.3.5.3 Liens complémentaires

Un notebook de prise en main des API est disponible à l'adresse `https://colab.research.google.com/drive/1_Fb2wCVZlc810POJsy5zG4I3HgKfdQ1i?usp=sharing` et de plus amples informations peuvent être trouvées sur le site du projet `https://platform.x5gon.org`.

2.3.6 Congrès

La participation ou la mise en place d'ateliers/workshops scientifiques a joué un rôle important dans la politique de diffusion du projet. Plusieurs ateliers thématiques (https://www.x5gon.org/event/) ont notamment été organisés dans le but d'intégrer de nouveaux partenaires dans le projet et de sensibiliser sur les possibilités offertes par les API.

2.3.7 Hackathon

L'Hackathon « AI for the Common Good : F'AI'R Education Hackathon » a été un évènement de médiation important dont le projet X5GON est l'initiateur [9]. Cet hackathon était soutenu par le « UK Science & Innovation Network » de l'ambassade britannique à Paris, l'ANCSSC des Nations unies (alliance des ONG et des OSC pour la coopération Sud-Sud), X5GON et l'UCL science informatique. Il avait pour but de permettre aux étudiants de plusieurs universités internationales de devenir acteurs de l'éducation ouverte. Les étudiants avaient pour rôle de proposer des solutions basées sur l'intelligence artificielle pour répondre à des problématiques en lien avec l'éducation ouverte ; pour ce faire, ils ont pu bénéficier du support des membres du projet X5GON. Ce support était d'une part technique grâce à l'accès aux différentes API développées par le projet, mais correspondait aussi à une expertise offerte par les partenaires dans le domaine. Les étudiants ont eu le choix entre trois thèmes :

1. L'échange de connaissances entre ONG : comment renforcer les capacités par l'échange de connaissances entre des organisations établies et moins accomplies et leur personnel ?
2. L'échange de connaissances dans le domaine de la santé : comment pouvons-nous renforcer les capacités là où l'accès à la formation et à l'éducation en matière de santé est limité ?
3. Échange de connaissances en ingénierie : comment pouvons-nous renforcer les capacités des apprenants par l'enseignement de l'ingénierie pour tous, incluant l'accès à des connaissances provenant des conférences internationales ?

Chacun des thèmes offrait aux étudiants la possibilité de répondre à des problématiques sociétales grâce à la mise en place de solutions d'éducation ouverte.

9. Site du Hackathon : https://chaireunescorel.ls2n.fr/2020/02/29/nantes-gagne-le-hackathon-fair/

L'hackathon s'est déroulé en deux parties. Une première partie de pré-sélection de trois équipes de candidats par université partenaire du projet. Cette partie s'est déroulée à des dates et avec des modalités différentes dans les universités de Nantes, de Ljubjana, d'Osnabruck et UCL. Durant cette partie, les étudiants se voyaient encouragés à développer des outils s'appuyant sur les API X5GON.

La deuxième partie était une finale qui s'est déroulée sur deux jours en février 2020 à l'ambassade britannique à Paris. Durant cette finale, les équipes ont pu tester leurs solutions grâce à un jeu de données de traces utilisateurs généré depuis les données recueillies dans le projet par l'équipe nantaise. Le jury final était composé d'un panel d'experts en charge d'évaluer l'utilité, la faisabilité et l'originalité des solutions proposées :

- Ha Cole : Directeur des Nouvelles Technologies, Microsoft Philanthropy (panel industriel),
- Dr Husna Ahmad : Autrice et membre de l'alliance des ONG et des OSC pour la coopération Sud-Sud (panel ONG),
- Dr Louisa Zanoun : Attaché principal pour la science et l'innovation de l'ambassade britannique à Paris (panel gouvernement),
- Sasha Rubel : Spécialiste du programme IA de l'UNESCO (panel internationnal),
- Professeur Kai-Uwe Kühnberger : Vice President pour la recherche et professeur en science cognitive de l'université d'Osnabrück (panel académique),
- Davor Orlic : Directeur des opérations de la fondation Knowledge4All (panel X5GON).

Deux des équipes nantaises : (i) « Buddhisteam » composée d'Aniss Bentebib, Camille-Amaury Juge et Vincent Kowalski et (ii) « Next Wave Learning » composée de Mohamed Reda Marzouk, Sofiane Elguendouze et Timothee Poulain ont respectivement remporté la première et la troisième places.

Du point de vue du projet, l'intérêt de cet évènement a pris plusieurs formes :

- tester l'efficacité pratique des API développées par le projet,
- inciter les étudiants à s'intéresser aux problématiques en lien avec l'éducation ouverte et leur permettre de prendre part à son développement,
- démontrer l'efficacité des outils développés au sein du projet.

Durant cet hackathon, nous avons eu la charge du développement des jeux de données : un jeu de données de RELs nommé « catalogue » utilisé durant la première étape de l'hackathon et un jeu de données de traces utilisateurs utilisé lors de la finale.

Pour l'élaboration du catalogue, nous avons directement utilisé les services proposés par la MAIN API et la LAM API.

Pour l'élaboration du jeu de données de traces utilisateurs, une étape nécessaire indispensable concerne l'anonymisation des données. Les données utilisées ont été celles recueillies grâce au *connect service* et l'algorithme d'anonymisation utilisé a été développé en local à Nantes et sera présenté au chapitre 3. Les deux jeux de données générés sont disponibles avec de plus amples détails sur ce dépôt : https://gitlab.univ-nantes.fr/x5gon/x5gon-hackathon-datasets.

2.4 Données récoltées durant le projet X5gon

Dans cette section, nous allons présenter un ensemble de statistiques au sujet des données récoltées durant le projet. Nous nous intéresserons ici en particulier aux données contenus stockées dans la base de donnée X5GON, et aux données utilisateurs provenant du *connect service*. Les autres données utilisateurs récupérées durant le projet, en particulier celles provenant de X5MOODLE et X5LEARN étant arrivées tardivement dans la chronologie du projet, elles n'ont pas pu être analysées et utilisées dans le cadre de cette thèse et ne seront donc pas traitées ici. Les données provenant du X5RECOMMAND ne nous paraissent quant à elles non utilisables en l'état.

Toutes les statistiques présentées ici sont en date du 21/02/2022 et sont susceptibles de changer dans le futur.

2.4.1 Données contenu

Les données contenu sont les REL provenant des sites visités par le *crawler* ; les 118490 RELs indexées proviennent de 22 sites différents que l'on peut regrouper en 4 grandes catégories d'acteurs :

Associations : Les acteurs de type associatif sont représentés grâce à trois répertoires : un répertoire français (Association de Cryptographie Théorique et Appliquée), et deux répertoires d'envergure internationale : OER Africa, Videolectures.NET. Ces trois répertoires fournissent majoritairement des ressources de niveau universitaire.

Université : Une majorité des répertoires proviennent d'initiative universitaire en particulier : INRIA, LIMSI, l'école centrale de Marseille, Nantes Université et Université Paris-Est Marne-la-Vallée pour les universités françaises ; MIT OpenCourseWare et OpenStax CNX (Rice University) pour les universités américaines ;

OpenLearnWare (Tecnische Universität Darmstadt) et Osnabrück Universität pour les universités allemandes et University of Bologna Digital Library en Italie. Les domaines les plus fréquemment représentés sont les mathématiques, la physique et l'informatique.

Entreprise : Deux répertoires sont portés par des entreprises : Cythelia enerrgy (entreprise française développant des solutions logicielles et matérielles pour répondre aux enjeux de la transition énergétique) et The Siemens Stiftung Media Portal.

Organisme public : Enfin, la quatrième catégorie de répertoire concerne des organismes publics souvent en lien étroit avec le milieu universitaire : AUNEGe, Canal-U pour les initiatives françaises ; eCampusOntario Open Library, Engage NY pour les initiatives américaines ; Učbeniki une initiative slovène et TIB AV-Portal une initiative allemande.

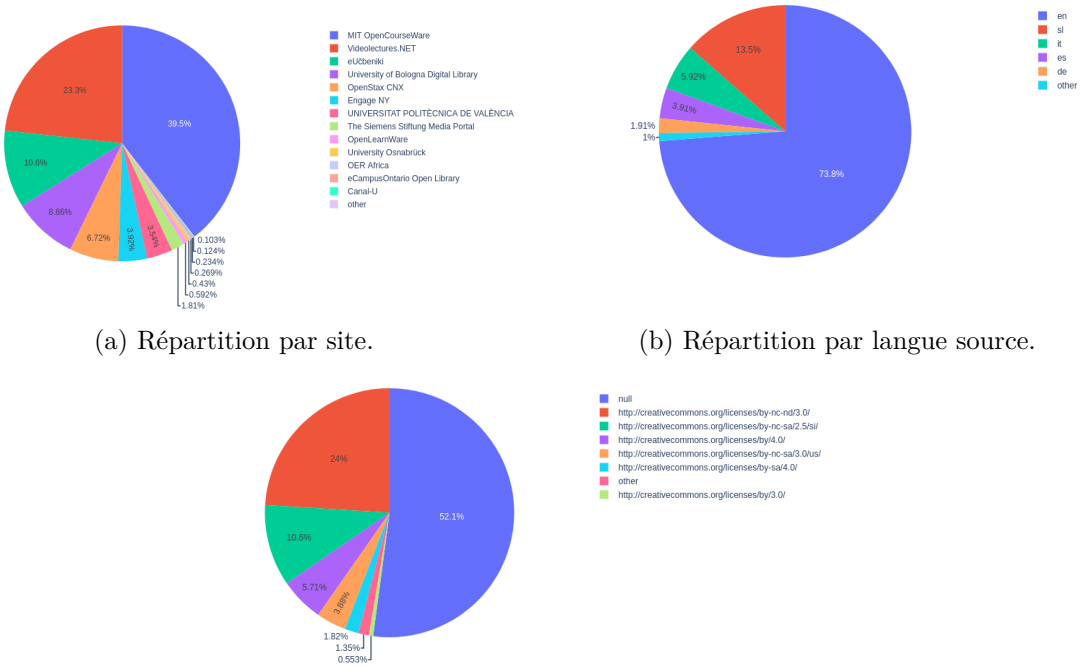
Si la liste des différents sites offre une variété géographique intéressante, certains continents ne sont pas ou peu représentés, en particulier l'Asie, l'Amérique centrale, l'Amérique du Sud, l'Océanie ou encore l'Afrique. De plus, le public cible d'une majorité des répertoires semble être un public averti, voire expert, peu de ressources, s'adressent au grand public ou aux enfants.

D'un point de vue quantitatif, malgré cette diversité, les acteurs les plus importants prédominent en nombre de ressources dans le corpus. Comme en témoigne la Figure 2.8a, les RELs provenant du MIT et Videolectures.NET représentent à elles seules plus de 50% des ressources.

De plus, on observe une majorité de contenu textuel (74 % pdf), web (17 % html) ou vidéo (10% mp4) au détriment d'autres types de contenu tels que les contenus audios.

Du point de vue multilingue, on observe une large majorité d'Anglais comme langue source (73% anglais, 13% slovène, 3.9% espagnol). La traduction semble en partie capable de compenser ce biais puisque du point de vue des langues disponibles, l'ensemble des langues intégrées dans le projet (en, sl, es, de, fr, it, ca, pt) présente un nombre de ressources comparables. Les autres langues intégrées plus tard dans le projet semble accuser un certain retard qui devrait se résorber avec l'avancement du *pipeline* de traduction.

Du point de vue des licences (Figure 2.8c), de manière étonnante pour une majorité de ces contenus - pourtant présentés comme ouverts - il reste impossible de détecter la licence de manière automatique. Ces contenus sont donc ouverts « dans l'esprit » car mis à disposition sur des répertoires de RELs, néanmoins ils n'ont pas de licence explicitement mentionnée et ne sont donc pas ouverts du point de vue légal. Parmi les contenus disposant



(a) Répartition par site.

(b) Répartition par langue source.

(c) Répartition par licence.

FIGURE 2.8 – Répartition du nombre de REL dans le corpus X5GON selon différent facteurs.

de licences, ceux sous licence Creative Commons sont majoritaires.

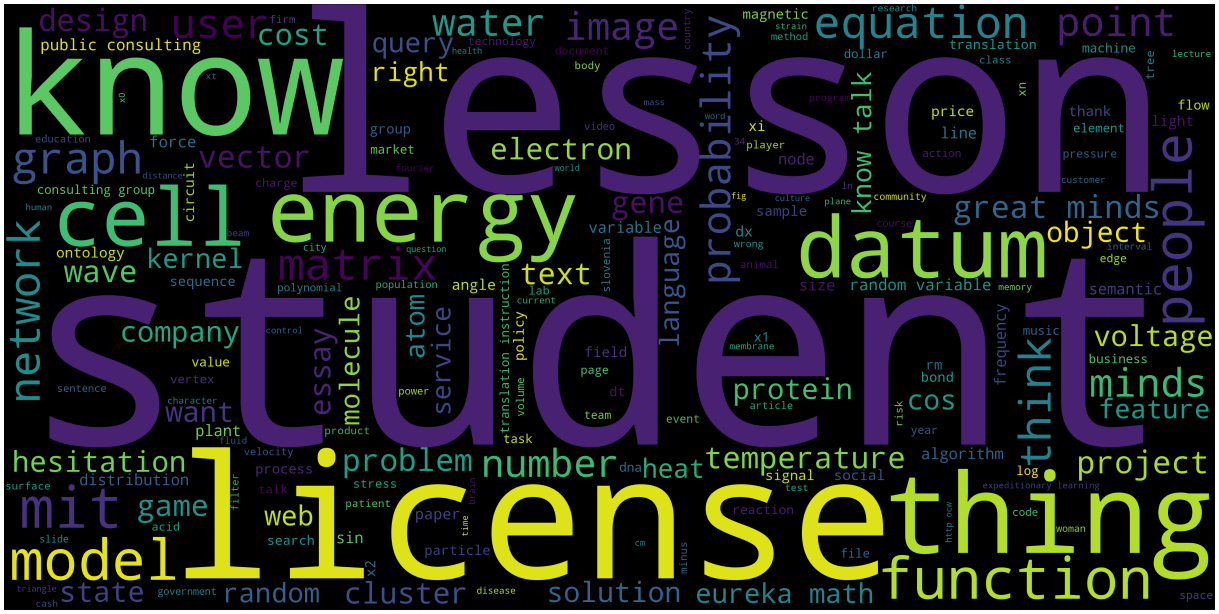


FIGURE 2.9 – Mots-clés les plus importants dans le corpus X5GON au sens du TF-IDF.

Enfin, on observe une prédominance nette de certains sujets ou domaines dans les ressources, en particulier les mathématiques, l'informatique et les contenus en lien avec l'éducation ouverte. Pour illustrer cela, la Figure 2.9 représente un nuage de mots dans lequel la taille d'un mot est corrélé avec son importance (au sens du TF-IDF) dans le corpus.

2.4.2 Données utilisateurs

Les données utilisateurs recueillies durant le projet peuvent être utilisées pour apprendre et caractériser le comportement des apprenants sur les plateformes éducatives. Sunar et al. (2020) proposent par exemple une analyse des comportements des utilisateurs à partir des données récoltées dans le projet. Ici, nous avons fait le choix de faire une analyse descriptive des données récupérées.

Les données utilisateurs provenant du *connect service* sont fournies par quatre partenaires : eUčbeniki, Videolectures.NET, l'université de Nantes et l'université de Valence. Elles concernent 50 millions d'accès à des RELs par des utilisateurs représentés par plus de 9 millions de cookies de session.

Les données recueillies sur le site d'eUčbeniki correspondent à plus de 95% des en-

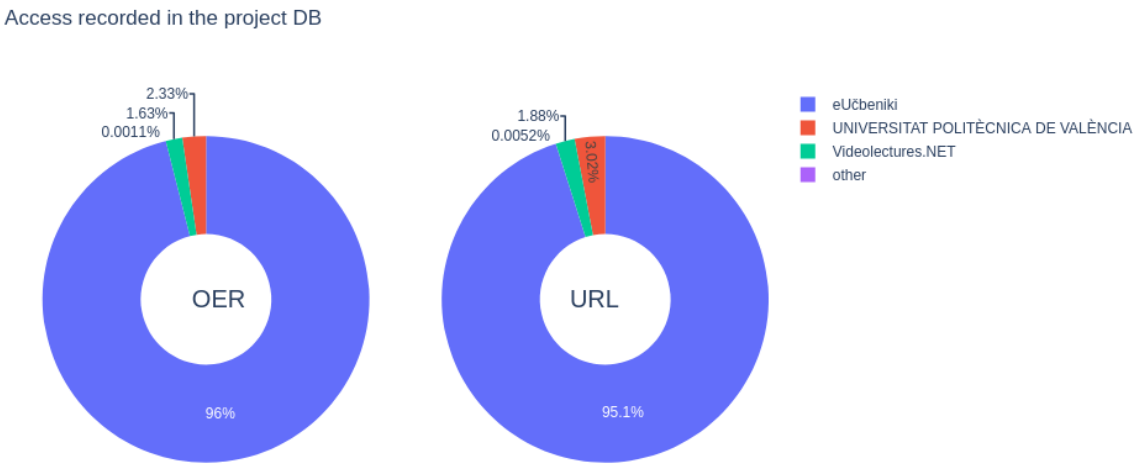


FIGURE 2.10 – Répartition des accès par nom de domaine.

registrements, ce qui s'explique par deux facteurs : (i) une forte affluence sur le site et (ii) la nature des REL hébergées. Sur le site d'eUčbeniki la majorité des REL sont des exercices prenant l'apparence de formulaires html typiquement composés de réponses à choix multiples. Cela signifie que chaque tentative de validation de l'exercice et donc de soumissions du formulaire entraînent un enregistrement dans la base de données du projet. À l'inverse, sur les sites des autres partenaires, les ressources sont statiques (texte, vidéo) entraînant mécaniquement moins de rafraîchissement de la page et donc moins d'enregistrements dans la base de données. Ce phénomène questionne la pertinence effective du stockage des interactions de différents sites de manière uniformisée, nous discuterons de cela en Section 2.5.4.

Indépendamment de ce biais, on remarque qu'une large majorité des cookies ne correspond qu'à un faible nombre d'accès, en particulier, 90% des cookies concerne 4 accès ou moins. En pratique, cela signifie qu'une majorité des utilisateurs a consulté 4 ressources ou moins.

Une majorité des apprenants semblent favoriser un apprentissage en journée durant la semaine (voir Figure 2.11). Cela s'explique en partie par le fait que dans le cas d'eUčbeniki et de l'université de Valence, les RELs sont utilisées durant des activités d'enseignement formelles en présentiel se déroulant à ces horaires.

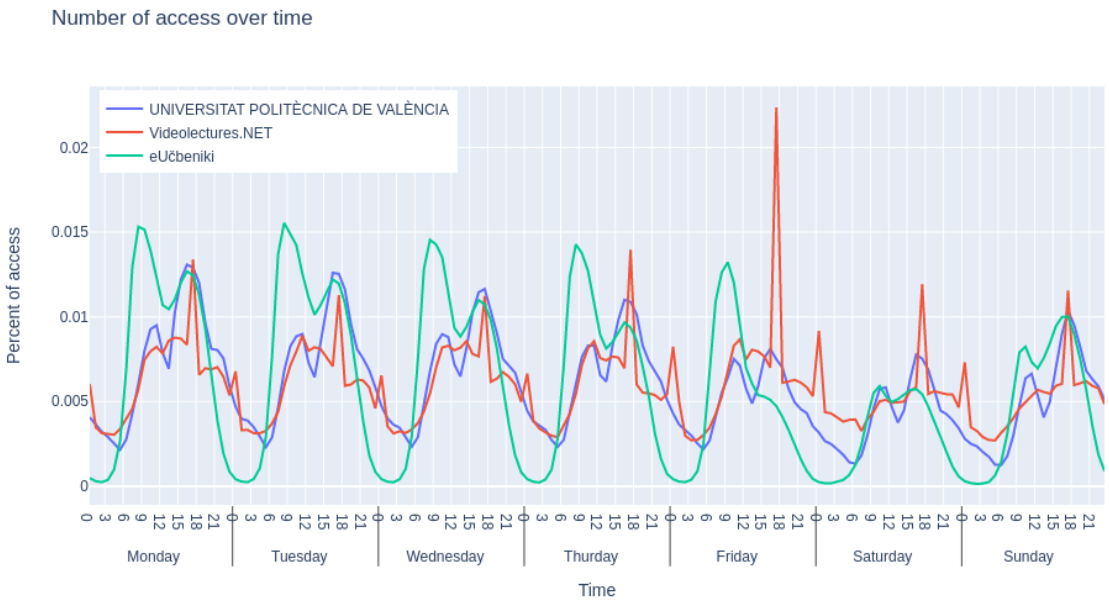


FIGURE 2.11 – Horaire d'apprentissage par nom de domaine.

Une des promesses d'X5GON était de connecter les dépôts de RELs entre eux. Comme en témoigne la Figure 2.12 cette promesse semble tenue pour les sites où le *plug-in* de recommandation est implémenté (navigation entre les domaines de Videolectures.NET, de l'université de Valence et l'université de Nantes). En revanche, pour les sites n'implémentant pas le *plug-in* de recommandation, ces interactions sont inexistantes (eUčbeniki).

2.5 Retour d'expérience

Le contexte du projet nous a permis de confronter les réflexions et les stratégies de départ aux contraintes réelles. Dans cette section, nous allons proposer un ensemble de remarques et de retours d'expériences pratiques dont des projets futurs pourraient tirer avantage.

2.5.1 Difficultés d'intégration des partenaires

Une des premières difficultés rencontrées par le projet concerne la difficulté d'intégration de nouveaux partenaires, cette difficulté est particulièrement prégnante dans le cas

FIGURE 2.12 – Quantification des navigations inter-domaines.

de la mutualisation de données utilisateurs. Plusieurs facteurs, dont certains identifiés en amont du projet, expliquent cette difficulté, en particulier, la méconnaissance légale et la méfiance en lien avec le partage de données utilisateurs. Néanmoins, d'autres facteurs non identifiés en amont du projet sont à considérer.

Premièrement, la politique d'X5GON visant à déployer une instance du *connect service* semble difficilement applicable dans le cas de petits sites n'ayant pas l'ossature légale, logistique et technique permettant de mettre en œuvre une politique de partage des données utilisateurs. Initialement, la stratégie d'X5GON pour palier à ce problème consisté à apporter une assistance technique et légale, tout en proposant des outils clé en main nécessitant peu de développement. En pratique, en dépit de l'assistance, un engagement en faveur du partage des données utilisateurs ne peut être pris à la légère et les petits acteurs ont du mal à consacrer du temps à une réflexion sur ce sujet. De plus, ils se trouvent directement exposé en cas de potentiel problème quant à la publication des données utilisateurs. L'hypothèse de poursuite juridique constitue pour eux un risque structurel important pouvant potentiellement mettre fin à leur activité. Même si ce type de poursuite est extrêmement rare, le danger encouru reste souvent trop important pour ces acteurs et la peur de faire une erreur semble prédominer. Enfin, si d'un point de vue macroscopique l'absence de ces petits acteurs est problématique, à l'échelle individuelle chaque acteur peine à voir

l'intérêt de partager ces données.

À l'inverse, pour de plus gros acteur, par exemple, tels que de grande université, le verrou principal semble politique. En effet, ces acteurs, grâce à leurs importances, ont les moyens de déployer une infrastructure technique et légale suffisante en faveur du partage des données utilisateurs. Néanmoins, pour ce faire, une stratégie globale et cohérente en faveur des REL est nécessaire. Dans beaucoup de cas, une politique en faveur de l'éducation ouverte ne semble pas une nécessité immédiate, de plus, le choix de l'ouvert ne fait pas toujours l'unanimité.

2.5.2 Limites de l'indexation semi-automatique

Au cours du projet, il a été observé plusieurs limitations quant à la stratégie de l'indexation semi-automatique. Une des promesses de cette stratégie concernait la possibilité de récupérer à moindre coût et rapidement un ensemble important de RELs. Pour ce faire, l'utilisation d'un *crawler* généraliste capable de s'adapter à chaque site était indispensable. Dans la pratique, dû à la large variété des usages et des pratiques d'organisation des sites, l'utilisation d'un *crawler* complètement autonomes s'est avéré impossible. Le choix a donc été fait d'utiliser un *crawler* paramétrable, ce choix nécessitant une intervention experte avant l'indexation de chaque nouveau site, il a mécaniquement négativement impacté la vitesse d'indexation des RELs. De plus, le paramétrage par l'expert devenant le goulot d'étranglement du processus, il convenait alors de maximiser le nombre de REL indexées pour chaque paramétrage. Cette incitation ayant pour conséquence d'encourager l'indexation de gros répertoires de RELs au détriment des petits ou des REL isolées explique pour partie la sur-représentation des RELs en langue anglaise et des dépôts occidentaux dans le projet.

Un des aspects négatifs identifiés en amont du projet du choix de l'indexation semi-automatique et généraliste était l'impossibilité de récupérer des métadonnées riches sur les RELs. En effet, les métadonnées présentes sur les sites étant très hétérogènes, elles contraignent une indexation automatique à récupérer uniquement un ensemble de métadonnées très restreint, afin de conserver un stockage et une chaîne d'ingestion efficace métadonnées Parmi ces métadonnées, le projet X5GON avait choisi de se concentrer sur cinq champs principaux : la licence, le titre, l'auteur, l'url et la langue. En pratique, même avec un choix aussi restrictif sur le nombre de champs, nombre de ressources indexées contenaient des champs vides, notamment la licence, l'auteur et la langue. Dans le cas de la langue, une méthode palliative utilisée permettait la détection automatique de

langue depuis le contenu et a permis de résoudre partiellement cette absence. Dans le cas de la licence et de l'auteur, cette absence est beaucoup plus problématique, car elle questionne la possibilité de réutiliser convenablement la ressource avec les bonnes pratiques d'attribution au cœur de la philosophie ouverte. En effet, sans l'information de l'auteur et de licence, il est alors impossible de correctement mentionner l'attribution de l'œuvre et de la réutiliser pour créer un nouveau contenu.

De plus, le faible nombre de métadonnées renseignées rend impossible la collection automatique de manière généraliste et automatique des liens sémantiques entre les ressources. Parmi ces liens, on distingue des informations telles que : « cet exercice correspond à cette partie du cours » ou « cette image illustre la notion présentée dans ce chapitre ». Cela est dommageable, car ces informations pourraient être fort utiles pour des tâches spécifiques de recommandation. Par exemple, une tâche visant à recommander un exercice en lien avec une partie d'un cours, ou encore, une tâche visant à recommander une illustration en lien avec un billet de blog. Initialement, la stratégie d'X5GON prévoyait de récupérer au moins une partie de ces liens, à savoir les liens de séquentialités entre les RELs. Ces liens permettant de capturer des informations telles que : « cette REL est la quatrième conférence de ce cours ». Ces informations devaient ensuite être utilisées comme jeux de données de validation pour les algorithmes d'ordonnancement au cœur des systèmes de recommandation. Dans la pratique, récupérer ces informations s'est avéré particulièrement difficile, car les sites contenant des ressources pédagogiques exhibent une diversité encore plus importante que celle envisagée au début du projet. En particulier, certains sites sont des catalogues de RELs. Au sens où ils proposent à leurs utilisateurs de naviguer à travers leur catalogue et prennent par conséquent en charge le moteur de recherche, le système de recommandation et les différents outils permettant une navigation facilitée pour l'apprenant. En particulier, ces sites comportent souvent des taxonomies susceptibles de contenir les informations qui nous intéressent. Sur ces sites, les REL sont souvent présentées de manière uniformisée. À l'inverse, d'autres sites se présente comme des répertoires, c'est-à-dire une liste de lien externe vers des RELs. Sur ces sites, les ressources ne sont pas présentées de manière uniforme et aucun outil n'est mis en place pour faciliter la navigation de l'apprenant. Dans le cas des catalogues, le sur-coût d'adaptation du *crawler* pour récupérer les liens de séquentialités est important, mais envisageable grâce à une cohérence organisationnelle commune des ressources. Sur les répertoires, la seule manière de procéder confine à un développement spécifique du *crawler* ressource par ressource, presque équivalent à une annotation manuelle.

Enfin, la tâche d'identification d'une REL de manière automatique est aujourd'hui une tâche complexe non résolue. Tous d'abord, il est difficile d'identifier la licence automatiquement et donc de déterminer si le contenu est ouvert ou non, ensuite, il est difficile d'identifier automatiquement si un contenu est à usage éducatif ou non. En particulier, cela donne lieu à des contenus récupérés en tant que RELs qui ne sont en pratique pas des RELs comme des listes d'émargements par exemple. Par ailleurs, cela introduit une autre intervention humaine dans le processus d'indexation, lors du choix des sites à indexer. Cela introduit un biais en faveur des gros répertoires pour la même raison de rentabilité de l'intervention humaine que celle mentionnée dans le cas de la contrainte de paramétrage. De plus, il introduit un autre biais, en effet, seuls les répertoires connus par là où les personnes en charge de définir la liste de domaine à indexer sont susceptibles d'être indexés. Par ailleurs, la charge de la politique d'indexation repose alors sur un faible nombre de personnes. Cela explique en partie la sur-représentation de domaines tels que l'informatique ou l'éducation ouverte dans le corpus X5GON, en effet, les membres du projet étant des acteurs actifs de ces communautés, ils sont naturellement mieux informés sur les répertoires disponibles sur ces sujets.

Malgré ces limitations, l'indexation semi-automatique présente des avantages indéniables. D'abord, elle a permis d'indexer de nombreuses REL en déployant peu de moyens. Ensuite, elle repose sur un *pipeline* d'ingestion dans lequel différents modules peuvent facilement être intégrés au fil du temps. Parmi ces modules, on peut supposer des développements futurs tels que des méthodes automatiques d'inférences de licence, ou des détecteurs automatiques de contenu éducatif. Néanmoins, notre recommandation consiste à complémenter la méthode automatique avec de l'indexation participative permettant de compenser les problèmes que nous venons de mentionner. C'est en suivant cette recommandation qu'a été lancé le projet Florilège dont nous parlerons en Section 2.6.1.

2.5.3 Limite de la méthode de récupération des données implémentée dans *connect service*

Nous l'avons vu en Section 2.5.1 durant le projet et dû à la difficulté d'intégration de nouveaux partenaires, le choix a été fait de simplifier un maximum la méthode de récupération des traces utilisateurs du *connect service* afin de faciliter son implémentation sur les sites partenaires. Ce choix a abouti à une méthode de récupération générique des traces d'apprentissages, cette méthode de récupération indépendante de l'architecture du

site a eu des conséquences inattendues sur les données récupérées. Pour comprendre le problème, commençons par décrire brièvement la méthode de récupération implémentée, l'idée générale de cette méthode consistait à récupérer l'url accédée à chaque rafraîchissement de la page. Un rafraîchissement de la page peut se produire dans trois situations bien spécifiques :

1. lors de la saisie manuelle d'une nouvelle dans la barre d'adresse ou en suivant un lien hypertexte,
2. lors de la soumission d'un formulaire,
3. lors du téléchargement d'un contenu via un lien de téléchargement.

L'objectif de la méthode consistait à pouvoir capturer deux informations : l'url accédée et le temps de consultation de la ressource.

En pratique, cette méthode ne permet pas toujours de récupérer le temps de consultation de la ressource, en effet, le temps effectif durant lequel le lien est ouvert reste un indicateur, mais ne permet pas de connaître le temps de consultation réel. Par exemple, un utilisateur s'éloignant de son écran aura le même temps de consultation calculé qu'un utilisateur assidu. Une apprenante ayant pour habitude de garder les RELs dans des onglets de son navigateur pendant de longues périodes aura des temps de consultations calculés ne correspondant pas aux temps de consultations réels. De la même manière, pour les contenus téléchargés, le temps de consultation calculé ne correspond pas au temps de consultation réel, nous n'avons en pratique aucune informations sur le comportement de l'utilisateur une fois la ressource téléchargée.

Une autre limitation importante de la méthode de récupération implémentée concerne la dépendance à l'architecture du site. Un exemple typique concerne le cas d'eUčbeniki. Comme mentionné en Section 2.4.2 ce répertoire contenant une majorité d'exercices sous forme de formulaire html, il obtient mécaniquement un nombre bien plus important d'accès enregistré que des répertoires de vidéos tels que Videolectures.NET.

L'ensemble de ces limitations questionne la pertinence effective du stockage des interactions de différents sites de manière uniformisée. Néanmoins, l'avantage principal de la méthode est d'avoir démontré la faisabilité d'une politique de mutualisation des traces d'apprentissage. Notre recommandation pour le futur serait de conserver cette politique de mutualisation, mais en déployant des méthodes de récupération des traces d'apprentissages différentes selon l'architecture des sites.

2.5.4 Bilan sur les données recueillies durant le projet

Les données contenu créé par X5GON représentent un pas important en faveur d'une recherche ouverte et basée sur des évidences pour la question de la recommandation à usage éducatif. Néanmoins, ces données ne remplissent pas pleinement les objectifs initialement fixés dans le projet, nous retenons trois remarques principales : (i) le manque de diversité dans les données contenu recueillies (ii) la question du partage des données utilisateurs à des fins de recherche (iii) la difficulté d'obtention de jeux de données de validation.

2.5.4.1 Manque de diversité dans le corpus X5gon

Le corpus de REL recueilli par X5GON présente une diversité intéressante relativement à l'absence de corpus similaire dans la littérature. Néanmoins, sur bien des aspects, la diversité escomptée n'est pas atteinte. Nous distinguons quatre axes sur lesquelles se concentrer pour améliorer la diversité du corpus. En premier lieu, on observe une nette sur-représentation de RELs provenant de culture occidentale et majoritairement en langue anglaise.

Deuxièmement, la prédominance de certaines disciplines, en particulier les disciplines scientifiques, et l'absence d'autres, en particulier les disciplines rattachées aux sciences humaines (droit par exemple) est problématique.

Troisièmement, certains types de contenus tels que les contenus vidéos sont largement majoritaires dans le corpus au détriment des contenus audio et des images.

Enfin, les contenus recueillis sont de niveau universitaire et s'adressent à des experts, à l'inverse, on observe un manque contenu de vulgarisation, de contenu destiné aux enfants ou de contenu d'éducation populaire.

2.5.4.2 Partage des données utilisateurs

À travers le *connect service* X5GON a réussi à recueillir et mutualiser plus de 50 millions d'accès d'utilisateur sur des REL. À travers les autres outils s'adressant directement aux utilisateurs tels que : X5LEARN, X5MOODLE et X5RECOMMAND, X5GON devrait dans le futur être capable de collecter des traces d'apprentissage plus riches, plus diversifiées et plus nombreuses. L'objectif de la récupération de ces données est de fournir une expérience d'apprentissage collective plus enrichissante pour les utilisateurs en promouvant la recherche sur la recommandation à usage pédagogique. Pour ce faire, un point

critique concerne la publication de jeux de données provenant de ces traces d'apprentissages, l'étape préalable à cette publication concerne l'anonymisation de ces données. À ce jour, la quête d'une méthode d'anonymisation correcte est un sujet de recherche et les travaux que nous présenterons au Chapitre 3 présentent un algorithme utilisable en pratique pour des données séquentielles ayant les spécificités des traces d'apprentissage recueilli. Au cours de l'hackathon X5GON, un jeu de données généré à partir des traces d'apprentissages du projet a même pu être utilisé ; la mise en place d'une stratégie de publication des traces d'apprentissage n'a pas pu être mise en place dans le projet dû à la difficulté rencontrée pour collecter les données. C'est un enjeu important pour le futur de mettre en place tant du point de vue technologique que juridique une solution permettant de partager ces données.

2.5.4.3 Absence de jeu de données de validation

L'obtention de jeux de données de validation était un objectif initial du projet X5GON. Néanmoins, par la difficulté d'intégration de partenaires, X5GON a fait le choix de simplifier au maximum l'incorporation du *connect service*. Cette simplification a conduit à minimiser les données recueillies par le *connect service* et donc la richesse des méta-données. Par ailleurs, les difficultés rencontrées par le *crawler* pour récupérer les informations de lien séquentialité entre les ressources de manière automatique n'ont pas permis de compenser ce manque de méta-données. Par conséquent, aucun jeu de données de validation n'a pu être récupéré tel qu'envisagé au début du projet. Pour palier à ce manque, nous avons fait le choix de récupérer un corpus de validation grâce au site YaleOpenCourseware, ce corpus est présenté en Section 2.6.2.

2.6 Développements additionnels

2.6.1 Florilège

Le projet Florilège est un projet visant à créer un jeu de données de RELs francophones et tirant profit d'une annotation participative pour obtenir des métadonnées plus riches et de meilleure qualité. Ce projet offre une solution aux difficultés rencontrées par X5GON à la fois pour récupérer des jeux de données de validation (comprendre ici des séquences de ressources prédéfinies) et pour inclure plus largement des ressources françaises. De plus, par son système d'annotation et de référence collaboratif, Florilège ambitionne un

référencement plus large, en particulier, en référençant également les petits répertoires de RELs ou des REL isolés.

Sur la plateforme Florilège, chaque utilisateur se voit invité à annoter un ensemble de RELs sur la base de ces intérêts. L'annotation en elle-même utilise un plug-in de navigateur nommé « welearn » développé par le Learning Planet Institut (ex : Centre de recherches interdisciplinaires (CRI)) [10]. Le plugin-in propose d'annoter chaque page web avec un ensemble de concepts, Wikipédia et Hashtags. L'utilisateur doit alors parcourir la page web contenant la REL et annoté par l'intermédiaire du plug-in sur la base de ses observations. Parmi les Hashtags applicables à une page, Florilège ambitionne dans le futur la possibilité pour l'annotateur d'indiquer la limite de la ressource, son organisation interne, sa difficulté, le public qu'elle cible ou encore sa licence. L'annotation est alors récupérée par la plate-forme Florilège et sera à terme utilisé pour enrichir les méta-données des RELs. La plate-forme Florilège est aujourd'hui en développement, mais une version alpha est disponible en libre accès à cette adresse : https://florilege.ls2n.fr/catalog/.

Du point de vue des données de validation, l'annotation participative offre la possibilité de recueillir des retours utilisateurs et des informations sur les RELs difficiles à obtenir de manière automatique ou implicite. En effet, si délimiter le contexte et l'organisation interne d'un cours est une tâche difficile à réaliser de manière automatique, c'est une tâche relativement simple pour un annotateur humain. Parmi les questions pertinentes auxquels l'annotation participative peut répondre, on retrouve des questions partiellement traitées de manière automatique, par exemple, la difficulté d'une ressource, la licence ou les concepts engagés. Dans ce contexte, l'annotation participative vient en complément des méthodes automatiques. D'une part, les méthodes automatiques peuvent permettre de faciliter l'annotation en proposant par exemple un ensemble de concepts susceptibles d'être pertinents. D'autre part, l'annotation participative fournit des jeux de données permettant d'entraîner les méthodes automatiques et de les rendre plus performantes.

2.6.2 YaleOpenCourseware

Le projet de récupération du corpus YaleOpenCourseware est une initiative portée par la cellule nantaise. Parmi les objectifs mentionnés au début du projet figurait la nécessité de récupérer des données provenant d'études utilisateurs (Section 1.3.5). En particulier, des exemples de séquence d'apprentissage correcte validée par des experts devaient être

10. Plus d'informations sur WeLearn à : https://welearn.cri-paris.org/pages/onboarding.html

utilisées en tant que jeux de données de validation. Une des méthodes envisagée pour récupérer de tels jeux de données consistait à récupérer directement sur les répertoires de RELs de telles séries créées par des enseignants. Il peut typiquement s'agir d'un cours découpé en plusieurs parties séquentiellement liées, par exemple une série de conférences. Constatant la difficulté observée d'adapter le *crawler* pour récupérer les informations de séquentialités, nous avons choisi de récupérer un ensemble de séquences d'apprentissages créées par des enseignants sur le site du projet YaleOpenCourseware. Ce jeu de données sera notamment utilisé au Chapitre 4 pour entraîner le système d'ordonnancement de ressource pédagogique proposé en tant que service dans la LAM API et implémenté sur la plate-forme X5learn.

Le projet YaleOpenCourseware [11] fournit un accès libre et gratuit à une sélection de cours d'introduction enseignés en anglais par d'éminents professeurs et universitaires de l'université de Yale.

Chaque cours est composé d'une séquence de conférences ; les transcriptions manuelles et la division en chapitres sont disponibles pour chaque épisode.

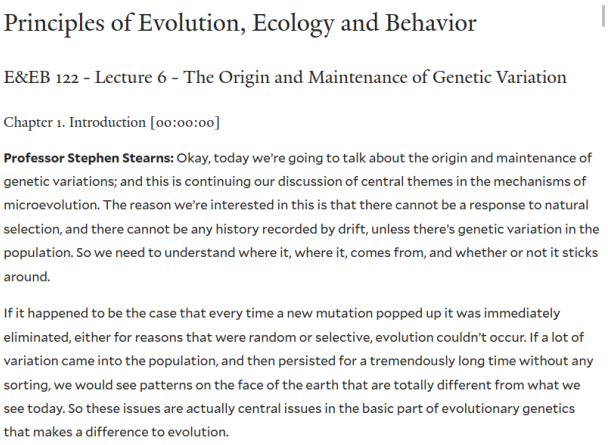
Le corpus moissonné à partir du site web du projet https://oyc.yale.edu/ contient 40 séries provenant de 36 enseignants différents pour un total de 1058 épisodes avec une moyenne de $26,45 \pm 4,8$épisodes/série. Les conférences ont une durée avoisinant les 45 mins et le format est celui d'un cours magistral suivi de question. Les conférences récoltées vont d'automne 2006 à automne 2011. Pour chacune des conférences, un ensemble de contenu additionnel ainsi que de références fournies par l'enseignant est disponible.

Le travail de moissonnage a nécessité le développement d'un *crawler* spécifique pour le site permettant de récupérer les transcriptions, l'organisation internes et les méta-données de chaque cours au préalable dispersés sur les pages. L'ensemble du *crawler* est une initiative de la cellule nantaise du projet. L'ensemble du code et les détails relatifs à l'implémentation sont disponibles à cette adresse : https://gitlab.univ-nantes.fr/connes-v/yaleocw-corpus.

La Figure 2.13 donne un exemple de texte brut provenant du corpus, en l'occurrence : la conférence numéro six intitulée « The Origin and Maintenance of Genetic Variation » du cours « Principles of Evolution, Ecology and Behavior » donné par le professeur Stephen Stearns en juin 2006.

Les sujets abordés proviennent d'une large variété de domaine et le niveau cible est

11. L'ensemble des données et des informations supplémentaires se trouvent sur https://gitlab.univ-nantes.fr/connes-v/yaleocw-corpus



Principles of Evolution, Ecology and Behavior

E&EB 122 - Lecture 6 - The Origin and Maintenance of Genetic Variation

Chapter 1. Introduction [00:00:00]

Professor Stephen Stearns: Okay, today we're going to talk about the origin and maintenance of genetic variations; and this is continuing our discussion of central themes in the mechanisms of microevolution. The reason we're interested in this is that there cannot be a response to natural selection, and there cannot be any history recorded by drift, unless there's genetic variation in the population. So we need to understand where it, where it, comes from, and whether or not it sticks around.

If it happened to be the case that every time a new mutation popped up it was immediately eliminated, either for reasons that were random or selective, evolution couldn't occur. If a lot of variation came into the population, and then persisted for a tremendously long time without any sorting, we would see patterns on the face of the earth that are totally different from what we see today. So these issues are actually central issues in the basic part of evolutionary genetics that makes a difference to evolution.

FIGURE 2.13 – Exemple de transcription provenant du corpus YALEOPENCOURSEWARE.

toujours celui d'élève de troisième cycle. La répartition des cours entre les différents départements est représentée dans la Fig. 2.14. On peut observer une variété de sujets avec une prédominance de l'histoire.

On observe également une nette différence dans le nombre moyen de conférences constituant un cours, notamment au niveau des domaines (Fig. 2.15). Cette différence ouvre la question de la nécessité d'apprendre un modèle par domaine ou un même modèle pour tous les domaines.

Avantages Les principaux avantages de ce corpus sont la disponibilité de séquences "correctes" faîtes par les enseignants, la disponibilité d'une transcription manuelle et d'un découpage en chapitres, ainsi que la variété des domaines traités.

Limitations La principale limitation est l'absence de variétés en termes de langue, de modalité et de format. Cette limitation met en exergue la difficulté de produire un jeu de données réunissant toutes ces contraintes. L'approche proposée par X5GON consisté en la récolte d'une large variété de jeux de données de séquences d'origine variées afin de pouvoir apprendre de cette diversité. L'agglomération de ces jeux de données formerait un corpus se rapprochant le plus possible du contexte de la recommandation à visée éducative dans un contexte informel. Le corpus YaleOpenCourseware constitue en cela une pierre à cet édifice.

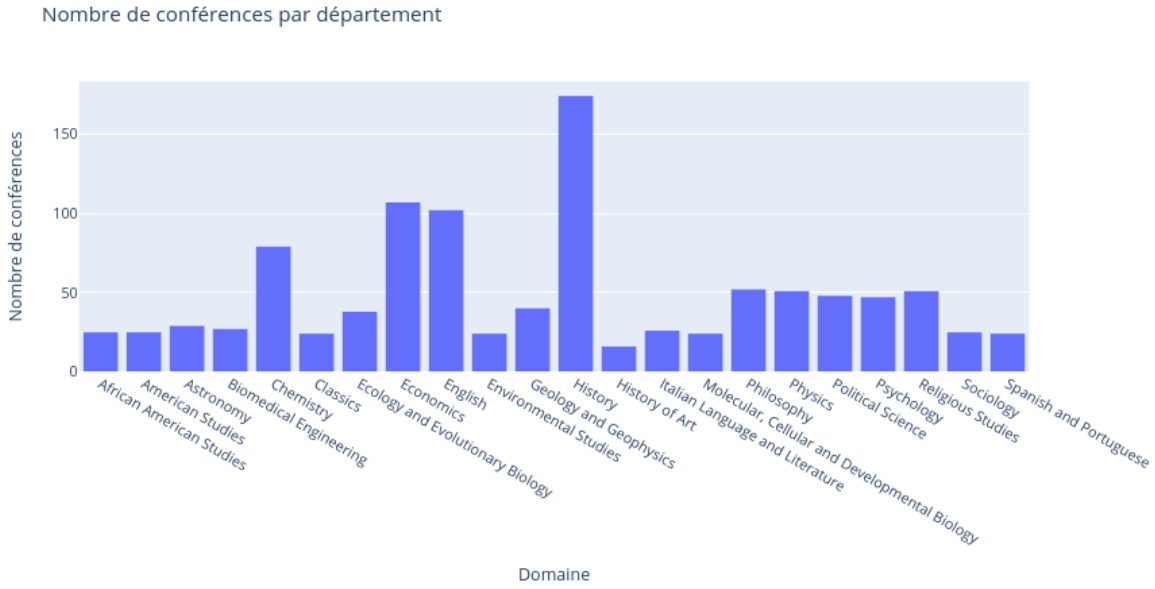


FIGURE 2.14 – Répartition des cours par département dans le corpus YALEOPENCOURSEWARE.

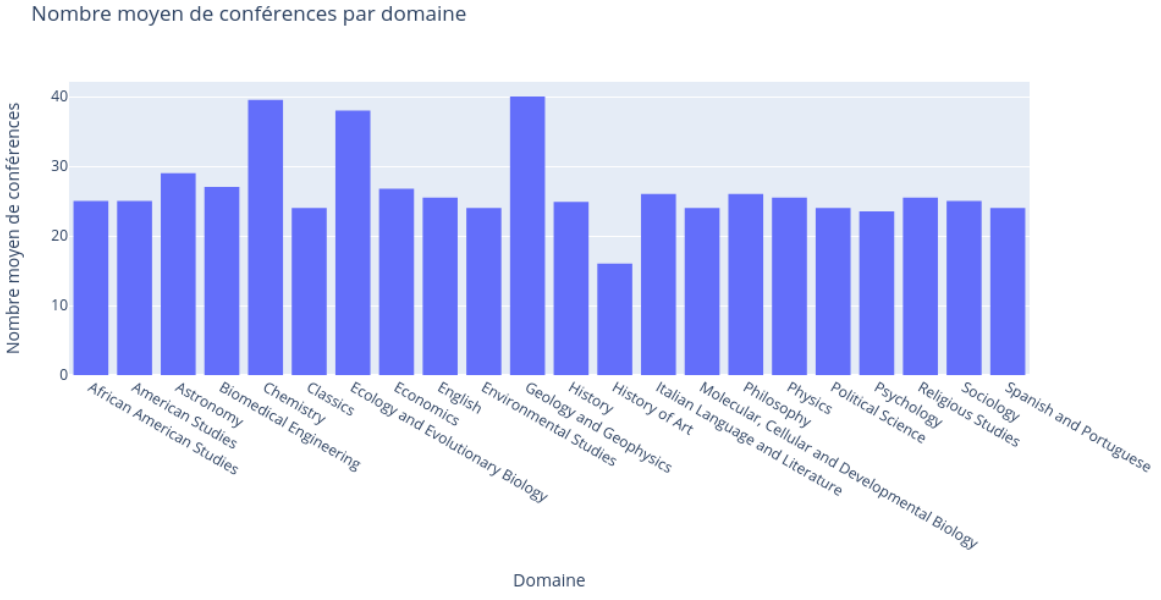


FIGURE 2.15 – Nombre moyen de conférences composant un cours par domaine.

2.7 Conclusion sur le projet X5gon

Dans ce chapitre, nous avons dépeint le contexte de lancement du projet X5GON, présenté les principales contributions du projet et fait un retour critique sur le travail accompli. La description du contexte nous a permis de mettre en lumière deux enjeux important de l'écosystème RELs. En premier lieu, la difficulté ressentie d'accès aux ressources (Wang and Towey, 2017; Belikov and Bodily, 2016) qui s'explique par l'éparpillement des ressources sur différents sites non connectés entre eux. En second lieu, les enjeux légaux des licences ouvertes (Amiel and Soares, 2016) qui sont de deux natures : la difficulté d'anonymiser les données utilisateurs qui engendre un déficit de jeu de données, de traces d'apprentissage et la position par défaut « tous droits réservés » qui complexifie l'accès et la réutilisation des ressources ouvertes.

Dans ce contexte, le projet X5GON se proposait de connecter les RELs entre-elles grâce à des méthodes automatiques, en particulier un système automatique d'indexation des ressources, mais aussi des systèmes de recherche et de recommandation de REL multisite. Fort des interactions utilisateurs et des RELs collectées grâce à ces technologies, le projet X5GON propose également une réponse au manque de jeu de données, de contenu, de validation et d'apprentissage observé dans le domaine. Après trois ans de projet, X5GONa réussi à produire le plus gros jeu de données de RELs multilingue a notre connaissance tout en fournissant un ensemble d'outils (API) permettant un accès facilité aux RELs. Du point de vue des jeux de données de traces d'apprentissage, le projet a réussi à collecter des interactions d'apprentissage multisite et à déployer des plates-formes d'apprentissage qui devraient dans le futur enrichir les données collectées. Un enjeu reste ensuite de partager ces données de manière publique et anonymisée, pour ce faire, nous proposons une contribution dans le chapitre suivant (Chapitre 3).

Sur la question des données de validation, l'idée centrale du projet consistait à tirer profit des méta-données en lien avec les RELs (difficulté, thématique...) ainsi que de leurs relations (séquentialité, lien, exercice-notions...). Dans les deux cas, la faible présence des méta-données et les limitations en lien avec les méthodes d'indexations n'ont pas rendu possible la collection de telles données. Pour palier a ce problème, nous pensons que l'annotation participative peut venir compenser l'indexation automatique et remplir les métadonnées manquantes. Cette complémentarité permettrait également de rendre les modèles automatiques - généralement basés sur des modèles d'apprentissage - plus performant. C'est dans cette optique que le projet Florilège a été lancé (Section 2.6.1).

De plus, dans le but de combler pour partie ce manque de jeu de données de validation, nous avons conçu le jeu de données YaleOpenCoursewareCorpus (Section 2.6.2) grâce à un travail de spécifique de récupération des cours du répertoire de données : `https://oyc.yale.edu/` porté par l'université de Yale. Ce jeu de données ouvre la voie à de nouvelles méthodes d'évaluations alignées sur la tâche de recommandation pédagogique pour les modèles d'apprentissage d'ordre qui sont la pierre angulaire de la recommandation à large-échelle. Nous explorerons ce point en particulier au Chapitre 4.

Chapitre 3

Méthode d'anonymisation de traces d'apprentissage

3.1 Introduction

L'accès à des jeux de données de traces d'apprentissages ouverts est une question cruciale pour promouvoir la recherche et les initiatives sur la recommandation à usage pédagogique. Ceci est d'autant plus vrai qu'il y a un manque de jeux de données à usage libre pour les données liées à l'éducation ouverte. Ce manque de jeux de données peut s'expliquer par plusieurs causes, au premier rang desquelles figure la nécessité de préserver la vie privée des utilisateurs (Section 1.4.2).

Dans l'état actuel du projet X5gon, les parcours d'apprentissages à travers les URL -et les REL correspondantes- sur différentes plates-formes sont collectés grâce au *connect service* (Section 2.3.2) mis en place sur les différents sites partenaires du projet. Chaque utilisateur le souhaitant peut fournir son activité d'apprentissage à la plate-forme. Ces informations sont utilisées ultérieurement pour fournir des recommandations personnalisées et plus généralement une expérience d'apprentissage plus satisfaisante à l'utilisateur. Par sa position, le projet X5gon est un excellent candidat pour construire un jeu de données ouvert de séquences d'apprentissages.

La publication de jeux de données de traces utilisateurs ouverts nécessite une méthode d'anonymisation permettant de garantir l'anonymat des utilisateurs ayant fourni leurs traces. Cette anonymisation est à la fois un impératif pratique et légal fort. Néanmoins, aucun algorithme connu ne peut à ce jour garantir une protection totale contre la ré-identification en préservant la pertinence des données (pour plus de détails sur les enjeux de l'anonymisation et le risque de ré-identification, voir la Section 2.1.4). Ainsi, le processus d'anonymisation est un compromis constant entre le risque de ré-identification et l'utilité des données publiées. L'utilité se définit informellement par la capacité des données à répondre à des questions. Aux deux extrémités de ce compromis, un algorithme

maximisant l'utilité revient à publier directement le jeu de données et à ne fournir donc aucune protection aux utilisateurs. À l'opposé, un algorithme annulant le risque de ré-identification conduit à la publication d'un jeu de données aléatoire inutilisable. L'objectif d'une méthode d'anonymisation est de conjuguer une bonne utilité des données publiées avec un risque de ré-identification suffisamment faible pour être acceptable. Les données séquentielles n'échappent pas au risque de ré-identification.

Une nouvelle définition mathématique de la confidentialité introduite en 2004 (Dwork and Roth, 2013) fournit une garantie théorique, mais probabiliste contre ce type d'attaque. Elle est considérée comme la définition la plus forte de la confidentialité par les experts. L'application de cette définition à des données séquentielles est difficile et a tendance à fournir des jeux de données peu utilisables (Abay et al., 2019).

Dans ce chapitre, nous tentons de nous inspirer de cette définition et de tirer profit de l'efficacité connue des méthodes à base d'automates pour la modélisation de séquences, afin de proposer une méthode permettant d'anonymiser des jeux de données de séquences tout en préservant une bonne utilité.

Pour ce faire, nous suivrons une approche de publication de données utilisant un modèle génératif appris sur les données réelles (ici les traces d'apprentissage). En l'occurrence, nous utiliserons un modèle d'automate temporisé constructible à partir du k-test vecteur des séquences introduit pour l'occasion. Cet automate permettra de modéliser des jeux de données de séquences en conservant les dépendances courts termes et l'information temporelle. Dans le but d'assurer la confidentialité des données générées, nous introduirons un algorithme d'élagage de l'automate exposant des garanties de confidentialités. Enfin, nous montrerons la pertinence de notre approche sur les données d'apprentissages recueillies grâce au *connect service* développé dans le projet X5GON. Les résultats démontreront la pertinence de l'approche au global pour créer des jeux de données proches des données réelles et en particulier conserver les dépendances court terme indispensables dans le contexte de la recommandation.

L'organisation du chapitre suit l'architecture suivante. Dans un premier temps, nous introduisons les outils théoriques pertinents provenant du domaine de l'inférence grammaticale (Section 3.2, 3.3) et proposons un modèle d'automate original permettant la modélisation de ces séquences. Puis, nous proposons une approche d'élagage de l'automate offrant une garantie de confidentialité (Section 3.4.4), enfin, nous démontrons la pertinence de cette approche, notamment en termes d'utilité sur les données de traces d'apprentissages issues de X5GON (Section 3.5).

3.2 Notations

Nous notons $\mathbb{N}$ pour désigner l'ensemble des entiers naturels, et utilisons i, j, k, m, et n comme éléments de $\mathbb{N}$.

Nous appelons *alphabet* un ensemble de cardinal fini de *symboles* noté Σ. Σ=$\{a, b, c \ldots\}$. Un *mot* $x = a_1 \ldots a_n$ est une séquence finie de ces symboles. La *longueur* d'un mot notée $|w|$ correspond au nombre de symboles dans le mot. Le mot vide (de longueur 0) est noté λ. Nous notons Σ^* l'ensemble de tous les mots issus de Σ, et Σ^+ l'ensemble de tous les mots non vides issus de Σ (c.-à-d. $\Sigma^* = \Sigma^+ \cup \{\lambda\}$). Dans le même esprit, $\Sigma^{<i}$, Σ^i et $\Sigma^{>i}$ désignent respectivement l'ensemble des mots issus de Σ de longueur inférieure à i, égale à i, et supérieure à i.

Étant donnés deux mots u et v, on note $u \cdot v$ la concaténation de u et v. Fort de cette notation, u est un *préfixe* de v si et seulement s'il existe un mot w tel que $u \cdot w = v$. Et u *suffixe* est un de v si et seulement s'il existe un mot w tel que $w \cdot u = v$.

Dans la suite, nous utiliserons les symboles $\{a, b, c, d\}$ comme lettres de l'alphabet dans nos exemples. En opposition, nous utiliserons les symboles $\{x, y, z\}$ comme des variables, nous permettant par exemple d'itérer à travers un mot ou un alphabet.

3.3 Quelques définitions

Les données correspondant aux traces des utilisateurs peuvent être définies et récupérées de manières très différentes ; pour la suite, nous utiliserons les définitions suivantes.

Un évènement Nous appellerons un évènement l'accès à un temps spécifique à une URL. Ainsi un évènement est un 2-tuple (s_i, d_i) où s_i correspond à l'identifiant de la ressource consultée et d_i correspond à la durée de l'événement i. L'univers des événements n'est pas infini dans notre contexte : il correspond typiquement à l'ensemble des pages web visitables ou des ressources consultables.

Base de données de log-items Une *séquence temporelle* est donc une séquence ordonnée d'événements X :

$$X = (s_1, d_1), \ldots, (s_i, d_i) \ldots, (s_n, d_n)$$

Nous étendons cette définition pour introduire une *base de données de séquences temporelles* comme un multiensemble de séquences temporelles. $DL = \{X_1, X_2, \ldots, X_{|DL|}\}$.

Les définitions classiques de la littérature utilisent non pas des durées, mais des temps. Néanmoins, comme il est facile dans la pratique de transformer les temps en durées, nous avons fait le choix d'utiliser une définition avec des durées qui s'avère plus pratique pour expliquer notre approche.

Bases de données de mots Des *bases de données de séquences temporelles*, nous pouvons simplement extraire un multiensemble de séquences d'événements sans tenir compte de leurs durées respectives. Nous appelons une telle séquence un *mot* tenant compte du fait qu'il est constitué d'une séquence de symboles dans l'alphabet des événements. Nous définissons donc une *base de données de mots* D, de taille $|D|$ composée d'un multiensemble d'événements organisés en séquence, simplement appelés *mots*, $D = \{w_1, w_2, \ldots, w_{|D|}\}$. Pour construire D depuis DL on applique :

$$D = \{s | (s, d) \in X,\ X \in DL\}$$

3.3.1 Automates *k-testables*

Les langages *k-testables* au sens strict (k-TSS) ont été introduits par McNaughton and Papert (1971). Intuitivement, un langage k-TSS est déterminé par un ensemble fini de facteurs de longueur inférieure ou égale à k autorisés dans le langage. A l'inverse des langages réguliers, les langages k-TSS peuvent être identifiés à la limite à partie de texte par des algorithmes (Yokomori and Kobayashi, 1998) : cette propriété a rendu ces modèles très attractifs pour de nombreuses applications (Bex et al., 2006; Coste, 2016; Rogers and Pullum, 2011; Tantini et al., 2010).

Plus formellement, un langage k-TSS est déterminé par un ensemble de mots de longueur $k-1$ autorisés comme préfixes et suffixes dans le langage ainsi que par un ensemble de facteurs de longueur k, et enfin un dernier ensemble composé par tout les mots courts (longueur inférieure à $k-1$) contenus dans le langage.

La définition suivante est adaptée de celle de De La Higuera (2010).

Definition 1. *Posons $k > 0$. Un* k-test vecteur *est un 4-tuple $Z = \langle I, F, T, C \rangle$ où*

- *$I \subseteq \Sigma^{k-1}$ est l'ensemble des* préfixes *autorisés,*
- *$F \subseteq \Sigma^{k-1}$ est l'ensemble des* suffixes *autorisés,*

— $T \subseteq \Sigma^k$ *est l'ensemble des* segments *autorisés,*

— $C \subseteq \Sigma^{<k}$ *est l'ensemble des* mots courts *autorisés satisfaisant* $I \cap F = C \cap \Sigma^{k-1}$.

On appelle $\mathcal{T}_k$ *l'ensemble des* k*-test vecteur.*

À partir d'un k-test vecteur, il est possible de construire un automate déterministe à états fini (Dfa) avec les états suivants :

1. $\forall u \cdot v \in I, q_u$ est un état,
2. $\forall x \cdot u \in T, q_u$ est un état,
3. $\forall u \cdot x \in T, q_u$ est un état.
4. $q_0 = q_\lambda$.
5. Les états finaux sont les q_u avec $u \in F$.

Les transitions sont alors construites de la manière suivante :

1. si $u \cdot x$ est préfixe d'un mot $\in I \cup C$, alors $\delta(q_u, a) = q_{ua}$
2. $\forall x \cdot u \cdot y \in T, \delta(q_{xu}, y) = q_{uy}$

Nous ne formalisons pas plus ici la construction : on trouvera les détails dans (De La Higuera, 2010). Nous noterons **Construire-k-*testable*** l'algorithme prenant en entrée un échantillon D et un entier k et retournant le plus petit automate k-*testable* reconnaissant D.

Illustrons cela à partir d'un cas exemple : la Figure 3.1 détaille le processus de création de l'automate correspondant au k-test vecteur à partir d'un corpus de mots. Tout d'abord, on déduit l'alphabet $\Sigma(D)$, ensuite, on peut déduire le k-test vecteur à partir respectivement des préfixes, des suffixes, des segments et des mots courts du corpus. Finalement, il suffit de dérouler l'algorithme de construction de l'automate. On remarque que chaque état correspond à un facteur de taille maximum $k-1$ symboles d'un mot dans les données. C'est une propriété caractéristique d'un automate k-*testable*. Une fois l'automate obtenu, il est possible d'ajouter les fréquences de chaque transition et par conséquent d'en déduire un automate probabiliste.

Quelques propriétés des automates et langages k-*testables*. En premier lieu, l'algorithme **Construire-k-testable** retourne le plus petit langage k-*testable* contenant D.

Si $D_1 \subseteq D_2$ et $\mathcal{A}_{D_1}$ est l'automate k-*testable* obtenu par l'algorithme **Construire-k-testable** à partir de D_1 ($\mathcal{A}_{D_2}$ pour D_2), $\mathcal{A}_{D_1}$ est un sous-automate de $\mathcal{A}_{D_2}$, et tous les

$$D = \begin{array}{|l} a \\ aa \\ abba \\ abbbba \end{array}$$

On choisit $k = 3$, on déduit donc :

$\Sigma(D) = \{a, b\}$
$I(D) = \{aa, ab\}$
$F(D) = \{aa, ba\}$
$C(D) = \{a, aa\}$
$T(D) = \{abb, bbb, bba\}$

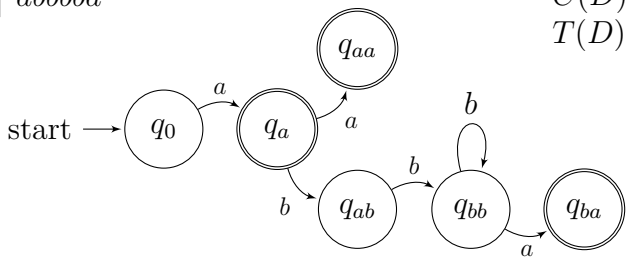


FIGURE 3.1 – Exemple de construction d'un automate k-testable

états et transitions de $\mathcal{A}_{D_1}$ apparaissent également dans $\mathcal{A}_{D_2}$. Ceci est une conséquence de la méthode de construction du k-test vecteur.

D'autre part, chaque état d'un automate *k-testable* correspond à un facteur particulier de longueur $k-1$ ou à un préfixe de longueur inférieure à k. Tout mot contenant ce facteur utilisera nécessairement cet état, et, inversement, si le facteur n'apparaît pas dans le mot, l'état ne sera pas utilisé. Ainsi soit $w = gfd$ avec $g, d \in \Sigma^*$, $f \in \Sigma^{k-1}$, $\delta(q_0, gf) = q_f$.

3.3.2 Automates fréquentiels et probabilistes

Automate fréquentiel déterministe à états fini (FDFA) Les FDFA ont initialement été introduits par (De La Higuera, 2010) comme un modèle comptant le nombre d'occurrences d'un état ou d'une transition lors de l'analyse grammaticale d'une base de données de séquences. Un FDFA peut facilement être converti par normalisation en un modèle voisin : l'automate déterministe probabiliste à états fini (PDFA) en transformant les fréquences en fréquences relatives, puis en probabilités.

Definition 2. *Un* automate fréquentiel déterministe à états fini *(FDFA) est un 6-tuple* $\langle Q, q_0, \Sigma, \delta, C, I \rangle$ *où*

Q **:** $Q = \{q_0 \dots q_n\}$ *est l'ensemble des états de l'automate*

q_0 **:** *l'état initial*

δ **:** $Q \times \Sigma \to Q$ *est la fonction de transition entre états*

C **:** *les fréquences sont données par* $C : (Q \times \Sigma) \cup Q \to \mathbb{N}$ *où* $C(q, x)$ *indique le nombre de fois où la transition* (q, x) *est utilisée et* $C(q) \in \mathbb{N}$ *indique le nombre de*

fois où l'état q est utilisé comme état de sortie.

I **:** *Enfin, $I \in \mathbb{N}$ indique le nombre de fois que l'état q_0 est utilisé comme état d'entrée.*

Chaque transition est identifiée par la paire (q, x). Nous pourrons donc nous référer à la transition (q, x). L'automate respecte toujours la contrainte suivante :

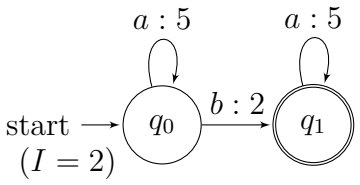
$$\begin{cases} \text{Si } q = q_0, & C(q) + \sum_{x\in\Sigma} C(q,x) = I \\ \text{Sinon } \forall q \in Q, q \neq q_0, & C(q) + \sum_{x\in\Sigma} C(q,x) = \sum_{q'\in Q, x\in\Sigma \text{ t.q. } \delta(q',x)=q} C(q',x) \end{cases} \tag{3.1}$$

Pour simplifier les notations, nous écrirons $C(q,*) = C(q) + \sum_{x\in\Sigma} C(q,x)$.

Nous ajoutons les notations suivantes :

— $C(w,q,x)$ dénote le nombre de fois que la transition (q,x) est utilisée pour parser la chaîne w,
— $C(w,q)$ vaut 1 si $\delta(q,w) = q$, 0 sinon,
— $C(w,q,*) = C(w,q) + \sum_{x\in\Sigma} C(w,q,x)$.

où w est un mot quelconque, q un état et x un symbole de Σ.



$C(q_0) = 0; \quad C(q_1) = 2$

FIGURE 3.2 – Exemple d'illustration d'un FDFA.

Exemple Notons que les fréquences ne se réfèrent pas à un échantillon particulier. Ainsi, le FDFA de la Figure 3.2, peut servir aussi bien pour l'échantillon $\{a^5b, ba^5\}$ ou $\{a^5ba^5, b\}$

ou $\{a^2ba^3, a^3ba^2\}$ etc. Dans chacun de ces cas, les équations suivantes seront vraies :

$$\begin{aligned}
&\delta(q_0, a) = q_0 \text{ fréquence } 5,\ C(q_0, a) = 5\\
&\delta(q_0, b) = q_1 \text{ fréquence } 2,\ C(q_0, b) = 2\\
&\delta(q_1, a) = q_1 \text{ fréquence } 5,\ C(q_1, a) = 5\\
&C(q_0) = 0\\
&C(q_1) = 2\\
&C(q_0, *) = 7\\
&C(q_1, *) = 7\\
&I = 2
\end{aligned}$$

Et il s'ensuit que pour la chaîne a^5b, $C(a^5b, q_0, a) = 5$, $C(a^5b, q_0, b) = 1$, $C(a^5b, q_0)$=0, $C(a^5b, q_1)$=1.

Definition 3. *Un* automate probabiliste déterministe à états fini *(PDFA) est un 5-tuple* $\langle Q, q_0, \Sigma, \delta, \mathbb{P}\rangle$ *où*

Q **:** $Q = \{q_0 \dots q_n\}$ *est l'ensemble des états de l'automate*

q_0 **:** *l'état initial*

δ **:** $Q \times \Sigma \to Q$ *est la fonction de transition entre états*

$\mathbb{P}$ **:** *les probabilités sont données par* $\mathbb{P} : (Q \times \Sigma) \cup Q \to [0; 1]$ *où* $\mathbb{P}(q, x)$ *indique la probabilité de la transition* (q, x) *et* $\mathbb{P}(q) \in [0; 1]$ *la probabilité d'arrêt dans l'état* q.

L'automate respecte toujours la contrainte suivante :

$$\forall q \in Q, \mathbb{P}(q) + \sum_{x \in \Sigma} \mathbb{P}(q, x) = 1 \tag{3.2}$$

En normalisant les fréquences, il est aisé de transformer un FDFA en PDFA. Ainsi, pour un automate $\mathcal{A}$, nous obtenons pour chaque transition une probabilité qui est : $\mathbb{P}^Q_{\mathcal{A}}(q, x) = \frac{C(q,x)}{C(q)+\sum_{y\in\Sigma} C(q,y)} = \frac{C(q,x)}{C(q,*)}$ de générer le symbole $x \in \Sigma^*$ et une probabilité $\mathbb{P}^Q_{\mathcal{A}}(q) = \frac{C(q)}{C(q)+\sum_{y\in\Sigma} C(q,by}} = \frac{C(q)}{C(q,*)}$ de finir sur cet état.

On peut alors en déduire la probabilité $\mathbb{P}_{\mathcal{A}}(w)$ de générer un mot quelconque $w \in \Sigma^*$ avec cet automate $\mathcal{A}$. Pour cela, nous étendons de manière récursive la définition de δ :

$$\begin{aligned}
&\delta(q, \lambda) = q\\
&\delta(q, x_1 \dots x_n) = \delta(\delta(q, x_1), x_2 \dots x_n)
\end{aligned} \tag{3.3}$$

De la même manière, nous étendons les définitions sur les probabilités de transitions :

$$\begin{aligned}&\mathbb{P}(q,\lambda) = \mathbb{P}^Q(q)\\&\mathbb{P}(q,x_1\dots x_n) = \mathbb{P}^Q(q,x_1)\cdot\mathbb{P}(\delta(q,x_1),x_2\dots x_n)\cdot\mathbb{P}(\delta(q,x_1\dots x_n))\end{aligned} \tag{3.4}$$

Finalement, nous obtenons $\mathbb{P}_{\mathcal{A}}(w) = \mathbb{P}_{\mathcal{A}}(q_0, w)$. Pour alléger la notation, nous retirerons l'indice $\mathcal{A}$ en l'absence d'ambiguïté.

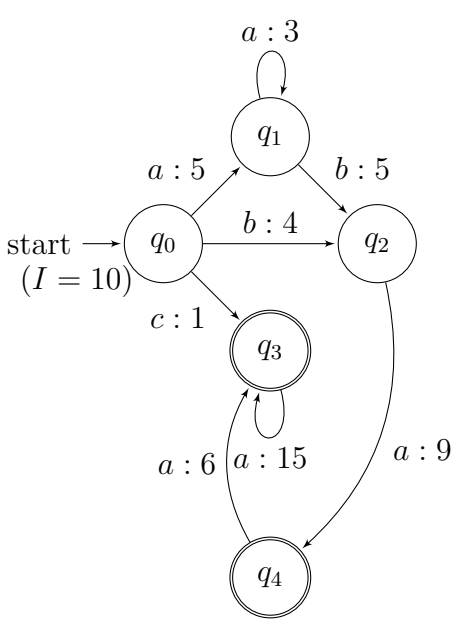
Les différentes constructions décrites ci-dessus donnent lieu aux algorithmes suivants :

— **DFA2FDFA** qui prend en arguments un multi-ensemble de données et un DFA et construit un FDFA de même topologie qui respecte D (Algorithme 2) ;

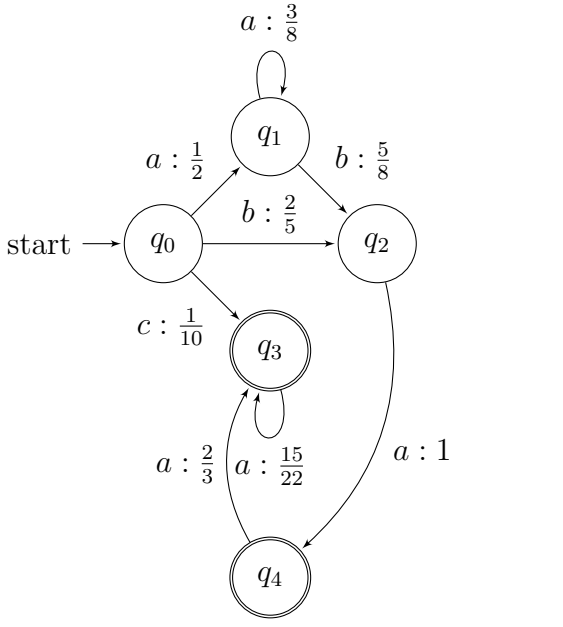
— **FDFA2PDFA** qui prend en argument un FDFA et le transforme en PDFA. L'algorithme est trivial et n'est pas décrit formellement ici.

Comme les automates k-testables sont déterministes, ces constructions s'appliquent à partir d'un *k-testable*.

Et il en découle que **FDFA2PDFA**(**DFA2FDFA**(D, **Construire-k-testable**(D, k))) construit un automate *k-testable* probabiliste à partir d'un échantillon D.



FDFA



PDFA

FIGURE 3.3 – Exemple d'illustration d'un FDFA et du PDFA correspondant.

Données: $\mathcal{A} = \langle Q, q_0, \Sigma, \delta, F\rangle$: DFA, D : Base de données de mots
Résultat: $\langle Q, q_0, \Sigma, \delta, C, I\rangle$: FDFA de même structure que A et respectant D [1]
$\forall q \in Q, \forall x \in \Sigma, C(q, x) \leftarrow 0$
$\forall q \in Q, C(q) \leftarrow 0$
pour chaque $w \in D$ **faire** `// Itération à travers les mots`
  $q \leftarrow q_0$ `//` q_0`: état initial`
  pour chaque $x \in w$ **faire**
    $C(q, x) + +$
    $q \leftarrow \delta(q, x)$
  fin
  $C(q) + +$
fin
$I \leftarrow |D|$
retourner $\langle Q, q_0, \Sigma, \delta, C, I\rangle$

Algorithme 2: Algorithme **DFA2FDFA**, de génération du FDFA à partir de D et d'un automate déterministe

Nous allons illustrer ces notions avec un exemple de FDFA. La Figure 3.3 montre un FDFA et son PDFA correspondant, les états sont $q_0, \ldots, q_4$; l'état initial est q_0, les états q_3 et q_4 sont les deux seuls états finaux, cela est caractérisé sur le schéma par une bordure doublée du cercle représentant l'état. On peut par exemple lire sur la figure que lorsque nous nous trouvons dans l'état q_0 nous pouvons accéder à trois états successeurs distincts q_1, q_2, q_3. L'automate étant déterministe, la transition empruntée découle directement du caractère lu. Ainsi, lorsque nous lisons un a dans l'état q_0 nous atteignons l'état q_1 : on écrira $\delta(q_0, a) = q_1$. A la lecture d'un b nous irons dans l'état q_2 et on pourrait écrire de manière condensée $\delta(q_0, ab) = q_3$. Dans certains cas, il est possible de boucler sur le même état ; ainsi $\delta(q_1, a) = q_1$. On remarque également que certaines transitions ne sont pas définies : par exemple $\delta(q_2, b)$. Dans le cas non probabiliste, cela signifie qu'un mot $w = ubv$ tel que $\delta(q, u) = q_2$ ne sera pas reconnu. Dans le cas probabiliste, la probabilité associée à la transition (q_2, b) est nulle et la probabilité de générer le mot w est 0.

Le nombre indiqué à droite du symbole « : » sur chaque transition indique la fréquence de la transition, il représente la fonction compteur C. La contrainte de l'équation 3.1 est

respectée pour tous les états de l'automate, par exemple pour q_3 :

$$C(q_3, *) = \sum_{q' \in Q, x \in \Sigma \text{ t.q. } \delta(q', x) = q_3} C(q', x)$$
$$C(q_3) + C(q_3, a) = C(q_4, a) + C(q_0, c) + C(q_3, a)$$
$$7 + 15 = 6 + 1 + 15 = 22$$

Ou encore pour le cas particulier de q_0 :

$$C(q_0, *) = I$$
$$C(q_0, a) + C(q_0, b) + C(q_0, c) = I$$
$$5 + 4 + 1 = 10$$

Sur la droite de la figure se trouve l'automate probabiliste dérivé du FDFA : le calcul des probabilités est facile à réaliser en appliquant les formules de $\mathbb{P}^Q_{\mathcal{A}}(q, x)$ et $\mathbb{P}^Q_{\mathcal{A}}(q)$ données plus haut. Par exemple, $\mathbb{P}^Q_{\mathcal{A}}(q_4, a) = \frac{C(q_4,a)}{C(q_4,*)} = \frac{C(q_4,a)}{C(q_4,a)+C(q_4)} = \frac{6}{6+3} = \frac{6}{9} = \frac{2}{3}$.

3.3.3 Les automates k-*testables* probabilistes

L'algorithme **DFA2FDFA** (Algorithme 2) permet de construire le plus petit FDFA *k-testable* consistant (au sens de la Définition 1) avec une base de données de mots D. Ce FDFA peut ensuite être transformé en PDFA avec l'algorithme **FDFA2PDFA**. Le DFA peut lui-même être obtenu par l'algorithme **Construire-k-testable**.

Si on décompose, la construction est la suivante :

1. $\mathcal{A}_1 \leftarrow$**Construire-k-testable**(D, k)
2. $\mathcal{A}_2 \leftarrow$ **DFA2FDFA**$(\mathcal{A}_1, D)$
3. $\mathcal{A}_3 \leftarrow$ **FDFA2PDFA**$(\mathcal{A}_2)$

Le paramètre k détermine la longueur maximale de la facteur représentant l'état. Intuitivement, dans une configuration générative, il contrôlera la mémoire du modèle : lorsque k est fixé à la longueur maximale des mots dans l'ensemble de données, les FDFA sur-apprennent la distribution originale. À l'inverse, avoir un k fixé à 1 correspond à ne considérer qu'un seul état sans mémoire. Avoir un k fixé à 2 correspond à un unigramme (pour $k = 2$ symbole et état sont confondus). En pratique, k reste souvent petit en raison de la croissance exponentielle du nombre d'états avec k. Pour cette raison, la valeur

classique de k varie entre 1 et 5. Dans notre cas, nous privilégierons des valeurs plus faibles car notre objectif est principalement de préserver la dépendance à court terme.

Une implémentation efficace de cette suite algorithme peut créer le FDFA dynamiquement à partir de D pendant la construction de l'automate *k-testable*.

L'ensemble a une complexité temporelle de $O(||D||)$ où $||D||$ est le nombre total de symboles dans les chaînes. C'est la complexité de l'algorithme de construction des automates *k-testables*.

Une mise en œuvre efficace peut utiliser une matrice éparse ou une structure similaire.

3.3.4 Les automates déterministes temporisés

L'objectif de ce travail étant d'obtenir un modèle génératif de séquences temporelles, nous souhaitons que le modèle soit capable de capturer à la fois les facteurs récurrents des sessions, mais aussi leurs durées (durée totale de la session, durée de chaque événement dans ces sessions). Avec cet objectif en ligne de mire, commençons par ajouter à un automate déterministe un modèle de durée sur les transitions.

Dans la littérature des automates temporisés, le temps est mesuré à travers les transitions à l'aide d'horloges qui sont principalement utilisés pour déterminer l'acceptation ou le rejet d'un mot temporisé (*timed word* en anglais) (Alur and Dill, 1994). Notre modèle étant génératif, nous n'utilisons pas d'horloge. À la place, chaque transition se voit attribuer une distribution de durée déterminée par une loi normale. Nous définissons cela dans le contexte des automates déterministe, fréquentiels et probabilistes :

Definition 4. *Un automate déterministe temporisé (*DFA-t*) est un six-tuple* $\langle Q, q_0, \Sigma, \delta, F, \Delta_t \rangle$ *où* $\langle Q, q_0, \Sigma, \delta, F \rangle$ *est un automate déterministe et* Δ_t *est une fonction retournant, pour chaque transition, une Gaussienne positive sur la durée des transitions* $\Delta_t : Q \times \Sigma \to \mathbb{R}^+ \times \mathbb{R}^+$.

Definition 5. *Un automate déterministe fréquentiel temporisé (*FDFA-t*) est un six-tuple* $\langle Q, q_0, \Sigma, \delta, C, \Delta_t \rangle$ *où* $\langle Q, q_0, \Sigma, \delta, C, I \rangle$ *est un* FDFA *et* Δ_t *est une fonction retournant, pour chaque transition, une Gaussienne positive sur la durée des transitions* $\Delta_t : Q \times \Sigma \to \mathbb{R}^+ \times \mathbb{R}^+$.

Definition 6. *Un automate déterministe probabiliste temporisé (*PDFA-t*) est un six-tuple* $\langle Q, q_0, \Sigma, \delta, P, \Delta_t \rangle$ *où* $\langle Q, q_0, \Sigma, \delta, \mathbb{P} \rangle$ *est un* PDFA *et* Δ_t *est une fonction retournant, pour chaque transition, une Gaussienne positive sur la durée des transitions* $\Delta_t : Q \times \Sigma \to \mathbb{R}^+ \times \mathbb{R}^+$.

Ces définitions peuvent être étendues pour d'autres distributions servant à modéliser la durée des transitions.

Étant donné un automate déterministe probabiliste temporisé A, une *séquence temporelle* est générée en exécutant l'algorithme **GénèreSéquence** (Algorithme 3).

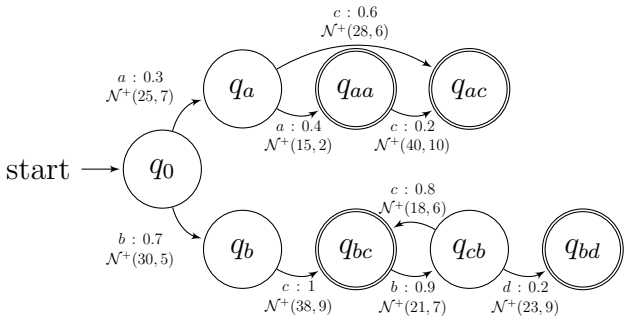
Si la structure de l'automate est celle d'un k-testable les définitions ci-dessus s'appliquent sans difficulté.

Données: $\mathcal{A} = \langle Q, q_0, \Sigma, \delta, \mathbb{P}, \Delta_t \rangle$: PDFA-t
Résultat: Une séquence temporelle X
$q \leftarrow q_0$; $X \leftarrow []$
$x \sim next(q)$// x `est tiré conformément à la distribution sur l'état` q
tant que $x \neq \$$ **faire** // `$ est le symbole associé à la fin de mot`
  $\mu, \sigma \leftarrow \Delta_t(q, x)$
  $d \sim \mathcal{N}^+(\mu, \sigma)$// d `est tiré depuis une distribution Gaussienne positive`
  $X \leftarrow X \cdot (x, d)$
  $q \leftarrow \delta(q, x)$
  $x \sim next(q)$// x `est tiré conformément à la distribution sur l'état` q
fin
retourner X

Algorithme 3: Algorithme **GénèreSéquence**, de génération d'une séquence depuis un PDFA-t

Des statistiques peuvent également être construites pour chaque automate temporisé, et nous pouvons calculer la durée espérée d'une séquence, par exemple.

Pour chaque transition (q, a), $\Delta_t(q, a)$ est une paire (μ, σ), utilisé comme paramètres d'une Gaussienne positive $\mathcal{N}^+(\mu_{(q,x)}, \sigma^2_{(q,x)})$, avec $\mu_{(q,x)}$ le temps moyen de consultation observé sur la transition et $\sigma_{(q,x)}$ son écart-type.



$$\mathbb{P}(q_{aa}) = 0.8; \quad \mathbb{P}(q_{ac}) = 1; \quad \mathbb{P}(q_{bc}) = 0.1; \quad \mathbb{P}(q_{bd}) = 1$$

FIGURE 3.4 – Exemple d'illustration d'un PDFA-t

La figure 3.4 représente un automate probabiliste temporisé 3-*testable* simplifié. L'alphabet est $\{a, b, c, d\}$. Chaque transition a un modèle de durée modélisé par une distribution Gaussienne positive ; par exemple, l'événement c, lorsqu'il a lieu dans l'état q_a, nous conduit à l'état q_{ac}, la durée associée suit la distribution Gaussienne $\mathcal{N}^+(\mu_{(28,6)})$, la durée moyenne de cette transition est donc de 28 secondes avec une variance de six secondes. Sa probabilité associée est de 0.4.

Une séquence générée à partir de ce PDFA-t pourrait être $[(b, 22), (c, 27), (b, 15), (d, 21)]$. Une telle séquence correspondrait à un utilisateur parcourant la page b pendant 22 secondes, puis la page c pendant 27 secondes, puis à nouveau la page b pendant 15, et enfin la page d pendant 21 secondes et s'arrêtant. Les automates *k-testables* sont déterministes : la construction ci-dessus s'applique donc bien.

La construction d'un automate temporisé à partir d'un automate (DFA, FDFA ou PDFA) et un échantillon D est assez simple : elle consiste à construire, pour chaque transition (q, w), l'ensemble des durées correspondant aux utilisations par les mots de D de cette transition et d'en déduire la moyenne et l'écart type empiriques. Les algorithmes correspondants sont alors notés **DFA2Temp** pour les DFA, **FDFA2Temp** pour les FDFA, et **PDFA2Temp** pour les PDFA.

Rappelons qu'étant donné une séquence temporelle X, nous notons $\mathbf{Mot}(X)$ le mot obtenu en ignorant les durées. Ainsi, $\mathbf{Mot}([(b, 22), (c, 27), (b, 15), (d, 21)])$=*bcbd*.

3.3.5 Les constructeurs

Nous avons donc introduit les algorithmes et fonctions suivantes :

— **Construire-K-testable** prend un entier positif k et un échantillon et construit l'automate *k-testable* correspondant ; par définition, cet automate est un DFA ;

— **DFA2FDFA** permet d'utiliser un DFA pour parser un multi-ensemble D et obtenir ainsi les fréquences d'utilisation des états finaux et de toutes les transitions ;

— **FDFA2PDFA** transforme un FDFA en PDFA ;

— **DFA2Temp** va calculer les moyennes et les écarts types des utilisations des transitions et associer à chacune un écart type ; dans ce cas le support est un DFA ;

— **FDFA2Temp** va calculer les moyennes et les écarts types des utilisations des transitions et associer à chacune un écart type ; dans ce cas le support est un FDFA ;

— **PDFA2Temp** va calculer les moyennes et les écarts types des utilisations des transitions et associer à chacune un écart type ; dans ce cas le support est un

PDFA ;

— **Mot** extrait d'une séquence d'évènements-durées le mot obtenu en concaténant les événements.

En combinant ces différentes fonctions, il est possible d'obtenir un automate temporisé à partir des données. Bien entendu, une implémentation efficace combinera les aspects de ces algorithmes en une seule passe.

3.4 Vers l'anonymisation des données

Le PDFA temporisé obtenu à partir des constructions précédentes peut être utilisé pour générer de nouvelles chaînes : la distribution de ces chaînes est proche de la distribution originale (à partir de l'ensemble de données). L'intérêt de l'approche par automates *k-testables* est de découper les données en morceaux (facteurs) et de se priver ainsi des dépendances à long terme : cet aspect est parfois considéré comme un obstacle à l'utilisation de cette méthode, mais peut être vu comme un avantage ici.

En effet, dans notre contexte, les données sont des journaux de navigation des utilisateurs et le risque majeur en particulier pour les données séquentielles est le risque de ré-identification. Un scénario typique d'attaque considère deux bases de données : une base non anonymisée avec de nombreuses informations -qui peuvent-être publiques- sur les individus et la base supposée protégée car pseudonymisée. Un attaquant cherchera à croiser les deux fichiers. L'objectif de l'anonymisation est de protéger les individus contre ce type d'attaques [2] Pour la suite, nous ferons toujours l'hypothèse qu'un utilisateur correspond à une unique séquence qui peut être arbitrairement longue. Cette hypothèse est cohérente avec la méthode de récupération des traces d'utilisateurs dans le projet X5GON qui se fait au niveau du cookie. Plusieurs chercheurs ont déjà démontré la possibilité de ré-identifier les utilisateurs sur ce type de jeu de données et mis en exergue les risques qui en découlent (Narayanan and Shmatikov, 2006; Rocher et al., 2019). Ces algorithmes de ré-identification tirent profit des séquences longues comme quasi-identificateurs des utilisateurs. Plus une séquence est longue, plus il est probable qu'elle n'ait été empruntée que par un faible nombre d'utilisateurs et plus l'utilisateur correspondant sera vulnérable.

De ce point de vue, le fait de ne pas conserver les longues dépendances constitue un avantage.

2. Cette explication est simplifiée dans un but didactique : le lecteur pourra se référer à (Pedersen, 2005) pour une explication plus approfondie.

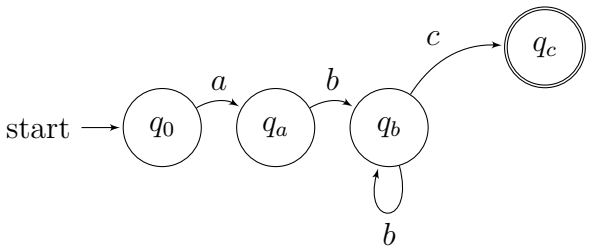
Pourtant, de nombreux problèmes de confidentialité ne sont toujours pas résolus. Ils sont étudiés dans le domaine de la *publication de données confidentielles* (Fung et al., 2010) et notre objectif est de suivre leurs directives.

Commençons par présenter les problèmes d'une manière plus informelle. La question posée est la suivante : « Certaines informations du modèle appris peuvent-elles dépendre trop fortement d'un enregistrement unique ? ». Une réponse positive à cette question serait problématique pour deux raisons : (i) cela montrerait une faible robustesse du modèle, et (ii) trop d'informations sur cet utilisateur unique pourraient être vues, ce qui crée un risque de ré-identification. On parle de risque de ré-identification lorsqu'un attaquant est capable, sur la base des données anonymisées et de potentielles connaissances extérieures, de casser l'anonymisation et ainsi de pouvoir déterminer la présence d'un individu dans les données originelles.

Nous voulons donc détecter cette situation et nettoyer la base de données en conséquence.

Dans le contexte des PDFA temporisés, cela signifie que la fréquence d'une transition est lourdement modifié en la présence ou en l'absence d'un individu dans le jeu de données de départ, c.-à-d. par une séquence unique du jeu de données. Il faut examiner les différentes transitions du FDFA correspondant et vérifier si les fréquences observées avec et sans la chaîne ne sont pas trop éloignés. Pour cela, nous fixerons un seuil multiplicatif (voir plus loin la Définition 7).

Illustrons cela sur un exemple et supposons un jeu de données contenant quatre séquences différentes : $\{abc, abd, abbc\}$. De plus, nous supposons que ce jeu de données provient des traces d'apprentissage de plusieurs utilisateurs, parmi lesquels Bob dont la séquence représentative est *abbc*. Bob a la particularité d'être le seul à avoir emprunté une séquence contenant deux *b*. Si l'on construit l'automate 2-testable correspondant, on obtient :



Dans cet automate 2-*testable*, on remarque que la transition (q_b, b) n'est due qu'aux traces de Bob. Si l'on choisit d'utiliser cet automate pour générer un jeu de données,

nous obtiendrons avec une probabilité non-nulle des séquences contenant le facteur *bb*. Un attaquant ayant comme connaissance préalable la spécificité de Bob pourra alors utiliser cette séquence comme quasi-identificateur de Bob et ainsi déduire que Bob fait effectivement partie du jeu de données. Bien sûr, a grand renfort d'inférence, il devient alors possible de déduire des informations supplémentaires sur le comportement de Bob. Ici par exemple, on peut savoir avec certitude que Bob a également visité *a* ou *c*. Bien sûr, dans des cas réels, des cas aussi triviaux sont peu fréquents ; néanmoins, comme l'ont démontré Narayanan and Shmatikov (2006) ou encore Rocher et al. (2019), il reste possible d'extraire des facteurs quasi-identificateurs et d'en tirer profit pour casser la confidentialité.

Dans ce type de cas, une manière intuitive de résoudre le problème est de retirer la séquence quasi-identificatrices de Bob. En pratique, on voudra fixer un seuil à partir duquel une séquence est trop prédominante pour une transition donnée et possède alors un risque trop important de ré-identification.

De plus, si le nombre d'utilisateurs est grand, les comportements trop spécifiques peuvent être considérés comme des artefacts. Ainsi, supprimer ces séquences améliore la robustesse du modèle.

3.4.1 Les principes de la confidentialité différentielle

La confidentialité différentielle (CD) est un cadre mathématique permettant d'assurer une garantie de confidentialité prouvable introduite par Dwork et al. (Dwork et al., 2006). Ce cadre a été développé à l'origine pour faire face aux attaques par ré-identification (section 2.1.4). Celles-ci sont des attaques particulièrement problématiques dans le cas des données séquentielles (section 2.1.4).

L'avantage principal de la CD est d'être complètement indépendante du domaine ou des connaissances préalables de l'attaquant. La CD se fonde sur l'observation suivante : un utilisateur pour lequel aucune information n'a été publiée n'encourt aucun risque de fuite de données lors d'une publication statistique de données dans le domaine public. Cette affirmation, bien que triviale, permet d'observer le corollaire suivant : si l'absence ou la présence des données d'un utilisateur n'influe pas sur la distribution statistique des données mises dans le domaine public, tout se passe pour lui comme si ces données n'étaient pas dans la base de données. Un tel individu n'encourt alors aucun risque de fuite de ses données confidentielles. Ainsi, avec la confidentialité différentielle, l'objectif est de donner à chaque individu une confidentialité proche de celle qui résulterait de la

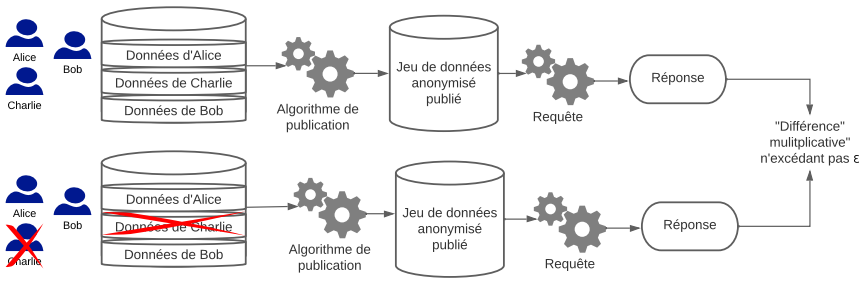


FIGURE 3.5 – La confidentialité différentielle dans le cadre de la publication de données.

suppression de ses données.

Bien sûr, il ne sera pas toujours impossible que chaque individu ait un apport complètement nul pour cela, on introduit un seuil multiplicatif qui évalue la variation acceptable du résultat en l'absence ou en la présence des données de l'utilisateur.

Dans notre cas, nous cherchons à publier les données sans contraintes sur leur utilisation. On parle de publication de données non interactive : on souhaite alors que les données publiées soient robustes en termes de confidentialité à tout post-traitement et donc tout type de requête. La Figure 3.5 résume la mise en place de la publication de données non interactives dans le cadre de la CD. L'approche que nous allons présenté ne garantit pas la CD, néanmoins, elle s'en inspire en la transposant dans une version relaxée au niveau de l'automate génératif (Voir Section 3.4.3).

3.4.2 La démarche et les propriétés désirées

Par les constructions précédentes, nous avons construit un FDFA $\mathcal{A}$ qui peut sans difficulté être enrichi en PDFA, puis en PDFA $-$ t $\mathcal{B}$ à partir de la base de séquences DL.

Avec l'algorithme **GénèreSéquence** (Algorithme 3), il est possible d'utiliser $\mathcal{B}$ pour construire une nouvelle base DL'. Mais cette base DL' pourrait contenir des séquences quasi-identificatrices, au sens défini dans la section précédentes. Une alternative serait d'élaguer l'ensemble DL'. Mais il est bien plus efficace et intéressant d'élaguer DL et d'obtenir la garantie que DL' n'aura pas (avec une forte probabilité) de séquences quasi-identificatrices.

Les durées peuvent également être quasi-identificatrices : une personne anormalement rapide (ou lente) pourra par exemple voir ce motif reconnu. De nombreux algorithmes ont été envisagés pour anonymiser des données continues telles que les durées : le mécanisme

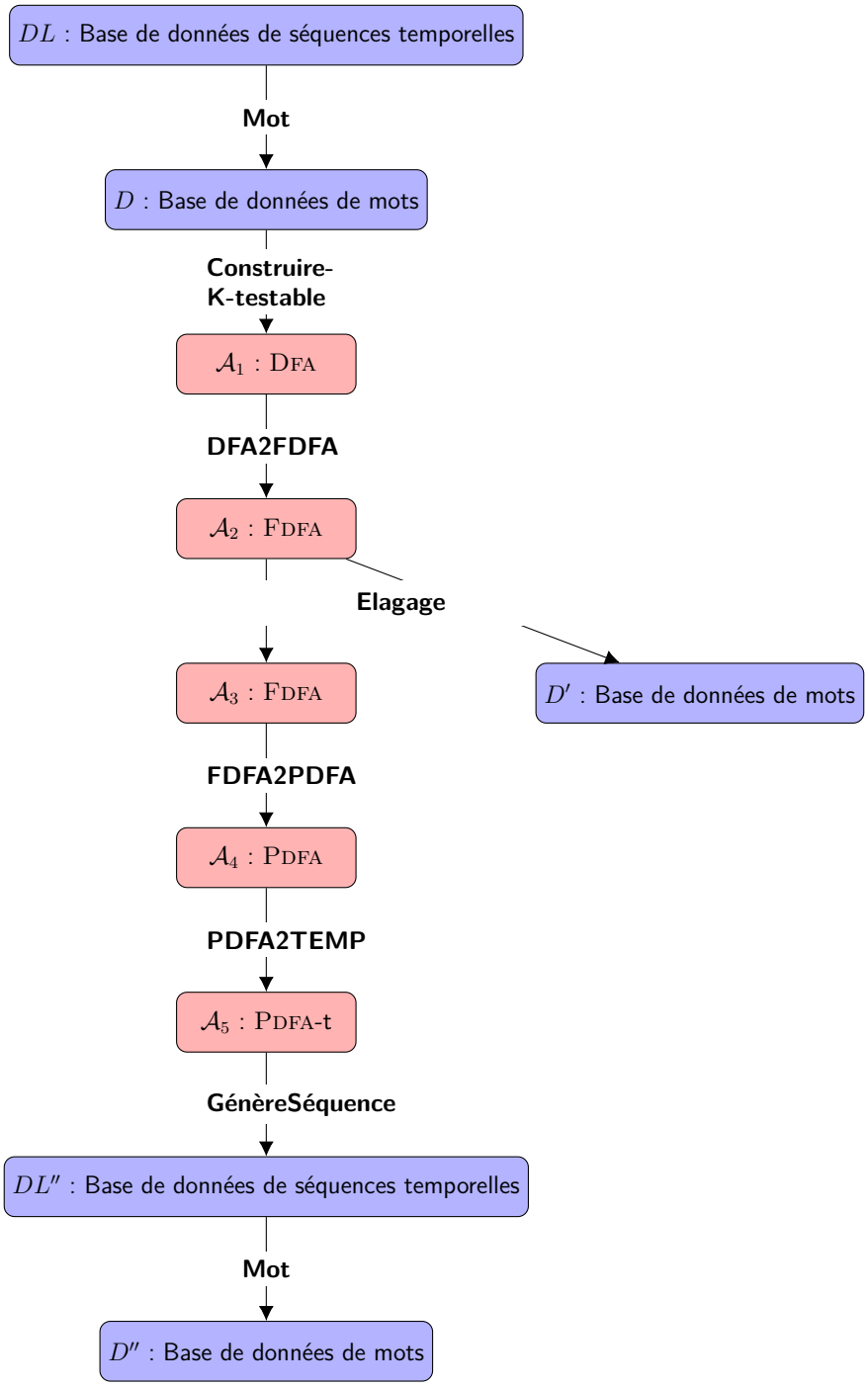


FIGURE 3.6 – Démarche globale de traitement des données.

Laplacien introduit par (Dwork and Roth, 2013) permet par exemple de garantir la CD sur ce type de donnée. Bien sûr, garantir indépendamment la confidentialité sur les séquences et sur les durées ne garantit pas la confidentialité sur l'ensemble. Néanmoins, dans notre contribution, nous choisirons de nous concentrer sur l'anonymisation des séquences. Ainsi, nous travaillerons sur D plutôt que sur DL.

Le point de départ est donc un échantillon D. Cet échantillon contient des données quasi-identificatrices. Mais nettoyer D trop généreusement peut rendre l'ensemble restant inutile. Il s'agira donc de réussir un compromis entre

— ne pas trop s'éloigner de D pour avoir des données utiles, c'est-à-dire qui ressemblent aux données initiales,
— éliminer les séquences quasi-identificatrices,
— faire en sorte qu'aucune donnée, isolément, n'ait un poids trop important.

Nous noterons donc

— D l'échantillon de départ ;
— $\mathcal{A}_1$ le Dfa obtenu à partir de D (pour la valeur de k choisie) ;
— $\mathcal{A}_2$ le Fdfa obtenu à partir de $\mathcal{A}_1$ et D ;
— D' l'échantillon obtenu en élaguant l'échantillon de départ des données sensibles, et $\mathcal{A}_3$ le Fdfa correspondant :
— $\mathcal{A}_4$ et $\mathcal{A}_5$ respectivement le Pdfa et le Pdfa-t obtenus à partir de $\mathcal{A}_3$ et D' ;
— DL'' une base de séquences temporelles générée à partir de $\mathcal{A}_5$
— D'' la base de données de mot correspondant à DL''.

Nous résumons la démarche dans le diagramme représenté en Figure 3.6.

Pour simplifier les notations dans la suite, nous noterons :

DFA(D, k) : Un Dfa $\mathcal{A}$ tel que $\mathcal{A} =$ **Construire-k-testable**(D, k). Par exemple, dans la Figure 3.6, on a $\mathcal{A}_1 =$ **DFA**(D, k).

FDFA(D, k) : Un Fdfa $\mathcal{A}$ tel que $\mathcal{A} =$ **DFA2FDFA**$(D,$ **DFA**$(D, k))$. Par exemple, dans la Figure 3.6, on a $\mathcal{A}_2 =$ **FDFA**(D, k).

PDFA(D, k) : Un Pdfa $\mathcal{A}$ tel que $\mathcal{A} =$ **FDFA2PDFA**(**FDFA**$(D, k))$. Par exemple, dans la Figure 3.6, on a $\mathcal{A}_4 =$ **PDFA**(D', k).

PDFA-t(D, k) : Un Pdfa-t $\mathcal{A}$ tel que $\mathcal{A} =$ **PDFA2temp**(**PDFA**$(D, k))$.

conf-FDFA(D, k, ϵ) : Un Fdfa $\mathcal{A}$ tel que $(\mathcal{A}, D') =$ **Elagage**(**FDFA**$(D, k), D, \epsilon)$ avec D' la base de mot issue de l'élagage de D.

conf-PDFA(D, k, ϵ) : Un Pdfa $\mathcal{A}$ tel que $\mathcal{A} =$ **FDFA2PDFA**(**conf-FDFA**$(D, k, \epsilon))$.

conf-PDFA-t(D, k, ϵ) : Un PDFA-t $\mathcal{A}$ tel que $\mathcal{A} = \textbf{PDFA2temp}(\textbf{conf-PDFA}(D, k, \epsilon))$.

Voici maintenant les 4 propriétés que nous espérons obtenir. Les notations sont celles de la Figure 3.6, page 194 :

Propriété 1 D' et D'' sont proches.

C'est une propriété qui requiert une définition de distance. Puis un résultat en probabilité et à la limite. Autrement dit, une convergence de D'' vers D' avec une taille croissante de D''.

Propriété 2 Si D et D' sont proches, D et D'' sont proches.

Il s'agit là d'un corollaire assez évident lié à l'inégalité triangulaire, qui sera justifié si la proximité est mesurée par une métrique.

Propriété 3 Dans D' il n'y a pas de séquences quasi-identificatrices.

Ce résultat sera démontré page 203 : l'algorithme d'élagage que nous proposons élimine les séquences responsables, à elles seules, d'une transition ou d'un état final.

Propriété 4 Pour toute chaîne w de D', $\mathcal{A} = \textbf{PDFA}(D', k)$ et $\mathcal{B} = \textbf{PDFA}(D' - \{w\}, k)$:

$$\forall u \in \Sigma^*; \mathbb{P}_{\mathcal{A}}(u) \approx \mathbb{P}_{\mathcal{B}}(u)$$

.

Autrement dit, les chaînes restantes, après élagage, sont en quelque sorte robustes : en enlever une n'a qu'une faible influence sur la distribution. Cette propriété sera délicate à démontrer. Mais dans des cas particuliers raisonnables (un même facteur de longueur $k-1$ n'apparaît pas deux fois dans la chaîne) la propriété sera vérifiée.

3.4.3 L'ϵ-*sensibilité*

Afin de traiter le problème de confidentialité mentionné précédemment, nous introduisons la notion d'ϵ-*sensibilité*. Intuitivement, une transition sera ϵ-*sensible* pour une chaîne donnée si le fait d'enlever cette chaîne à l'échantillon entraîne une différence de plus de ϵ de la fréquence relative associée à cette transition.

Definition 7. *(ϵ-sensibilité) Soit $\mathcal{A}$ un* FDFA *et w un mot. Nous disons que $w = uxv$ est ϵ-sensible* pour la transition (q, x) *dans $\mathcal{A}$ avec $q = \delta(q_0, u)$ si :*

$$C(w, q, x) = C(q, x) \tag{3.5}$$

ou

$$\frac{C(q, x) - C(w, q, x)}{C(q, *) - C(w, q, *)} \geq (1 + \epsilon)\frac{C(q, x)}{C(q, *)} \tag{3.6}$$

ou

$$\frac{C(q, x) - C(w, q, x)}{C(q, *) - C(w, q, *)} \leq (1 - \epsilon)\frac{C(q, x)}{C(q, *)} \tag{3.7}$$

Nous disons également que w est ϵ-sensible pour l'état $q = \delta(q_0, w)$ *dans $\mathcal{A}$ si :*

$$C(q) = 1 \tag{3.8}$$

ou

$$\frac{C(q) - 1}{C(q, *) - C(w, q, *)} \geq (1 + \epsilon)\frac{C(q)}{C(q, *)} \tag{3.9}$$

ou

$$\frac{C(q) - 1}{C(q, *) - C(w, q, *)} \leq (1 - \epsilon)\frac{C(q)}{C(q, *)} \tag{3.10}$$

Par commodité, on écrira également qu'une transition (resp. état) est *ϵ-sensible* pour une chaîne donnée quand cette chaîne est *ϵ-sensible* pour la dite transition (resp. état). On dit d'un FDFA $\mathcal{A}$ sans aucune transition *ϵ-sensible* et sans aucun état *ϵ-sensible* pour toute chaîne dans D qu'il est ϵ-confidentiel sur D.

Nous écrirons également qu'un mot w est *ϵ-fragile* s'il existe une transition (q, x) pour laquelle ce mot est *ϵ-sensible* ou si ce mot est *ϵ-sensible* pour l'état $\delta(q_0, w)$.

On remarque que cette définition est applicable pour n'importe quel type de FDFA, indépendamment de la manière donc il est construit. Néanmoins, dans notre contexte, l'ensemble des mots étudiés provient d'une base de données de mot D. Et le FDFA $\mathcal{A}$ est construit à partir de D, $\mathcal{A} = \mathbf{FDFA}(D, k)$.

De plus, les valeurs considérées pour ϵ seront positives et inférieures à 0.5, une valeur usuelle pourra être $\epsilon = 0.01$.

Commençons par expliciter les termes utilisés dans la définition :

— $\frac{C(q,x)}{C(q,*)}$ est le nombre de fois que la transition (q,x) est utilisée sur le nombre total de fois que l'état q est atteint. Cela correspond à la probabilité $\mathbb{P}^Q_{\mathcal{A}}(q,x)$ de la transition dans le PDFA correspondant à $\mathcal{A}$.

— $\frac{C(q,x)-C(w,q,x)}{C(q,*)-C(w,q,*)}$ est le nombre de fois que la transition (q,x) est utilisée sur le nombre total de fois que l'état q est atteint, mais calculé sur un échantillon dans lequel on a enlevé la chaîne w. Elle correspond à la probabilité $\mathbb{P}^Q_{\mathcal{B}}(q,x)$ de la transition dans le PDFA $\mathcal{B} = \mathbf{PDFA}(D - \{w\}, k)$.

— $\frac{C(q)}{C(q,*)}$ est le nombre de fois que l'état q est utilisé comme état final sur le nombre total de fois que l'état q est atteint. Cela correspond à la probabilité $\mathbb{P}^Q_{\mathcal{A}}(q)$ d'arrêt dans l'état q pour le PDFA correspondant à $\mathcal{A}$.

— $\frac{C(q)-1}{C(q,*)-C(w,q,*)}$ est le nombre de fois où l'état q est utilisé comme état final sur le nombre total de fois où l'état q est atteint, mais calculé sur un échantillon dans lequel on a enlevé la chaîne w. Cela correspond à la probabilité $\mathbb{P}^Q_{\mathcal{B}}(q)$ d'arrêt dans l'état q pour le PDFA $\mathcal{B} = \mathbf{PDFA}(D - \{w\}, k)$.

Explicitons la définition 7 : il y a six conditions individuellement indépendantes et suffisantes pour satisfaire ϵ-*sensibilité*, représentées par les 6 équations de la définition, analysons les en détail :

— La première condition (équation 3.5) est remplie quand la chaîne w est la seule chaîne à employer la transition (q,x). Dans le cas dans lequel $\mathcal{A}$ est un k-*testable* construit à partir de D cela signifie soit (i) que w contient un facteur de longueur k unique dans D, soit (ii) que w contient un préfixe de longueur maximum $k-1$ unique dans D.

— Les deux conditions suivantes (équations 3.6 et 3.7) expriment une borne multiplicative entre les probabilités $\mathbb{P}^Q_{\mathcal{A}}(q,x)$ et $\mathbb{P}^Q_{B}(q,x)$ avec $\mathcal{B} = \mathbf{PDFA}(D - \{w\}, k)$. Ces conditions contraignent la transition à ne pas voir sa probabilité être grandement influencée (en positif ou en négatif) par la présence ou l'absence du mot dans l'échantillon. En ce sens, cette condition peut être reformulée avec les probabilités :

$$\mathbb{P}^Q_{B}(q,x) \geq (1+\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q,x) \text{ ou } \mathbb{P}^Q_{B}(q,x) \leq (1-\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q,x) \tag{3.11}$$

— La quatrième condition (équation 3.8) est remplie quand la chaîne w est la seule chaîne à s'arrêter dans l'état q. Dans le cas dans lequel $\mathcal{A}$ est un k-*testable* construit

à partir de D cela signifie que w contient un suffixe de longueur maximum $k-1$ unique dans D.

— Les deux conditions suivantes (équations 3.9 et 3.10) expriment une borne multiplicative entre les probabilités $\mathbb{P}^Q_{\mathcal{A}}(q)$ et $\mathbb{P}^Q_{\mathcal{B}}(q)$ avec $\mathcal{B} = \mathbf{PDFA}(D - \{w\}, k)$. Ces conditions contraignent l'état à ne pas voir sa probabilité d'arrêt être grandement influencée (en positif ou en négatif) par la présence ou l'absence du mot dans l'échantillon. En ce sens, cette condition peut être reformulée avec les probabilités :

$$\mathbb{P}^Q_{\mathcal{B}}(q) \geq (1+\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q) \text{ ou } \mathbb{P}^Q_{\mathcal{B}}(q) \leq (1-\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q) \tag{3.12}$$

Quelques résultats découlent de cette définition et nous serviront pour la suite :

1. Seuls les états finaux peuvent être ϵ-*sensibles* ;
2. Si une chaîne n'utilise pas une transition ou n'atteint pas un état, celle-ci (ou celui-ci) n'est pas ϵ-*sensible* sur cette transition ou cet état ;
3. Plus ϵ est petit, plus la tolérance est faible et donc plus le nombre de chaînes ϵ-*sensible* est susceptible d'être grand. À l'inverse, plus ϵ est grand, plus la tolérance est élevée et donc plus le nombre de chaînes ϵ-*sensible* est susceptible d'être petit. Toutes choses égales par ailleurs, soit $\epsilon_2 < \epsilon_1$ si une chaîne est ϵ_1-*sensible* elle est aussi ϵ_2-*sensible*.
4. Si une chaîne ne passe par l'état q qu'une seule fois, et emploie une transition (q, x) employée par au moins une autre chaîne, alors la chaîne n'est pas ϵ-*sensible* sur cette transition pour toutes valeurs de $\epsilon > 0$.
5. Si une chaîne ne passe par l'état q qu'une seule fois, et s'y arrête, et qu'une autre chaîne s'arrête également dans l'état q, alors la chaîne n'est pas ϵ-*sensible* sur cette transition pour toutes valeurs de $\epsilon > 0$.
6. Si un mot $w \in D$ contient un facteur de longueur k unique dans D alors le mot est ϵ-*sensible* pour la transition correspondante à ce facteur. Il est donc ϵ-*fragile*.
7. Si un mot $w \in D$ contient un préfixe de longueur au plus $k-1$ unique dans D alors le mot est ϵ-*sensible* pour la transition correspondante à ce préfixe. Il est donc ϵ-*fragile*.
8. Si un mot $w \in D$ contient un suffixe de longueur $k-1$ unique dans D alors le mot est ϵ-*sensible* pour l'état correspondant à ce suffixe. Il est donc ϵ-*fragile*.

9. Si un FDFA $\mathcal{A}$ est ϵ-confidentiel sur D alors, pour tout mot w de D, pour toutes transitions (q, x) de $\mathcal{A}$ et pour tout état q de $\mathcal{A}$, considérant avec $\mathcal{B} = \mathbf{PDFA}(D - \{w\}, k)$ on a :

$$(1-\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q,x) < \mathbb{P}^Q_{\mathcal{B}}(q,x) < (1+\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q,x) \tag{3.13}$$

$$(1-\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q) < \mathbb{P}^Q_{\mathcal{B}}(q) < (1+\epsilon)\mathbb{P}^Q_{\mathcal{A}}(q) \tag{3.14}$$

.

Exemple

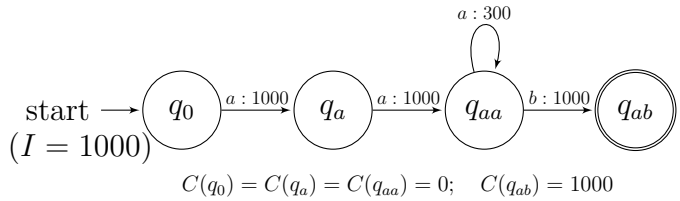


FIGURE 3.7 – Exemple d'illustration de l'ϵ-sensibilité.

Considérons-le FDFA de la Figure 3.7. Pour toute chaîne w de la forme aa^ib, on remarque que les transitions (q_0, a), (q_a, aa) et (q_{aa}, q_{ab}) ne sont empruntées qu'une unique fois. Donc aucune chaîne de la forme aa^ib n'est ϵ-*sensible* sur (q_0, a), (q_a, a) et (q_{aa}, b). De plus, aucune chaîne de la forme aa^ib n'est ϵ-*sensible* sur (q_{ab}) car $C(q_{ab}) > 1$.

Intéressons-nous maintenant au cas des transitions : $\{(q_{aa}, a),\ (q_{aa}, b)\}$. On a :

$$\begin{aligned}
&C(q_{aa}, a) = 300\\
&C(q_{aa}, b) = 1000\\
&C(q_{aa}, *) = 1300\\
&\mathbb{P}_{\mathcal{A}}(q_{aa}, a) = \frac{C(q_{aa}, a)}{C(q_{aa}, *)} = \frac{300}{1300}\\
&\mathbb{P}_{\mathcal{A}}(q_{aa}, b) = \frac{C(q_{aa}, b)}{C(q_{aa}, *)} = \frac{1000}{1300}
\end{aligned}$$

Et donc pour toute chaîne w de la forme aa^ib avec $i < 300$, on remarque que l'ϵ-*sensibilité* ne dépendra que du nombre d'occurrences du facteur aa (on en comptera toujours $C(w, q_{aa}, a)+1$) et de la valeur de ϵ. Considérant B le PDFA voisin de A construit

à partir $D - w$.

$$\text{Pour la transition } (q_{aa}, a)\ \mathbb{P}_{\mathcal{B}}(q_{aa}, a) = \frac{300 - C(w, q_{aa}, a)}{1300 - (C(w, q_{aa}, a) + 1)}$$
$$\text{Pour la transition } (q_{aa}, b)\ \mathbb{P}_{\mathcal{B}}(q_{aa}, b) = \frac{1000 - 1}{1300 - (C(w, q_{aa}, a) + 1)}$$

Posons $\epsilon = 0.1$. On observe alors que la transition (q_{aa}, a) est ϵ-*sensible* pour $C(w, q_{aa}, a) = 150$ mais pas pour $C(w, q_{aa}, a) = 25$. Comme le montre la Figure 3.8, la probabilité $\mathbb{P}_{\mathcal{A}}(q_{aa}, a)$ croît avec $C(w, q_{aa}, a)$. En fait, $C(w, q_{aa}, a) = 118$ est la valeur limite de fréquence imputable à w à partir de laquelle la transition devient ϵ-*sensible*.

Dans le cas de la transition (q_{aa}, b), comme le montre la Figure 3.8, la probabilité $\mathbb{P}_{\mathcal{A}}(q_{aa}, b)$ décroît avec $C(w, q_{aa}, a)$. En fait, $C(w, q_{aa}, a) = 38$ est la valeur limite de fréquence imputable à w à partir de laquelle la transition devient ϵ-*sensible*.

Au global, pour la chaîne w, la transition limitante est (q_{aa}, a). Comme on le voit sur la Figure 3.8 la transition (q_{aa}, a), w est ϵ-*fragile* si $C(w, q_{aa}, a) > 38$.

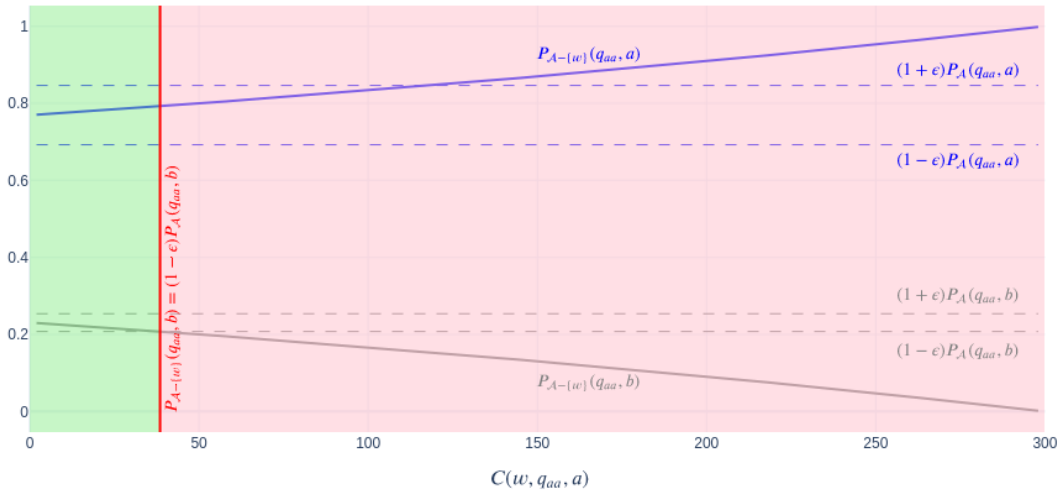


FIGURE 3.8 – Évolution de l'ϵ-*sensibilité* en fonction des valeurs de $C(w, q_{aa}, a)$ pour les transitions (q_{aa}, a) et (q_{aa}, b). ϵ est fixé à 0.1.

3.4.4 Élaguer l'automate et la base de données

Notre objectif est de construire, à partir d'une base de données D, un automate probabiliste générateur de chaînes. Dans un premier temps, l'ensemble D est nettoyé de ses chaînes les plus fragiles : il s'agit de celles qui ont une influence trop particulière sur le générateur : elles sont seules (ou presque) à posséder un facteur particulier. La recherche de chaînes fragiles se fait chaîne par chaîne en parcourant la chaîne et, sur chaque transition, en vérifiant si les conditions de la définition 7 sont remplies.

À partir de cet échantillon D' un FDFA est construit, puis transformé en PDFA. Ce PDFA peut maintenant générer autant de chaînes que nécessaire.

Données: $\langle Q, q_0, \Sigma, \delta, C, I\rangle$: FDFA, D : échantillon de mots, ϵ : $]0, 1[$
Résultat: FDFA ϵ-confidentiel $\langle Q, q_0, \Sigma, \delta, C, I\rangle$, D'
$convergence \leftarrow False$
$D' \leftarrow D$
tant que *non convergence* **faire**
  $convergence \leftarrow True$
  $to\_remove \leftarrow \{\}$
  pour chaque $w \in D'$ **faire** `// Recherche des chaînes fragiles`
    si w *ϵ-fragile* **alors**
      $convergence \leftarrow False$
      $to\_remove \leftarrow to\_remove \cup \{w\}$
    fin
  fin
  pour chaque $w \in to\_remove$ **faire** `// Mise à jour des fréquences`
    $q = q_0$
    pour chaque $x \in w$ **faire** `// Parcours de l'automate`
      $C(q, x) \leftarrow C(q, x) - 1$
      $C(q, *) \leftarrow C(q, *) - 1$
      $q \leftarrow \delta(q, x)$
    fin
    $C(q) \leftarrow C(q) - 1$
    $C(q, *) \leftarrow C(q, *) - 1$
  fin
  $D' \leftarrow D' - to\_remove$`// Mise a jour de l'échantillon`
fin
retourner ***DFA2FDFA****(****Construire_k-testable****(*D', k*),* D'*)*

Algorithme 4: Algorithme **Elagage** pour satisfaire l'ϵ-confidentialité

L'algorithme **Elagage** (Algorithme 4) élague un FDFA afin de garantir qu'aucune des transitions restantes dans le FDFA de sortie ne soit *ϵ-sensible* pour une chaîne quelconque de D' où D' et obtenu en enlevant les chaînes *ϵ-sensible*s de D.

L'algorithme s'arrête lorsqu'il ne reste plus aucune chaîne *ϵ-sensible* (pour une tran-

sition quelconque). Quand cette condition est respectée, l'algorithme retourne le FDFA ainsi obtenu sur le nouvel ensemble de données élagué D'.

L'algorithme garantit évidemment l'ϵ-confidentialité puisqu'il s'arrête quand toutes les chaînes ϵ-*fragile*s ont été élaguées.

Complexité Dans l'algorithme 4 la variable *convergence* ne peut prendre la valeur *False* qu'au plus D fois. Dans chaque itération de la boucle externe, chaque lettre de chaque mot est examinée. Et en fonction du résultat du test d'ϵ-*sensibilité*, des mises à jour ont lieu. Cela correspond à une complexité interne de $O(\|D\|)$. Donc au total, la complexité de l'algorithme **Elagage**(Algorithme 4) est $O(|D| \cdot \|D\|)$. Le pire cas pourrait-être atteint quand les chaînes sont éliminées une à la fois.

À titre d'illustration, sur un jeu de données avec $\approx 10^6$ log-items, une implémentation non optimisée en python obtient un temps d'exécution inférieur à quelques minutes sans parallélisation du processus.

Terminaison. L'algorithme **Elagage** (4) termine : à chaque itération de la boucle principale une chaîne au moins sera identifiée comme ϵ-fragile et sera ensuite éliminée. Or l'ensemble D' est fini et n'est modifié que par suppression de ses éléments.

Convergence. De plus, la procédure est indépendante de l'ordre dans lequel les éléments de D' sont visités. La première boucle `pour` permet de localiser les chaînes à éliminer mais ne met pas à jour les compteurs : l'ordre est donc sans importance. Cette propriété est importante et assure que le résultat de l'élagage est unique. Une autre conséquence est celle d'assurer une équité entre les utilisateurs (représentés par les chaînes).

Élimination des chaînes ϵ-fragiles. Lorsque l'algorithme se termine les chaînes ϵ-fragiles auront été éliminées, ce qui assure la Propriété 3.4.2, 196.

3.4.5 Cas d'illustration sur une base de données simple

Analysons maintenant un exemple d'exécution, afin de mieux comprendre le comportement de l'algorithme. Attention ! Les situations réelles prévues pour ce travail concernent des échantillons de milliers de logs. Cet exemple n'est donc pas représentatif des cas réels.

La figure 3.9 détaille un ensemble de données d'entrées pour l'algorithme **Elagage** (Algorithme 4) avec une base de données de mot D, un FDFA 3-*testables* que nous noterons

$\mathcal{A}$ et un seuil de sensibilité $\epsilon = 0.01$. L'ensemble de données D est représenté comme un multi-ensemble où, pour chaque chaîne, le nombre d'occurrences est comptabilisé. Le tableau résume quant à lui les fréquences de l'automate pour chacune des chaînes.

Conformément au reste du document, nous notons $\mathbb{P}_{\mathcal{B}}(q, x)$ la probabilité de la transition (q, x) dans le FDFA construit à partir de $D - \{w\}$. Ainsi $\mathbb{P}_{\mathcal{B}}(q, x) = \frac{C(q,x)-C(w,q,x)}{C(q,*)-C(w,q,*)}$ au niveau des transitions et $\mathbb{P}_{\mathcal{B}}(q, x) = \frac{C(q,x)-1}{C(q,*)-C(w,q,*)}$ au niveau des états atteints pas w.

Chaque itération de la boucle principale de l'algorithme se déroule en deux étapes : une étape de recherche des chaînes ϵ-*sensible* et une étape de suppression des chaînes sensibles et de mise à jour de l'automate.

Certaines chaînes du jeu de données sont identiques ; par exemple, il y a 500 occurrences de la chaîne *bc*. En appliquant strictement l'algorithme, il convient de considérer chacune des chaînes indépendamment. Néanmoins, la convergence nous garantit que le calcul sera le même pour les chaînes identiques. Comme pourrait le faire une implémentation efficace, nous ne considérerons qu'une occurrence de chaque chaîne lors de la phase de recherche. Si cette occurrence est ϵ-*sensible*, nous conclurons que toutes les chaînes identiques le sont et respectivement si cette occurrence n'est pas ϵ-*sensible*. De plus, comme seul les états finaux et les transitions atteintes peuvent être ϵ-*sensible* pour une chaîne donnée, nous ne considérerons que ces états et ces transitions dans l'étape de recherche. Le tableau de la 3.10 résume l'étape de recherche de la première itération.

La première chaîne considérée est la chaîne *a*. On remarque que la chaîne est ϵ-*sensible* pour la transition (q_0, a) car c'est la seule chaîne qui atteint la transition $(C(q_0, a) = C(a, q_0, a) = 1)$. On mémorise donc que la chaîne est fragile, et nous la supprimerons et mettrons à jour l'automate lors de la deuxième étape.

La deuxième chaîne considérée est *bcba*. En l'état actuel, elle n'est pas ϵ-*sensible* sur (q_0, b) car plusieurs chaînes empruntent la transition et que :

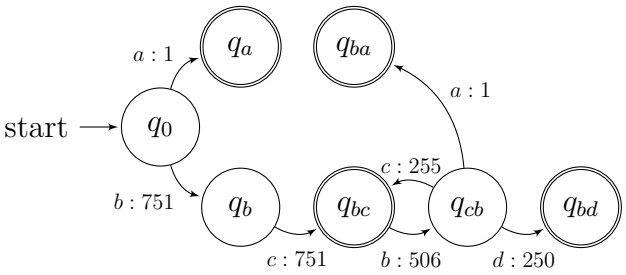
$$(1-\epsilon)\mathbb{P}_{\mathcal{A}}(q_0, b) < \mathbb{P}_{\mathcal{B}}(q_0, b) < (1+\epsilon)\mathbb{P}_{\mathcal{A}}(q_0, b)$$
$$0.99\frac{750}{751} < \frac{749}{750} < 1.01\frac{749}{750}$$

On observe également que *bcba* n'est pas sur (q_b, c) et (q_c, b) en revanche, il est ϵ-*sensible* sur $(q_c b, a)$ en étant le seul mot utilisant la transition. De la même manière que pour *a*, on mémorise que *bcba* est fragile.

En suivant le même processus, on observe que *bc*, $(bc)^2bd$ et $(bc)^7bd$ ne sont pas ϵ-

Données d'entrées :

$$D = \left| \begin{array}{l} a:1 \\ bcba:1 \\ bc:500 \\ (bc)^2bd:249 \\ (bc)^7bd:1 \end{array} \right.$$



Fréquences avant traitement :

Transition	$C(q,x)$	$C(q,*)$	$C(a,q,x)$	$C(bcba,q,x)$	$C(bc,q,x)$	$C((bc)^2bd,q,x)$	$C((bc)^7bd,q,x)$
(q_0,a)	1	752	1	0	0	0	0
(q_0,b)	751	752	0	1	1	1	1
(q_b,c)	751	751	0	1	1	1	1
(q_{bc},b)	506	1006	0	1	1	2	7
(q_{cb},a)	1	506	0	1	0	0	0
(q_{cb},c)	255	506	0	0	0	1	6
(q_{cb},d)	250	506	0	0	0	1	1
Etat	$C(q)$	$C(q,*)$	$C(a,q,x)$	$C(bcba,q,x)$	$C(bc,q,x)$	$C((bc)^2bd,q,x)$	$C((bc)^7bd,q,x)$
q_a	1	752	1	0	0	0	0
q_{ba}	1	1	0	1	0	0	0
q_{bc}	500	1006	0	0	1	0	0
q_{bd}	250	250	0	0	0	1	1

FIGURE 3.9 – Exemple d'illustration de l'algorithme d'élagage, avec $\epsilon = 0.01$

Mot w	Transition/Etat	$\mathbb{P}_{\mathcal{A}}(q,x)$	$\mathbb{P}_{\mathcal{B}}(q,x)$	ϵ-sensibilité
a	(q_0, a)	1/752	0/751	OUI
a est ajoutée aux chaînes à supprimer				
$bcba$	(q_0, b)	750/751	749/750	NON
$bcba$	(q_b, c)	750/750	749/749	NON
$bcba$	(q_{bc}, b)	506/1006	505/1005	NON
$bcba$	(q_{cb}, a)	1/506	0/505	OUI
$bcba$ est ajoutée aux chaînes à supprimer				
bc	(q_0, b)	750/751	749/750	NON
bc	(q_b, c)	750/750	749/749	NON
bc	q_{bc}	500/1006	449/1005	NON
$(bc)^2bd$	(q_0, b)	750/751	749/750	NON
$(bc)^2bd$	(q_b, c)	750/751	749/750	NON
$(bc)^2bd$	(q_{bc}, b)	506/1006	504/1004	NON
$(bc)^2bd$	(q_{cb}, c)	255/506	254/503	NON
$(bc)^2bd$	(q_{cb}, d)	250/506	249/503	NON
$(bc)^2bd$	q_{bd}	250/250	249/249	NON
$(bc)^7bd$	(q_0, b)	750/751	749/750	NON
$(bc)^7bd$	(q_b, c)	750/751	749/750	NON
$(bc)^7bd$	(q_{bc}, b)	506/1006	499/999	NON
$(bc)^7bd$	(q_{cb}, c)	255/506	249/499	NON
$(bc)^7bd$	(q_{cb}, d)	250/506	249/499	NON
$(bc)^7bd$	q_{bd}	250/250	249/249	NON

FIGURE 3.10 – Étape de recherche des chaînes *ϵ-sensible* (1ère itération) de l'algorithme **Elagage** (Algorithme 4) pour l'exemple de la Figure 3.9 avec $\epsilon = 0.01$.

fragiles. Une fois toutes les chaînes du jeu de données testées, il convient de supprimer les chaînes fragiles et de mettre à jour l'automate : c'est la deuxième étape de la boucle principale de l'algorithme.

Transition	$C(q,x)$	$C(q,*)$	$C(bc,q,x)$	$C((bc)^2bd,q,x)$	$C((bc)^7bd,q,x)$
(q_0,b)	~~751~~750	~~752~~750	1	1	1
(q_b,c)	~~751~~750	~~751~~750	1	1	1
(q_{bc},b)	~~751~~750	~~1006~~1005	0	2	7
(q_{cb},b)	255	~~506~~505	0	1	6
(q_{cd},d)	250	~~506~~505	0	1	1
Etat	$C(q)$	$C(q,*)$	$C(bc,q)$	$C((bc)^2bd,q)$	$C((bc)^7bd,q)$
q_{bc}	500	~~1006~~1005	1	0	0
q_{bd}	250	250	0	1	1

$$D' = \left| \begin{array}{l} \sout{a : 1} \\ \sout{bcba : 1} \\ bc : 500 \\ (bc)^2bd : 249 \\ (bc)^7bd : 1 \end{array} \right.$$

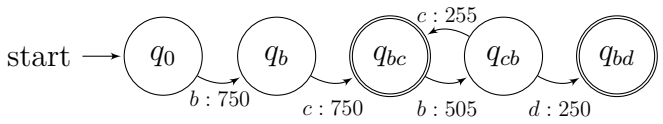


FIGURE 3.11 – Etape de mise à jour des fréquences (1ère itération) de l'algorithme **Elagage** (Algorithme 4) sur l'exemple de la Figure 3.9 avec $\epsilon = 0.01$.

Le tableau (Figure 3.11) résume les mises à jour des fréquences ; les nouvelles valeurs pour les fréquences altérées sont indiquées en vert dans le tableau, les anciennes valeurs sont indiquées en noir barré.

1. Pour tous les états atteint par les chaînes fragiles, les valeurs $C(q,*)$ sont modifiées, on doit retirer la contribution des chaînes supprimées. Ainsi, en notant C_{prec} la fonction de fréquence avant suppression de la chaîne, on a par exemple : $C(q_0,*) = C_{prec}(q_0,*) - C(a,q_0,a) - C(bcba,q_0,b)$
2. De la même manière, sur toutes les transitions atteintes, on retire la contribution des chaînes supprimées. Ainsi en notant C_{prec} la fonction de fréquence avant suppression de la chaîne, on a par exemple : $C(q_0,a) = C_{prec}(q_0,a) - C(a,q_0,a)$.
3. Enfin, pour l'état final, on retire la contribution de la chaîne supprimée à la fréquence finale. Par exemple $C(q_a) = C_{prec}(q_a) - 1$. Ici, on remarque, que $C(q_{ba})$ devient nul, et il n'apparaît donc pas dans le tableau. Au niveau de l'automate, cela signifie que l'état est supprimé. Il en va de même pour les transitions (q_{cb},a), q_0,a et l'état q_a.

La partie basse de la Figure 3.11 résume l'état de l'automate après la suppression des chaînes a et $bcba$. Comme une chaîne a été supprimée, la convergence n'a pas été atteinte, il faut donc parcourir à nouveau le jeu de données à la recherche de potentielles chaînes ϵ-*fragiles*.

La figure 3.12 résume ce deuxième parcours du jeu de données. Les chaînes bc, $(bc)^2bd$ et $(bc)^7bd$ sont testées dans cet ordre. De manière intéressante, on remarque que pour les états n'ayant pas subi de mise à jour de leurs fréquences, les équations restent les mêmes. Dans notre cas par exemple q_{bd}.

Une implémentation efficace pourrait tirer profit de cet invariant en ne testant les chaînes que sur les états ayant subi des mises à jour de leurs fréquences. Comme illustré par le tableau, les chaînes bc et $(bc)^2bd$ ne sont pas fragiles. Pour la chaîne $(bc)^7bd$ la transition (q_{cb}, d) est ϵ-*sensible* : ici la sensibilité vient du fait que la suppression de la chaîne altérerait trop fortement la probabilité de la transition, dans notre cas :

$$\mathbb{P}_{\mathcal{B}}(q_{cb}, d) = (1 + \epsilon)\mathbb{P}_{\mathcal{A}}(q_{cb}, d)$$

Ainsi la probabilité de la transition (q_{cb}, d) est trop fortement impactée par la chaîne $(bc)^7bd$. Les fréquences sont alors mises à jour en conséquence (voir tableau de la Figure 3.12). Comme la convergence n'est pas atteinte, une nouvelle passe est nécessaire.

La Figure 3.13 résume le troisième parcours sur ce jeu de données. Ici aucune chaîne n'est plus ϵ-*fragile* et la convergence est donc atteinte. On obtient alors le jeu de données et l'automate final. Comme aucune suppression n'a eu lieu, l'automate et les fréquences sont inchangés par rapport à l'itération précédente.

Etape 2 (pas de convergence) :

Mot w	Transition/Etat	$\mathbb{P}_{\mathcal{A}}(q,x)$	$\mathbb{P}_{\mathcal{B}}(q,x)$	ϵ-sensibilité
bc	(q_0, b)	750/750	749/749	NON
bc	(q_b, c)	750/750	749/749	NON
bc	q_{bc}	500/1005	499/1004	NON
$(bc)^2bd$	(q_0, b)	750/750	749/749	NON
$(bc)^2bd$	(q_b, c)	750/750	749/749	NON
$(bc)^2bd$	(q_{bc}, b)	505/1005	503/1003	NON
$(bc)^2bd$	(q_{cb}, c)	255/505	254/503	NON
$(bc)^2bd$	(q_{cb}, d)	250/505	249/503	NON
$(bc)^2bd$	q_{bd}	249/249	249/249	NON
$(bc)^7bd$	(q_0, b)	750/750	749/749	NON
$(bc)^7bd$	(q_b, c)	750/750	749/749	NON
$(bc)^7bd$	(q_{bc}, b)	505/1005	498/998	NON
$(bc)^7bd$	(q_{cb}, c)	255/505	249/498	NON
$(bc)^7bd$	(q_{cb}, d)	250/505	249/498	OUI

$(bc)^7bd$ **est ajoutée aux chaînes à supprimer**

Mise à jour des fréquences et de l'automate :

Transition	$C(q,x)$	$C(q,*)$	$C(bc,q,x)$	$C((bc)^2bd,q,x)$
(q_0, b)	~~750~~749	~~750~~749	1	1
(q_b, c)	~~750~~749	~~750~~749	1	1
(q_{bc}, b)	~~505~~498	~~1005~~998	0	2
(q_{cb}, b)	~~255~~249	~~506~~498	0	1
(q_{cd}, d)	~~250~~249	~~506~~498	0	1
Etat	$C(q)$	$C(q,*)$	$C(bc,q)$	$C((bc)^2bd,q)$
q_{bc}	500	~~1005~~998	1	0
q_{bd}	~~250~~249	~~250~~249	0	1

Automate et jeu de données à la fin de l'étape 2 :

$$D' = \left| \begin{array}{l} \sout{a : 1} \\ \sout{bcba : 1} \\ bc : 500 \\ (bc)^2bd : 249 \\ \sout{(bc)^7bd : 1} \end{array} \right.$$

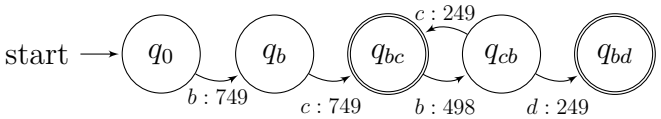


FIGURE 3.12 – Etape 2 de l'algorithme **Elagage** (Algorithme 4) sur l'exemple de la Figure 3.9 avec $\epsilon = 0.01$.

Mot w	Transition/Etat	$\mathbb{P}_{\mathcal{A}}(q,x)$	$\mathbb{P}_{\mathcal{B}}(q,x)$	ϵ-sensibilité
bc	(q_0, b)	749/749	748/748	NON
bc	(q_b, c)	749/749	748/748	NON
bc	q_{bc}	500/998	499/997	NON
$(bc)^2bd$	(q_0, b)	749/749	748/748	NON
$(bc)^2bd$	(q_b, c)	749/749	748/748	NON
$(bc)^2bd$	(q_{bc}, b)	498/998	496/996	NON
$(bc)^2bd$	(q_{cb}, c)	249/498	248/496	NON
$(bc)^2bd$	(q_{cb}, d)	249/498	248/496	NON
$(bc)^2bd$	q_{bd}	249/249	248/248	NON

Aucune chaîne restante n'est ϵ-fragile la convergence est atteinte.

FIGURE 3.13 – Étape de recherche des chaînes *ϵ-sensible* (3ème itération) de l'algorithme **Elagage** (Algorithme 4) pour l'exemple de la Figure 3.9 avec $\epsilon = 0.01$.

3.4.6 Quelques premiers résultats

Nous analysons l'algorithme **Elagage**(Algorithme 4) vis-à-vis des 4 propriétés identifiées page 196 :

— (Propriété 1) Les deux multi-ensembles D' et D" sont proches : même si cette propriété reste à démontrer formellement, elle est liée au fonctionnement des algorithmes d'apprentissage d'automates probabilistes. L'automate $\mathcal{A}2$ ayant généré D" est appris à partir de D. Le fait que l'algorithme d'apprentissage fonctionne a pour conséquence qu'un échantillon produit par le générateur appris est proche de l'échantillon d'apprentissage. Bien entendu, la notion de distance sous-jacente ici mériterait également être explicitée : on peut utiliser les distances usuelles entre distributions, comme définies et étudiées dans (De La Higuera, 2010).

— (Propriété 2) Par inégalité triangulaire, une distance d entre D et D" est majorée par $d(D, D') + d(D', D")$ si cette distance est une métrique.

— (Propriété 3) Une séquence quasi identificatrice est par définitions *ϵ-sensible*. L'algorithme **Élagage** élimine toutes les chaînes *ϵ-sensibles*. La propriété 3 tient donc. Mais il convient de nuancer : une séquence qui correspondrait à un utilisateur qui passerait beaucoup plus de temps (ou beaucoup moins) que les autres pourrait également être révélatrice : les durées associées aux séances pourraient également quasi-identifier. C'est un aspect qui n'est pas pris en compte dans cette étude.

— (Propriété 4) La propriété 4 s'exprime ainsi : Soit $\mathcal{A}(D)$ et $\mathcal{A}(D - \{w\})$ les deux PDFAs construits respectivement à partir de D et $D - \{w\}$. Soit u une chaîne

quelconque. On cherche à mesurer $|\mathbb{P}_{\mathcal{A}(D)}(u) - \mathbb{P}_{\mathcal{A}(D-\{w\})}(u)|$. Comme w n'est pas ϵ-*sensible* il en ressort que $|\mathbb{P}_{\mathcal{A}(D)}(u) - \mathbb{P}_{\mathcal{A}(D-\{w\}}(u)| < (1-\epsilon)^{|u|}$. En première analyse cette inégalité s'avère insuffisante pour établir une borne sur la distance entre des automates probabilistes proches, car si cette distance est sommée sur toutes les chaînes, la somme peut, en théorie diverger.
En seconde analyse, le poids des chaînes u pour lesquelles la différence est grande ne semble pas pouvoir être important. Mais à ce stade, il est impossible d'établir formellement un résultat théorique. Nous devrons nous contenter de proposer le problème suivant :

Problème 1. *Soit $\epsilon > 0$. Soit $\mathcal{A} = \langle Q, \Sigma, q_0, \delta, \mathbb{P}_{\mathcal{A}} \rangle$ et $\mathcal{B} = \langle Q, \Sigma, q_0, \delta, \mathbb{P}_{\mathcal{B}} \rangle$ deux PDFAs tels que :*

- *Support($\mathcal{A}$)=Support($\mathcal{B}$), c'est-à-dire $\mathcal{A}$ et $\mathcal{B}$ sont identiques aux probabilités près ;*
- $\forall q \in Q, \forall a \in \Sigma$, $(1-\epsilon)\mathbb{P}_{\mathcal{A}}(q,a) \leq \mathbb{P}_{\mathcal{B}}(q,a) \leq (1+\epsilon)\mathbb{P}_{\mathcal{A}}(q,a)$

Question *Peut-on majorer $d(\mathcal{A}, \mathcal{B})$? Dans ce qui précède d peut être la distance de Manhattan, Euclidienne,...*

3.5 Cas d'étude : X5gon

Table name : **user_activities**

Champs	Description
id	Identifiant unique de l'enregistrement
timestamp	Date de l'enregistrement
referrer_url	Url de provenance
cookie_id	Identifiant du cookie d'accès
url_id	Url accédée

TABLE 3.1 – Architecture de la table "transitions" du *Connect service* dans la base de données du projet X5GON.

Dans le projet X5GON, le parcours d'apprentissage de l'utilisateur est enregistré lorsque celui-ci navigue sur un site Web partenaire par l'intermédiaire d'un service nommé *connect service*)(Voir Section 2.3.2). Ce service permet de suivre un utilisateur a travers les différents sites partenaires et de conserver son parcours d'apprentissage.

Nous pouvons identifier la date d'accès aux ressources (*ua.timestamp*), l'url de provenance (*ua.referrer_url*), l'url actuellement accédée (*ua.url_id*) et l'utilisateur actuel à travers les champs cookie_id (*ua.cookie_id*), le tout est résumé dans le tableau 3.1.

Un exemple de trace d'apprentissage pourrait être le suivant : un utilisateur se connecte une première fois depuis son téléphone (le cookie 123 lui est attribué) sur le site de l'université de Valence et il y consulte trois ressources avec des durées respectives de consultation de 10min, 2min et 5min. Les entrées suivantes sont enregistrées dans la base de données (1, 04/11/2021 11H00, UPV, 123, url1), (2, 04/11/2021 11H10, UPV, 123, url2), (3, 04/11/2021 11H12, UPV, 123, url3). Le lendemain, le même utilisateur se connecte à `vidéolectures.NET` pour y consulter une série de quatre vidéos de la même manière. Les entrées suivantes sont enregistrées dans la base de données : (4, 05/11/2021 10H00, vidéolectures.NET, 123, url4), (5, 05/11/2021 10H08, vidéolectures.NET, 123, url5), (6, 05/11/2021 10H13, vidéolectures.NET, 123, url6), (7, 05/11/2021 10H22, vidéolectures.NET, 123, url7).

Si l'utilisateur se reconnecte avec son téléphone, son cookie restera le même ; en revanche s'il se connecte sur un autre appareil, il obtiendra un nouveau cookie et sera considéré comme un utilisateur indépendant.

Même si un utilisateur peut correspondre à plusieurs cookies (notamment en cas d'accès multi-appareils), nous créons une relation bi-univoque entre les utilisateurs individuels et les cookies. Néanmoins, pour notre application, nous serons intéressés par les sessions d'apprentissage à moyen/court terme, c'est pourquoi cette hypothèse nous semble raisonnable.

Pré-traitement Les données stockées dans la table *user_activities* sont regroupées par *cookie_id*, et pour chaque groupe, les données sont triées en augmentant le *timestamp* (du plus ancien au plus récent). De cette manière, pour chaque *cookie_id* u, on obtient une séquence L_u des urls accédées par l'utilisateur u, et une autre séquence T_u des dates d'accès par l'utilisateur u pour chaque url.

Nous évaluons notre méthode sur un ensemble de données d'utilisateurs du projet X5GON. Pour nos expériences, nous essayons différentes valeurs de k et pour chaque k, différentes valeurs de ϵ. Pour chaque couple de valeurs, un PTK-TA est déduit et nous générons un nouveau jeu de données artificielles de 100 000 séquences.

Statistiques et spécificités du jeu de données Le jeu de données X5GON est composé de 100 236 sessions totalisant plus de 2 millions d'accès sur 13 784 ressources. Les ressources sont accédées 154 fois en moyenne dans notre jeu de données, néanmoins la distribution suit une loi de puissance, ainsi 25% des ressources ont moins de 2 accès et 50% moins de 71 accès. Le tableau 3.3 détaille les statistiques en termes d'accès par sessions dans sa première ligne, avec une longueur de session variant entre 4 et ≈2400, mais plus de 90% des sessions contenant moins de ≈50 accès. De même, le tableau 3.4 détaille les statistiques en termes de durée des sessions. Dans la première ligne, la durée moyenne des sessions est d'environ $\approx 1h20min$, mais nous observons une grande variance : 50% des sessions durent moins de $30m$ et environ 90% plus de 4 heures.

Ces statistiques montrent qu'il va être difficile d'anonymiser l'ensemble des données tout en conservant une bonne utilité. Tout d'abord, la grande taille de l'alphabet (nombre de ressources) est peu fréquente dans les ensembles de données de la littérature. Par exemple, MSNBC et STM, qui sont des références classiques (Chen et al., 2012), ont respectivement une taille d'univers de 17 et 342. La taille de leur univers est très faible comparée à la taille de notre ensemble de données (13787) alors que le nombre de séquences est 10 fois plus grand (environ 1 million). En termes d'automates, cela engendre un alphabet de taille ≈ 13000, et par conséquent un très grand nombre d'états : en effet en théorie jusqu'à $\binom{|\Sigma|}{k}$ états. En pratique, tous les états ne sont heureusement pas atteints, mais néanmoins c'est un nombre très important d'états comparativement aux jeux de données usuels comme on peut le voir dans la tableau 3.2. Nous faisons donc face a une situation de forte disparité (voir Section 1.3.4.2), avec des états de l'automate visités très peu de fois. Dans ces conditions nous nous attendons logiquement à un découpage important des données : en effet, moins un état est atteint plus les transitions on des chances d'être ϵ-*sensibles*.

Il en va de même pour la longueur des chaînes de caractères, qui est en moyenne beaucoup plus importante dans notre ensemble de données (environ 21) que dans les autres ensembles de données (environ 5 ou 6) (Chen et al., 2012). Deuxièmement, de nombreuses ressources sont accédées très peu souvent, ce qui les rend sensibles en termes de confidentialité. En effet, l'objectif même des attaques de liaison est de trouver des événements suffisamment rares pour être utilisés comme quasi-identifiants pour les utilisateurs. Pour cette raison, conformément à la définition 7, les chaînes contenant ces ressources sont dites ϵ-*sensibles* et sont retirées de l'ensemble de données pendant l'algorithme d'élagage. De plus, des chaînes plus longues signifient également une plus grande variété possible de

motif fréquents à détecter.

3.6 Résultats expérimentaux

Pour les expériences, nous fixons différentes valeurs de k (de 1 à 5) et pour chaque valeur de k, différentes valeurs de ϵ ([0.01, 0.02, 0.05, 0.1, 0.2, 0.25]). Pour chaque couple de valeurs, un nouvel ensemble de données artificielles de 100 000 échantillons est généré. Ainsi, en appelant DL la base de log-items correspondant aux données réelles, pour chaque couple de valeur (k, ϵ), nous générons un automate en appliquant l'algorithme **conf-PDFA-t**(DL, k, ϵ).

Dans les tableaux qui suivent, les lignes portant la mention (RD) décrivent les résultats pour les données résultant de l'élagage de DL pour différentes valeurs de k et ϵ (on a : $\mathcal{B}, DL' =$ **conf-FDFA**(DL, k, ϵ)). Ils ne s'agit donc pas de données générées.

Ainsi une ligne $((RD), k = 1, \epsilon = 0.2)$, décrit le jeu de données DL' résultant de l'élagage de DL pour ces valeurs de k et ϵ ($\mathcal{B}, DL' =$ **conf-FDFA**(DL, k, ϵ)).

Les lignes portant la mention (GD) décrivent les résultats pour les données générées. Ainsi une ligne $((GD), k = 1, \epsilon = 0.2)$, décrit D'' généré à partir de l'automate $\mathcal{B} =$ **conf-PDFA-t**$(DL, 1, 0.2)$.

Enfin une ligne $((GD), k = 1, AE)$, décrit un échantillon de données générées à partir de l'automate $\mathcal{A} =$ **PDFA-t**$(DL, 1)$ qui correspond à l'automate 1-*testable* sans élagage.

Paramètres du modèle Le tableau 3.2 rapporte la taille de l'alphabet ($|\Sigma|$), le nombre d'états ($|Q|$), le nombre de transitions (n_{trans}), le nombre de chaînes ($|D|$) ainsi que la taille totale des chaînes ($||D||$) pour différentes valeurs de k et ϵ. Lorsqu'il y a élagage, les valeurs reportées sont celles de l'automate et de l'échantillon élagué. Tout d'abord, on remarque comme attendu que le nombre d'états croît exponentiellement avec k.[3] Nous remarquons également (comme attendu) que le nombre de chaînes conservées réduit avec ϵ ; en effet, ϵ représente un seuil de tolérance, et plus le seuil est bas, plus le nombre de suppressions est important. De plus, le nombre de chaînes conservées diminue également lorsque k augmente : en effet, plus k augmente plus les fréquences se répartissent sur les états, et donc plus les variations de fréquences apportées par une chaîne au niveau d'un état sont susceptibles d'être importantes. Enfin, le rapport $||D||/|D|$ qui correspond à la taille

3. Notre implémentation considère un état de sortie supplémentaire à l'automate (sans conséquences supplémentaires) : pour cette raison, nos automates 1-*testables* ont deux états.

Paramètres du modèle			$\|\Sigma\|$	$\|Q\|$	n_{trans}	$\|D\|$	$\|\|D\|\|$
		Avant élagage	13783	2	13784	100236	2136320
		ϵ=0.01	0	0	0	0	0
		ϵ=0.02	925	2	926	21749	405501
	k=1	ϵ=0.05	4138	2	4139	69253	1328733
		ϵ=0.1	6906	2	6907	94169	1948794
		ϵ=0.2	8225	2	8226	98495	2080745
		ϵ=0.25	8329	2	8330	98721	2089766
		Avant élagage	13783	13785	193393	100236	2136320
		ϵ=0.01	0	0	0	0	0
		ϵ=0.02	0	0	0	0	0
	k=2	ϵ=0.05	4	6	20	113	754
		ϵ=0.1	54	56	121	444	3005
		ϵ=0.2	702	704	1618	3553	30085
		ϵ=0.25	1442	1444	3845	7413	67627
		Avant élagage	13783	184600	471356	100236	2136320
		ϵ=0.01	0	0	0	0	0
		ϵ=0.02	0	0	0	0	0
	k=3	ϵ=0.05	4	6	5	14	56
		ϵ=0.1	37	40	48	95	454
		ϵ=0.2	437	490	667	820	4852
		ϵ=0.25	791	952	1383	1551	9453
		Avant élagage	13783	434225	754333	100236	2136320
		ϵ=0.01	0	0	0	0	0
		ϵ=0.02	0	0	0	0	0
	k=4	ϵ=0.05	4	6	5	14	56
		ϵ=0.1	30	34	39	82	374
		ϵ=0.2	423	476	597	643	3702
		ϵ=0.25	785	930	1190	1119	6337
		Avant élagage	13783	697179	1002413	100236	2136320
		ϵ=0.01	0	0	0	0	0
		ϵ=0.02	0	0	0	0	0
	k=5	ϵ=0.05	4	6	5	14	56
		ϵ=0.1	30	34	39	82	374
		ϵ=0.2	421	493	612	640	3686
		ϵ=0.25	796	994	1254	1127	6407

TABLE 3.2 – Paramètres du modèle.

moyenne d'une chaîne diminue après l'élagage. Il est d'autant plus petit qu'ϵ est petit. Cela signifie que l'élagage a préserver la distribution des facteurs courts au détriment des plus longs. C'est une conséquence attendue de la méthode d'anonymisation. En effet, plus une chaîne est longue, plus il est probable d'utiliser une transition où elle est ϵ-*sensible*.

Nous observons un fort élagage : même pour les plus grosses valeurs de ϵ, par exemple pour $\epsilon = 0.01$, l'élagage conduit a une suppression complète du jeu de données. Ce phénomène s'explique par le caractère très épars de notre jeu de données. On remarque en particulier que le rapport $|D|/n_{trans}$ est proche de 1 lorsque que $k = 5$ dans ces conditions une transition n'est en moyenne atteinte que par une chaîne, ce qui conduit inévitablement à sa suppression car le facteur représenté par l'état est alors un quasi-identifiant trivial.

Même dans le cas où $k = 2$, le rapport reste faible (un peu en dessous de 2). Cela signifie que beaucoup de ressources sont à elles seules des facteurs quasi-identificateurs, et seraient donc supprimées indépendamment de la valeur de ϵ. Pour cette raison, lorsque ϵ diminue, le rapport $|D|/n_{trans}$ croît, alors que le nombre d'états décroît. Il ne reste alors qu'un petit ensemble de chaînes conservées qui se partagent les mêmes états.

Dans le but d'alléger les tableaux pour la suite, nous ne conserverons que les valeurs $\epsilon = 0.2$ et $\epsilon = 0.25$ ainsi que les valeurs de k entre 1 et 4. Les résultats pour les autres valeurs de k et ϵ sont disponibles en Annexe 4.8.

Statistiques de longueur et de durée Les tableaux 3.3 et 3.4 rapportent les distributions pour la longueur et la durée des sessions pour les données réelles, les données restantes après l'étape d'élagage et les données finales générées.

Au sujet de la longueur des séquences, il est important d'observer que les données réelles contiennent quelques chaînes très longues. Ceci impacte fortement la moyenne : à titre d'exemple, la plus longue chaîne est de longueur 2609 alors que le 90-percentile est de 46. Pour cette, raison nous pensons qu'il est plus pertinent de comparer les distributions grâce aux percentiles qui sont des indicateurs plus robustes aux valeurs extrêmes comme la moyenne ou la variance.

Comme expliqué précédemment, obtenir une garantie de confidentialité sur cet ensemble de données implique la suppression de nombreuses chaînes peu fréquentes. Par conséquent, nous observons un écart en termes de distributions entre les données réelles et les données restantes qui se propagent sur les données générées. Nous observons que les chaînes les plus longues sont plus susceptibles d'être supprimées que les courtes, ce qui est une conséquence logique de notre algorithme d'élagage. Ainsi, les modèles les plus

élagués ont tendance à générer des chaînes plus courtes que celles des données réelles, plus l'élagage est important plus les chaînes générées sont courtes en moyenne.

			count	min	max	moy	écart	25%	50%	75%	90%
Données brutes			100236	4	2609	21.31	31.81	6	12	23	46
(RD)	k=1	ϵ=0.2	98495	4	1513	21.12	30.36	6	12	23	46
		ϵ=0.25	98721	4	1513	21.16	30.47	6	12	23	46
	k=2	ϵ=0.2	3553	4	89	8.46	6.81	5	6	10	15
		ϵ=0.25	7413	4	124	9.12	7.8	5	6	10	17
	k=3	ϵ=0.2	820	4	61	5.91	4.77	4	5	6	8
		ϵ=0.25	1551	4	61	6.09	4.2	4	5	6	9
	k=4	ϵ=0.2	643	4	61	5.75	4.84	4	5	6	8
		ϵ=0.25	1119	4	61	5.66	4.02	4	5	6	8
(GD)	k=1	AE	100000	1	207	22.43	21.69	7	16	31	51
		ϵ=0.2	100000	1	215	21.81	21.51	7	15	30	50
		ϵ=0.25	100000	1	192	21.92	21.31	7	15	30	49
	k=2	AE	100000	1	191	21.72	21.49	7	15	30	50
		ϵ=0.2	100000	1	82	8.42	7.61	4	6	11	18
		ϵ=0.25	100000	1	112	9.16	8.47	4	7	12	20
	k=3	AE	100000	2	217	21.39	20.43	7	15	29	48
		ϵ=0.2	100000	2	61	5.91	4.75	4	5	6	9
		ϵ=0.25	100000	2	61	6.08	4.26	4	5	7	10
	k=4	AE	100000	3	180	21.22	19.71	7	15	28	46
		ϵ= 0.2	100000	3	61	5.82	4.94	4	5	6	8
		ϵ= 0.25	100000	3	61	5.67	4.05	4	5	6	8

TABLE 3.3 – Distribution de longueurs des sessions pour les données brutes, non-élaguées et générées. (RD) désigne les données non-élaguées, (GD) les données générées et AE avant élagage.

Au sujet de la durée des séquences un phénomène similaire est observé : les séquences les plus longues en temps sont également celles qui sont le plus fréquemment élaguées. Cela s'explique par la corrélation attendue entre la longueur d'une séquence (nombre de REL accédées) et sa durée. Ainsi plus l'automate est élagué plus les séquences restantes sont de courte durée.

Les données générées quant à elles contrebalancent cet effet en générant des chaînes plus longues : cela est particulièrement remarquable pour le cas $k = 1$. En effet, avec $k = 1$, si a est la ressource la plus probable alors a^n est la chaîne de longueur n ayant la plus grande probabilité d'être générée. Or ce cas est empiriquement peu observé dans nos données : en pratique, peu d'utilisateurs visitent plusieurs fois successivement la même REL

(ce qui reviendrait sans doute à rafraîchir la page). Néanmoins, un cas intéressant qui semble fréquent dans nos données est celui des ressources *formulaires* pour lesquelles l'utilisateur doit soumettre une réponse qui produit un rafraîchissement de la page. Logiquement, ces ressources formulaires ont un temps d'accès assez important puisque l'utilisateur doit prendre le temps de rédiger ses réponses.

De plus, ces ressources formulaires sont très fréquentes dans nos données puisqu'elles constituent l'essentiel des ressources du site eUčbeniki qui est notre plus gros pourvoyeur de données.

			count	min	max	moy	écart	25%	50%	75%	90%
Données brutes			100236	<1s	18h57m	1h21m	1h57m	6m	30m	1h43m	4h3m
(RD)	k=1	ϵ=0.2	98495	<1s	17h1m	1h20m	1h57m	6m43s	30m	1h42m	4h1m
		ϵ=0.25	98721	<1s	17h1m	1h20m	1h57m	6m44s	30m	1h43m	4h1m
	k=2	ϵ=0.2	3553	<1s	12h15m	40m	1h21m	1m30s	9m52s	37m	1h53m
		ϵ=0.25	7413	<1s	12h15m	35m	1h11m	1m32s	9m12s	33m	1h38m
	k=3	ϵ=0.2	820	2s	11h55m	36m	1h26m	42s	5m55s	27m	1h35m
		ϵ=0.25	1551	2s	11h55m	33m	1h17m	41s	5m7s	25m	1h31m
	k=4	ϵ=0.2	643	2s	11h55m	28m	1h19m	34s	3m53s	19m	1h38s
		ϵ=0.25	1119	<1s	11h55m	23m	1h7m	29s	2m51s	15m	50m
(GD)	k=1	AE	100000	<1s	>2d	5h12m	5h28m	1h18m	3h31m	7h18m	12h22m
		ϵ=0.2	100000	<1s	>2d	5h12m	5h28m	1h18m	3h30m	7h19m	12h20m
		ϵ=0.25	100000	<1s	>2d	5d12h	5h27m	1h10m	3h31m	7h19m	12h17m
	k=2	AE	100000	<1s	>2d	3h	3h27m	35m	1h48m	4h11m	7h27m
		ϵ=0.2	100000	<1s	>3d	1h34m	2h27m	17m	50m	1h53m	3h38m
		ϵ=0.25	100000	<1s	>2d	1h25m	1h58m	16m	50m	1h50m	3h21m
	k=3	AE	100000	<1s	1d6h	2h19m	2h38m	29m49s	1h23m	3h14m	5h46m
		ϵ=0.2	100000	<1s	>3d	1h7m	2h15m	13m	27m	57m	2h37m
		ϵ=0.25	100000	<1s	>2d	1h5m	2h3m	14m	29m	1h1m	2h25m
	k=4	AE	100000	<1s	1d1h	2h1m	2h23m	24m56s	1h10m	2h45m	5h5m
		ϵ=0.2	100000	3s	>2d	49m	1h50m	11m	21m	43m	1h31m
		ϵ=0.25	100000	<1s	>3d	42m	1h27m	11m	21m	42m	1h18m

TABLE 3.4 – Distribution de durées des sessions pour les données brutes et les données générées dans les cas élaguées et non élaguées. (RD) désigne les données non-élaguées, (GD) les données générées et AE avant élagage.

Extraction de motifs séquentiels fréquents Nous allons ici chercher à évaluer la capacité de notre approche à conserver la distribution des facteurs courts les plus fréquents dans les données.

Pour cela, nous considérons la tâche d'extraction de motifs séquentiels fréquents, comme défini dans la littérature (Chen et al., 2012). Dans notre contexte, nous appelons *motifs séquentiels fréquents* des facteurs de longueur minimale de 2 très fréquents dans le jeu de données réelles [4]. Nous sommes intéressés par le top $N = \mathcal{N}$ des motifs séquentiels les plus fréquents de D où $\mathcal{N}$ est un entier. Ces motifs séquentiels sont calculés à l'aide de l'algorithme PrefixSpan (Pei et al., 2001). Nous présentons le taux de vrais po-

4. Plus formellement nous ne considérons que les motifs générateur et fermés. Un motif est générateur s'il n'existe pas de sous-motif ayant la même fréquence. Un motif est fermé s'il n'existe pas de supermodèle de même fréquence.

sitifs comme le pourcentage des $\mathcal{N}$ motifs les plus fréquents correctement identifiés (c.a.d dire présents dans le top $\mathcal{N}$ des motifs séquentiels des données réelles et générées), en désignant par $\mathcal{F}_{\mathcal{N}}(D)$(respectivement $\mathcal{F}_{\mathcal{N}}(\tilde{D})$) l'ensemble des $\mathcal{N}$ motifs séquentiels les plus fréquents dans D (respectivement $\tilde{D}$), nous calculons le taux de vrais positifs comme :

$$TPR = \frac{|\mathcal{F}_{\mathcal{N}}(D) - \mathcal{F}_{\mathcal{N}}(\tilde{D})|}{\mathcal{N}}$$

			$\mathcal{N}$=20	$\mathcal{N}$=40	$\mathcal{N}$=60	$\mathcal{N}$=80	$\mathcal{N}$=100
Données générées	k=1	AE	0.05	0.03	0.03	0.04	0.03
		ϵ=0.2	0	0	0.03	0.06	0.05
		ϵ=0.25	0.05	0.025	0.06	0.05	0.06
	k=2	AE	0.70	0.75	0.78	0.71	0.75
		ϵ=0.2	0.25	0.28	0.27	0.24	0.24
		ϵ=0.25	0.35	0.30	0.32	0.29	0.28
	k=3	AE	0.25	0.45	0.48	0.48	0.51
		ϵ=0.2	0.00	0.00	0.05	0.09	0.07
		ϵ=0.25	0.25	0.23	0.25	0.25	0.23
	k=4	AE	0.25	0.32	0.43	0.50	0.49
		ϵ=0.2	0.10	0.13	0.10	0.10	0.09
		ϵ=0.25	0.10	0.13	0.12	0.10	0.10

TABLE 3.5 – Taux de vrais positifs dans les $\mathcal{N}$ motifs les plus fréquents.

Les résultats pour différentes valeurs de $\mathcal{N}$ sont rapportés ; nous observons dans le tableau 3.5 que même sans élagage, l'automate *k-testable* conserve moins de 50% des motifs les plus fréquents. Néanmoins, alors que l'élagage conserve moins de 0.1% des transitions à partir de $k = 2$, les données générées depuis l'automate élagué présentent des taux de vrais positifs comparables à ceux de l'automate non-élagué. Par exemple, pour $k = 2$ et $\epsilon = 0.25$, alors que plus de 98% des transitions sont supprimées, seulement 30% des motifs sont perdus. Ce résultat montre que l'algorithme d'élagage permet -malgré une forte compression de l'automate- de conserver une partie importante des motifs séquentiel les plus fréquents capturés par l'automate *k-testables* de départ.

Par ailleurs, on remarque que les meilleurs résultats sont atteints pour $k = 2$; cela s'explique grandement par le fait que la majorité des motifs les plus fréquents sont de longueur 2 et 3. Néanmoins, en considérant un plus grand nombre de motif les performances relatives augmentent avec k cela s'explique par le fait que les motifs plus longs (taille 3 et 4) sont en moyenne moins fréquents que les motifs de taille 2. Enfin, comme

cela est attendu, l'élagage a un effet néfaste sur le nombre de motifs conservés. Ainsi plus l'élagage est important (c.a.d plus ϵ est petit) plus le taux de motifs conservés décroît.

3.7 Conclusion

Dans ce chapitre, nous avons présenté un nouveau type d'automates à états finis, qui sont à la fois temporisés et probabilistes (PTk-TA). L'apprentissage de ces automates à partir d'un ensemble de traces d'apprentissages est possible grâce à des techniques d'apprentissage d'automates *k-testables*, adaptées ici pour prendre en compte les durées et les probabilités.

Nous montrons comment ce type d'automate est capable de générer efficacement un ensemble de données arbitrairement grand avec des caractéristiques statistiques proches de l'ensemble de données initial. Nous utilisons des idées issues du domaine de la recherche sur la confidentialité différentielle pour analyser l'automate obtenu et supprimer les points de données qui auraient typiquement une trop grande importance et pourraient donc être identifiés à partir de l'automate ou des nouvelles données générées.

Nous discutons également de certaines propriétés de cet algorithme.

Enfin, nous réalisons une étude de cas sur un grand ensemble de traces d'apprentissages récoltées en situation réelles (données X5gon). Les résultats montrent que même dans le contexte d'un alphabet très large, la méthode susmentionnée est toujours capable d'identifier une partie des motifs les plus fréquents dans l'ensemble de données.

En perspectives, nous sommes intéressés par l'application de notre méthode aux jeux d'évaluations classiques de la littérature, mais aussi par l'application des méthodes classiques de la littérature au jeu de données X5gon.

Lors des expérimentations, il est apparu que les données X5gon étaient extrêmement éparses. Cela conduit à un élagage très poussé pour résoudre les problèmes de réidentification. Parmi les stratégies envisageables pour gérer ces questions, citons le *clustering* et le *back-off*.

Le clustering consisterait à regrouper les ressources en clusters et d'utiliser un alphabet de clusters pour décrire les données. Cela permettrait de réduire la taille de l'alphabet et donc d'avoir un jeu de données moins épars. Les méthodes de catégorisation des documents semblent toutes indiquées pour former les clusters. Lors de la génération la véritable ressource pourrait alors être tirée dans le cluster en accord avec la distribution réelle.

Une autre approche pourrait considérer une stratégie de back-off (Katz, 1987; Manning

and Schütze, 1999). En effet, pour de faibles valeurs de k très peu de séquences sont élaguées mais les jeux de données générées s'éloignent la distribution réelle. La hiérarchie d'automates k-testables pourrait alors être utilisée en génération pour bénéficier de la plus grande généralisation des k-testables offerte par les petites valeurs de k mais également de la précision des k-testables pour des valeurs de k plus grandes.

Chapitre 4

Prédiction d'ordre sur des séquences d'apprentissages

4.1 Introduction

La recommandation de Ressources Éducatives Libres(RELs) dans un contexte comme celui d'X5gon présente toutes les contraintes d'une application à large-échelle. Le nombre de REL disponibles ne cesse d'augmenter et dépasse largement le million [1]. Le nombre d'utilisateurs potentiels est important et le système doit être conçu pour interagir avec des millions de requêtes par seconde (Section 0.2.4). Dans ce contexte « large échelle », les systèmes de recommandations prédominants utilisent une architecture à deux niveaux. Un premier niveau doit sélectionner un ensemble de ressources candidates susceptibles d'être pertinentes, alors qu'un deuxième niveau est en charge d'ordonnancer cette liste de ressources candidates.

Différentes distances ont été démontrées efficaces pour retrouver des ressources similaires (Le and Mikolov, 2014; Kusner et al., 2015), et peuvent être utilisées pour générer des ressources candidates à la recommandation. L'étape d'ordonnancement de l'algorithme - qui a pour but d'ordonnancer l'ensemble des candidats présélectionnés - pose un problème pour notre tâche, car elle nécessite de pouvoir définir une fonction objectif à maximiser. Elle est pourtant l'étape clé du processus, qui influe le plus sur les recommandations que les utilisateurs vont effectivement avoir (Granka et al., 2004). Dans les applications commerciales, ces fonctions objectifs sont définies en alignement avec les objectifs commerciaux de la plate-forme (augmenter le nombre de ventes, le temps passé sur la plate-forme, etc). Le but de la recommandation pédagogique est d'offrir à l'utilisateur une expérience d'apprentissage satisfaisante : il est beaucoup plus difficile de traduire cela en une fonction objectif à maximiser. De plus, les fonctions objectifs habituellement maximisées - que

1. Source : https://stateof.creativecommons.org/

nous avons présentées en Section 1.3.3.2 (taux d'engagement, taux de conversion, taux de clics, etc) - sont sujettes à un ensemble de biais non souhaitables dans un contexte éducatif. En particulier, la maximisation de l'engagement peut conduire l'algorithme à recommander des contenus viraux tels que la violence, les fake news, les théories pseudos-cientifiques (Allgaier, 2019) ou conspirationnistes (Rieder et al., 2018) [2]. En outre, leurs calculs nécessitent un historique des interactions des utilisateurs avec la plate-forme. Ces fonctions objectifs sont donc inopérantes dans un contexte de démarrage à froid.

Enfin, dans les applications commerciales, il suffit souvent de recommander les ressources une à une, à la volée. En reformulant, nous pouvons choisir de nous concentrer sur l'objectif à court terme qui est de maximiser localement notre fonction objectif et de passer d'une ressource à l'autre. On peut argumenter que l'apprentissage n'a pas que des objectifs immédiats et qu'il s'agit plutôt d'objectifs à moyen ou long terme. Par conséquent, un défi consiste à fournir à l'utilisateur un chemin d'apprentissage à travers plusieurs ressources. Cependant, la recommandation à long terme induit de pouvoir trouver un ordre dans lequel les ressources sont proposées. Plusieurs informations peuvent servir à inférer cet ordre : la difficulté des ressources ou les thématiques abordées, par exemple. Bien sûr, ces informations ne sont pas toujours disponibles, même sur des données hors-lignes (Urdaneta-Ponte et al., 2021). Néanmoins, les objets d'apprentissage en eux-mêmes et leur structuration sémantique constituent des informations pouvant s'avérer pertinentes et plus facilement accessibles. En particulier, dans le contexte du projet X5GON, il a été démontré qu'un texte brut représentant le contenu de la ressource était récupérable de manière automatique, contrairement aux métadonnées riches mentionnées plus haut (Section 2.5.2). Pour une tâche de recommandation à visée pédagogique sur des données non structurées, il est nécessaire de se focaliser sur des informations fréquemment présentes. Dans le cas contraire, seule une fraction de ressources riches en métadonnées serait susceptible d'être recommandée. Un défi consiste donc à trouver un modèle d'ordonnancement capable de s'adapter à la problématique de la recommandation à usage pédagogique. En particulier, les méthodes d'apprentissage automatique semblent incontournables dans l'apprentissage de tels modèles. Ces méthodes requièrent des ensembles de données d'entraînement ainsi que des tâches permettant de les évaluer. Dans le contexte commercial, nous venons de le voir, ces rôles sont respectivement remplis par les données historiques des utilisateurs captées sur les sites et par des fonctions objectifs héritées du modèle économique des plateformes. Dans le contexte éducatif, nous pensons que les séries de

2. https://algotransparency.org/

ressources produites par les enseignants peuvent constituer des ensembles d'entraînement ne souffrant pas du problème du démarrage à froid. Il s'agit donc d'apprendre l'ordre relatif de contenu textuel (les ressources des enseignants) depuis une vérité terrain donnée (l'ordre interne de la série) et d'inférer ensuite l'ordre de ressources non organisées en séries (les RELs). Pour ce faire, il est indispensable d'évaluer la capacité du modèle d'ordonnancement appris à généraliser l'ordre au-delà du contexte d'une série donnée.

Dans cette section, nous proposons donc un algorithme d'apprentissage permettant d'apprendre un *ordre cohérent de consommation des ressources*, en s'appuyant sur des séries de ressources ordonnées par des experts (en l'occurrence des séries de conférences). Pour cela, nous modélisons la phase de ré-ordonnancement comme une tâche de classification binaire des paires de ressources. Dans le but d'évaluer la capacité d'un tel modèle à prédire un ordre de consommation cohérent, nous définissons trois tâches d'évaluations. La première évalue la capacité d'un modèle à inférer l'ordre à l'intérieur du contexte d'une **série connue** et relativement à des **ressources connues**. La deuxième évalue la capacité du modèle à inférer l'ordre à l'intérieur du contexte d'une **série connue**, mais relativement à **des ressources inconnues**. Enfin, la troisième tâche évalue la capacité du modèle à inférer l'ordre **sans connaissance préalable ni des ressources, ni des séries**. Nous évaluons un modèle de référence et un modèle exploitant la construction du discours nouvellement introduit (TANN pour Timeline Aware Neural Network qui sera présenté en Section 4.4.2) sur ces trois tâches. Les résultats montrent la faisabilité de la tâche même dans un contexte agnostique (tâche 3), démontrant qu'un ordre cohérent de consommation peut être appris indépendamment du domaine et du locuteur. De plus, ces résultats accréditent l'hypothèse d'artefacts dans la construction du discours permettant d'inférer cet ordre sans connaissance du domaine. Le modèle exploitant la construction du discours (TANN) surpasse le modèle de référence, obtenant en particulier 80% de bonnes prédictions sur les ordres des paires à l'intérieur d'une série connue. Enfin, le modèle infère correctement l'ordre de 69% des paires de ressources provenant pour une série inconnue avec en particulier un domaine et un locuteur différent des données d'entraînements. Ce dernier résultat milite en faveur de l'application de ce modèle pour la tâche de ré-ordonnancement dans un contexte de recommandation.

La suite du chapitre est organisée comme suit : en 4.2 nous discuterons de la pertinence de l'approche et en particulier de l'utilisation de série de conférences comme vérité terrain relatant d'un ordre de consommation cohérent des ressources. En section 4.3 nous définirons plus précisément les trois nouvelles tâches d'évaluation et en section 4.4 nous

présenterons les deux modèles utilisés. La section 4.6 présente les performances des deux modèles sur les tâches d'évaluation, l'environnement expérimental utilisé (Section 4.5) sur le corpus YaleOpenCourseware (précédemment présenté en Section 2.6.2). La section 4.7 contient une discussion sur les applications pratiques de nos résultats ainsi que sur les perspectives qu'ils suscitent.

4.2 Approche

Dans la suite de ce travail, nous allons faire l'hypothèse de l'existence d'un ordre cohérent de consommation des ressources pédagogiques. Naturellement, cet ordre dépend de nombreux facteurs, dont les connaissances ou les compétences personnelles de l'apprenant, le temps dont il dispose pour apprendre, le but qu'il poursuit, son intention. Parfois, il se peut même qu'aucun ordre raisonnable n'existe : des ressources indépendantes issues de domaines complètement différents n'auront probablement pas de meilleur ordre. Au contraire, des ressources similaires r et r' pourraient admettre à la fois r suivi de r' et r' suivi de r selon des critères très différents.

Néanmoins, il n'est pas invraisemblable de supposer qu'un consensus humain existe dans un certain nombre de cas. À titre d'illustration, il est consensuel que l'addition doit être apprise avant la multiplication ou que si l'enseignant a défini un ordre pour ses cours, cet ordre prédéfini doit être suivi. Plus précisément, nous pensons que l'ordre d'apprentissage défini par les enseignants lors de la conception d'une série de lectures est un exemple d'ordre satisfaisant de consommation des ressources pédagogiques. Dans un contexte comme celui de la recommandation pour lequel les ressources à comparer abordent des sujets similaires, on peut supposer qu'un ordre de consommation préférentiel existe dans de nombreux cas. A minima, il existe de nombreux cas identifiés dans lesquels apprendre cet ordre est pertinent (série de cours, organisation en prérequis, etc).

Dans ce travail, nous supposons que l'ordre d'apprentissage défini par les enseignants lors de la conception d'une série de cours est un exemple d'ordre cohérent de consommation des ressources pédagogiques. Bien sûr, cet ordre n'est pas une vérité immuable sur la manière d'organiser un ensemble de ressources. Tout au contraire, on peut imaginer deux enseignants choisissant d'ordonnancer différemment une même série de ressources. Néanmoins, nous cherchons aussi ici à apprendre comment ne pas ordonnancer les ressources. Dans cette optique, il s'agit en pratique d'éviter les ordres incohérents du point de vue pédagogique ou sémantique. En effet, nous pensons que certains ordres sont à

proscrire de manière consensuelle. En pratique, nous supposons même que dans de nombreux cas, l'ensemble des ordres satisfaisants représente une très faible proportion de l'ensemble des ordres possibles pour une série de ressources. Malheureusement, les ressources du réseau Global OER ne sont pas toujours organisées en séries de cours. En pratique, les liens de précédences entre les ressources sont difficiles à récupérer de manière générique et automatique (Section 2.5.2). Compte tenu de la quantité de ressources, de l'éventail de domaines et de la vitesse de croissance de ce réseau mondial de REL, il serait très difficile de s'en remettre à l'annotation humaine pour le diriger. Récemment, des initiatives d'annotation participative ont commencé à émerger (Florilège [3], Tournesol [4]) mais à ce jour aucun jeu de données public n'est disponible à notre connaissance. Étant donnés les défis rencontrés avec l'absence de vérité de terrain et la difficulté d'une tâche d'étiquetage, nous suggérons d'utiliser des séries de ressources construites par des enseignants comme vérité de terrain. Pour ce faire, nous utiliserons le corpus YaleOpenCourseware que nous avons constitué pour l'occasion et mis à disposition en accès libre (`https://gitlab.univ-nantes.fr/connes-v/yaleocw-corpus`). La constitution et la mise en accès libre de ce corpus constitue une contribution importante dans le domaine de la recommandation à visée pédagogique tant l'absence de corpus a été identifiée comme une problématique centrale du domaine (Urdaneta-Ponte et al., 2021). Le lecteur souhaitant en savoir plus sur le processus de constitution du corpus et sur ses spécificités peut se référer à la Section 2.6.2. Pour la suite, nous considérerons donc un jeu de données de paires de ressources disposant d'un *ordre d'apprentissage cohérent* connu.

Pour cette raison, nous choisissons de formaliser notre problème comme un problème d'apprentissage pour lequel nous essayons d'apprendre un modèle prédictif d'un ordre d'apprentissage cohérent des ressources à partir des échantillons provenant de séries de cours du corpus. Afin de dupliquer le nombre d'exemples d'apprentissage et sans perte de généralisation, nous formalisons la tâche comme une classification binaire de la précédence des paires de ressources. Ainsi, le modèle produira pour chaque paire de ressource r et r' une prédiction binaire fonction des paramètres du modèle, que nous noterons $\hat{o}(r, r'|\theta_M)$ avec θ_M vecteur de paramètres du modèle. On interprétera la prédiction de la manière suivante : $\hat{o}(r, r'|\theta_M) = 1$ signifiera que l'ordre de consommation cohérent prédit est r puis r'. Inversement, $\hat{o}(r, r'|\theta_M) = 0$ signifiera r' puis r. Dans le cas des séries de notre jeu de données, l'ordre cohérent de consommation est connu, nous utiliserons la fonction

3. Section 2.6.1
4. Site de recommandation collaborative de contenu : `https://tournesol.app/`

o pour représenter cet ordre. De la même manière, $o(r, r') = 1 \Leftrightarrow o(r', r) = 0$ signifiera que l'ordre de consommation cohérent observé est r puis r' et $o(r', r) = 1 \Leftrightarrow o(r, r') = 0$ signifiera r' puis r. Dans le cas de la prédiction, la fonction $\hat{o}$, cherchera donc à estimer la fonction o sur les paires connues. Pour simplifier la notation, o a valeur dans $\{0, 1\}$ alors que $\hat{o}$ a valeur dans $[0, 1]$.

4.3 Tâches

Comme expliqué dans la section 4.1, les approches habituelles de la recommandation à grande échelle présentent plusieurs inconvénients. Tout d'abord, et c'est le plus problématique, ces approches ont besoin d'un grand nombre d'interactions utilisateur collectées afin d'apprendre un bon modèle d'ordonnancement pour la recommandation. Dans notre cas, nous ne disposons pas encore d'un tel historique d'interactions, de sorte que notre problème peut être assimilé à un problème de démarrage à froid. Deuxièmement, il a été démontré que les fonctions objectifs habituellement employées dans ces approches ne favorisent pas le meilleur contenu, mais le plus viral (Allgaier, 2019; Rieder et al., 2018). Nous voulons naturellement éviter ce cas, et souhaitons recommander de la manière la plus pédagogique possible.

Pour ce faire, nous utilisons des séries existantes de conférences faites par des enseignants. Plus précisément, nous cherchons à prédire l'ordre logique de consommation de deux cours aléatoires tirés d'une série donnée. Nous posons l'hypothèse suivante : l'organisation des cours faite par le professeur est un bon exemple d'ordre de consommation cohérent des ressources. Par conséquent, nous utilisons cet ordre comme vérité terrain. De plus, un algorithme d'ordonnancement capable d'apprendre cet ordre semble un bon candidat pour la recommandation pédagogique, raison pour laquelle nous utilisons cette tâche comme *proxy* pour la recommandation pédagogique. Enfin, nous pensons qu'un ordre d'apprentissage cohérent peut être appris en dehors du cadre d'une série. Nous chercherons à apprendre cet ordre indépendamment du ou des domaines traités par les ressources. Une autre approche possible aurait pu être d'apprendre un modèle spécifique à chaque domaine, nous discuterons ce choix en Section 4.7. En d'autres termes, nous supposons l'existence de modèles généraux permettant une meilleure continuité pédagogique. Pour vérifier cette dernière hypothèse, nous définissons trois tâches à difficultés croissantes permettant de mesurer la quantité d'informations contextuelles apprises à partir de la sé-

rie donnée (*tâches paires et ressources agnostique*) et d'informations non contextuelles apprises à partir d'autres séries vues pendant l'entraînement (*tâche série agnostique*). La tâche série agnostique est sans doute la plus intéressante dans notre cas, car elle évalue la capacité du modèle à ordonner une toute nouvelle série de cours d'un nouveau professeur. Détaillons les trois différentes tâches en question :

Tâche 1 : prédire avec des informations contextuelles sur les paires. Dans le premier cas, l'objectif est de prédire l'ordre des ressources d'une série connue lorsqu'un nouvel épisode (inconnu) est ajouté. Pour ce faire, l'ensemble de données est divisé en un ensemble d'entraînement, noté X_T, et un ensemble de test, noté X_S, de manière que chaque paire d'un nouvel ensemble, noté X_{TS}, soit composée d' **exactement un épisode** présent dans X_T et d'un épisode absent de ces paires, c'est-à-dire présent dans X_S. Ainsi, pour chaque paire, le modèle n'aura vu qu'un seul des deux épisodes durant sa phase d'entraînement. La tâche évalue ainsi la capacité d'un modèle prédictif à retrouver l'ordre de précédence d'un épisode inconnu par rapport à un épisode connu à l'intérieur d'une série connue. On parlera de *tâche épisode agnostique*. Il est intéressant de noter que par elle-même, cette tâche pourrait être utilisée comme un système de recommandation dans un scénario dans lequel un enseignant avec un cours construit veut introduire une nouvelle ressource dans son cours. L'algorithme pourrait alors être utilisé pour recommander une position préférentielle pour la nouvelle ressource.

FIGURE 4.1 – Tâche 1 : épisode agnostique.

Tâche 2 : prédire avec des informations contextuelles sur le cours. L'inconvénient de la tâche précédente est qu'un algorithme qui ignore complètement les caractéristiques de la nouvelle ressource dans la paire pourrait obtenir de bons résultats en apprenant simplement les positions relatives des autres ressources d'apprentissage. Supposons par exemple une série de 10 ressources numérotées de 1 à 10 et un modèle se contentant de prédire l'ordre relatif de la paire en fonction de l'espérance de l'ordre relatif observé dans les données d'entraînement. En supposant que la ressource 10 fasse partie des données d'entraînement, un tel modèle prédira correctement l'ordre d'une paire contenant la ressource 10 avec une probabilité 1, indépendamment de la deuxième ressource. En effet, cette ressource 10 ayant toujours été observée comme postérieure à toute autre ressource, elle sera prédite comme telle. Indépendamment de l'autre ressource considérée dans l'ensemble test, le résultat sera donc correct. Ceci est clairement problématique pour notre cas d'application. Pour évaluer la différence entre un tel modèle et un apprenant tirant parti des caractéristiques de la nouvelle ressource, nous avons donc défini la tâche 2. Une fois de plus, nous construisons un ensemble de données de paires de ressources provenant d'un même cours ; cet ensemble de données est donc divisé en un ensemble X_T et un ensemble X_S de telle sorte qu'une paire appartenant à un ensemble, noté X_S^2, ne contient **aucun élément** utilisé pendant la phase d'entraînement du modèle, c'est-à-dire aucune ressource de X_T. Contrairement, à la *tâche épisode agnostique*, le modèle n'a vu ici aucun des deux épisodes constituant la paire durant sa phase d'entraînement. Ainsi, si la série des ressources est partiellement connue, les deux ressources en questions sont nouvelles pour le modèle. La tâche évalue ainsi la capacité d'un modèle prédictif à retrouver l'ordre de précédence de deux épisodes inconnus à l'intérieur d'une série connue. On parlera de *tâche paire agnostique*.

Tâche 3 : tâche série agnostique. Enfin, nous concevons la dernière tâche pour qu'elle soit aussi proche que possible de notre objectif. Dans cette tâche, nous cherchons à évaluer dans quelle mesure la généralisation de l'ordre peut être apprise en dehors du cadre de la série d'entraînements. En d'autres termes, nous voulons mesurer la capacité du modèle à prédire correctement le classement des épisodes appartenant à une série non vue. Ici, le modèle n'a vu aucun des deux épisodes constituant la paire durant sa phase d'entraînement, de plus, il n'a également vu **aucune autre paire de leurs séries**. Nous appelons cela la *tâche série agnostique*. Puisque nous n'avons aucune connaissance des séries testées, nous supposons qu'il y a suffisamment d'informations dans le discours et la

FIGURE 4.2 – Tâche 2 : paire agnostique.

composante didactique pour inférer l'ordre entre deux épisodes de n'importe quelle série.

Pour ce faire, les ensembles X_T et X_S sont naturellement disjoints, mais aucune paire de l'ensemble d'entraînement ne partage de ressources avec l'ensemble de test. À la différence de la tâche 2, nous imposons une contrainte supplémentaire sur ces deux sous-ensembles : une série n'appartiendra qu'à un unique sous-ensemble (soit X_T, soit X_S).

En résumant de manière simplifiée, dans la première tâche, nous retirons exactement un épisode de chaque série pour l'ensemble de test, et nous combinons cet épisode avec un épisode de l'ensemble X_T pour construire une paire de test ; dans la deuxième tâche, deux épisodes par série sont retirés avant l'entraînement et ils constituent la paire de test. Enfin, dans le troisième cas, une série entière est supprimée avant l'entraînement et les paires de cette série seront utilisées comme paires de test. Pour des raisons pratiques, les trois protocoles ci-dessus ont été assouplis : au lieu de supprimer 1 ou 2 épisodes pour notre ensemble de test, nous supprimons toujours une partie fixe de 10% des épisodes. De plus, nous introduirons également un ensemble de validation pour chacune des tâches.

4.4 Les modèles

4.4.1 Modèle de référence

Nous choisissons comme approche de référence une représentation Doc2Vec suivie d'un perceptron multicouche (FNN). Cette architecture est reconnue comme une approche de

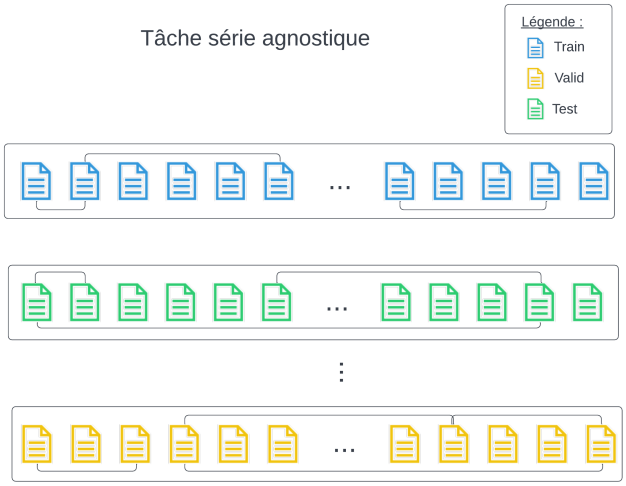


FIGURE 4.3 – Tâche 3 : série agnostique.

référence dans les domaines de la classification de texte et plus largement pour les tâches de haut niveau en traitement du langage naturel (Minaee et al., 2021). La première étape de l'approche de référence consiste à entraîner un modèle Doc2Vec global sur les ressources. Pour représenter les ressources, nous utilisons un plongement sémantique de leur contenu textuel en appliquant l'algorithme Doc2Vec. Ainsi, chaque ressource r^i est représentée par un vecteur $E^i \in \mathbb{R}^{d_e}$, où d_e est la dimension choisie pour la représentation Doc2Vec.

Cette nouvelle représentation alimente directement un FNN contenant L couches, suivi d'une unité de soft-max. Le FNN modélise donc une fonction $FNN : \mathbb{R}^{2d_e} \rightarrow [0, 1]$ en charge d'inférer à partir de la concaténation des plongements sémantiques des deux ressources leur ordre relatif. La valeur de sortie obtenue est interprétée comme la probabilité de la précédence de la ressource r par rapport à la ressource r'.

La Figure 4.4.1 résume l'architecture du modèle de référence.

Plus formellement, pour chaque paire de ressources (r^i, r^j), un vecteur d'entrée est construit par concaténation de leurs vecteurs correspondants E^i, E^j. Le modèle global fait correspondre à chaque paire de ressources (r^i, r^j) une valeur dans l'intervalle unité $[0, 1]$.

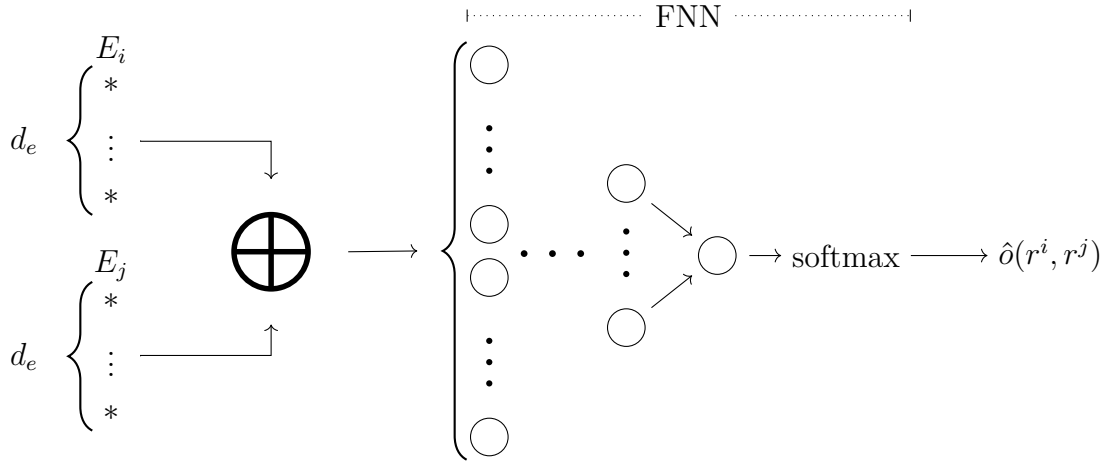


FIGURE 4.4 – Architecture du modèle de référence.

Plus précisément, le FNN effectue la transformation suivante :

$$h_0^{\text{FNN}} = \sigma(W^0[E^i \oplus E^j] + b^0), \tag{4.1}$$

$$h_l^{\text{FNN}} = \sigma(W^l h_{L-1}^{\text{FNN}} + b^l), \tag{4.2}$$

$$\hat{o} = \text{sigmoïde}(h_L^{\text{FNN}}), \tag{4.3}$$

avec $W^0 \in \mathbb{R}^{2d_e \times d_h^{\text{FNN}}}$, $\forall l \in \{1 \dots L-1\} W^l \in \mathbb{R}^{d_h^{\text{FNN}} \times d_h^{\text{FNN}}}$, $W^L \in \mathbb{R}^{d_h^{\text{FNN}} \times 1}$, $\forall l \in \{1 \dots L-1\} b^l \in \mathbb{R}^{d_h^{\text{FNN}}}, b^L \in \mathbb{R}$. L'opérateur $\oplus$ dénote la concaténation des vecteurs de représentations des ressources.

4.4.2 Notre modèle : TANN(Timeline Aware Neural Network)

Le modèle TANN est un modèle neuronal dont la spécificité est d'utiliser une représentation sémantique des ressources préservant la chronologie. Nous pensons que la chronologie a un rôle important à jouer dans la tâche de prédiction d'ordre à plusieurs égards. Par exemple, il est fréquent qu'en début de cours un enseignant fasse mention des prérequis de la semaine passée, ou qu'en fin de cours, il fasse une ouverture sur le sujet de la semaine suivante. Par ailleurs, l'apprentissage est souvent une construction incrémentale articulée autour de prérequis. De plus, certaines approches d'apprentissage se fondent sur une narration chronologique des évènements, des découvertes ou des connaissances. Dans le cas de vecteurs plats comme ceux de Doc2Vec, le document est considéré comme un ensemble non ordonné. L'ordre des mots à l'intérieur du document et par extension

celui des phrases, des paragraphes et des idées ne sont pas pris en compte. Ainsi, avec ces méthodes, les informations didactiques liées à la construction du discours sont donc perdues. Afin de prendre en compte la continuité pédagogique, la construction du discours et l'enchaînement des idées, nous proposons de construire une représentation sémantique des ressources préservant la chronologie. Pour ce faire, nous utilisons l'algorithme continuous-Doc2vec présenté en Section 2.3.1.3. Cette méthode consiste à découper le texte brut en un nombre choisi (*chunk_size*) de tronçons de la taille d'un mot. Chaque tronçon a un chevauchement de *overlap* mots avec le tronçon précédent. Nous entraînons ensuite un modèle Doc2Vec global sur tous les tronçons comme s'il s'agissait de documents distincts. Chaque ressource r^i est finalement représentée par une matrice $R^i \in \mathbb{R}^{d_e \times d_{r^i}}$, où d_e est la dimension choisie pour la représentation sémantique Doc2Vec et d_{r^i} le nombre de tronçons de la ressource r^i. Cette matrice est obtenue par concaténation des représentations sémantiques de chaque tronçon. Pour cette raison, pour une ressource donnée, la t-ème colonne de la matrice est la représentation sémantique du t-ème tronçon de la ressource. Dans la suite, nous désignons par R^i_t la représentation sémantique du t-ième tronçon de la ressource r^i, c'est-à-dire la t-ième colonne de R^i.

L'architecture neuronale proposée décrite dans la Fig. 4.5 est composée d'un réseau neuronal récurrent (RNN) suivi d'un perceptron multicouche (FNN). Le rôle du RNN est de projeter la représentation de la ressource, préservant la chronologie en un vecteur dense capturant les informations nécessaires pour prédire le positionnement de la ressource. Ce vecteur se doit alors de capturer complètement les aspects sémantiques et pédagogiques des ressources. Il modélise une fonction $RNN : \mathbb{R}^{d_e \times *} \rightarrow \mathbb{R}^{d_{inputs}}$, (en pratique, on peut fixer $d_{inputs} = d_e$) en charge de compresser l'information sémantique et didactique contenu dans la représentation vectorielle en un vecteur plat [5]. Le FNN prend en entrée une paire de ces vecteurs (sorties du RNN), et est entraîné à prédire l'ordre de la paire dans la série de cours. Il modélise donc une fonction $FNN : \mathbb{R}^{2d_{inputs}} \rightarrow [0, 1]$. Comme dans le cas du modèle de référence, la valeur de sortie obtenue est interprétée comme la probabilité de la précédence de la ressource r par rapport à la ressource r'.

Plus formellement, pour chaque paire de ressources, la représentation temporelle de chaque paire est traitée indépendamment par le RNN. Étant donné la matrice de représentation R^i d'une ressource donnée r^i, le RNN calcule des séquences de sorties $(y_0, \ldots, y_t, \ldots, y_{dr-1})$,

5. Ici le symbole $*$ signifie que la fonction modélisée peut s'appliquer sur une chaine indépendamment de sa longueur (dans notre cas le nombre de colonnes de la matrice R^i), c'est une propriété bien connue des RNN.

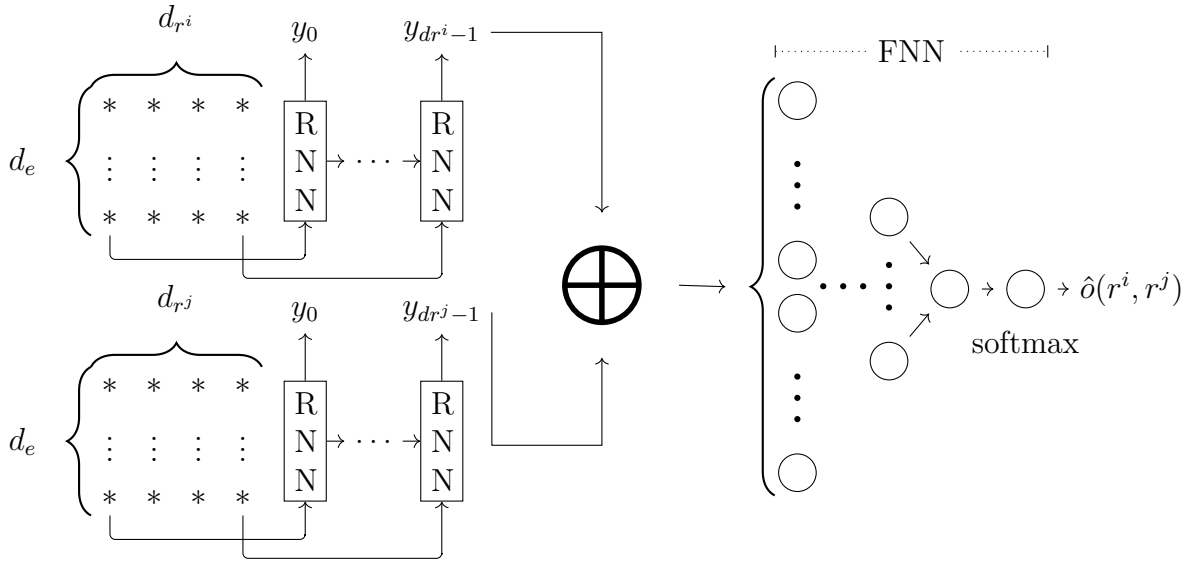


FIGURE 4.5 – Architecture du modèle TANN.

$(h_0^{\text{RNN}}, \ldots, h_t^{\text{RNN}}, \ldots, h_{dr-1}^{\text{RNN}})$ en itérant les équations suivantes :

$$h_t^{\text{RNN}} = \text{sigmoïde}(W^{hx} R_t^i + W^{hh} h_{t-1}^{\text{RNN}}) \tag{4.4}$$

$$y_t = W^{yh} h_t^{\text{RNN}} \tag{4.5}$$

avec W^{hx}, W^{hh}, W^{yh} les matrices de paramètres apprises pendant l'apprentissage. La dernière sortie de la matrice $y^i_{dr^i-1}$ est utilisée comme *représentation pédagogique dense de la ressource* r^i. Pour simplifier, nous introduisons la *matrice de représentation pédagogique* P comme $P = \bigoplus_i y^i_{dr^i-1}$ avec $\oplus$ l'opérateur de concaténation. Par définition, $P^i \equiv y^i_{dr^i-1}$.

Deuxièmement, pour chaque paire de ressources (r^i, r^j), les représentations pédagogiques correspondants (P^i, P^j) sont concaténées et directement données en entrée du réseau FNN. La suite de la procédure, se terminant par une unité soft-max, repose sur le même processus que celui décrit dans la section 4.4.1 :

$$h_0^{\text{FNN}} = \sigma(W^0[P^i \oplus P^j] + b^0), \tag{4.6}$$

$$h_l^{\text{FNN}} = \sigma(W^l h_{L-1}^{\text{FNN}} + b^l), \tag{4.7}$$

$$\hat{o} = \text{sigmoïde}(h_L^{\text{FNN}}), \tag{4.8}$$

avec L le nombre de couches du FNN et les paramètres de poids et de biais du réseau de neurones : $W^0 \in \mathbb{R}^{2d_{inputs} \times d_h^{\text{FNN}}}$, $\forall l \in \{1 \ldots L-1\} W^l \in \mathbb{R}^{d_h^{\text{FNN}} \times d_h^{\text{FNN}}}$, $W^L \in \mathbb{R}^{d_h^{\text{FNN}} \times 1}$,

$\forall l \in \{1 \dots L-1\} b^l \in \mathbb{R}^{d_h^{\text{FNN}}}, b^L \in \mathbb{R}$.

Enfin, pour le modèle de référence et le TANN nous entraînons le modèle neuronal global à l'aide d'une descente de gradient stochastique afin de minimiser la somme des erreurs quadratiques entre l'ordre estimé $\hat{o}(r, r')$ et l'ordre de vérité terrain observé dans la série $o(r, r')$:

$$\arg\min_{\theta_M} \sum_{\forall r, r' \in X_T \times X_T} [o(r, r') - \hat{o}(r, r' | \theta_M)]^2 \tag{4.9}$$

où X_T désigne l'ensemble d'entraînement.

4.5 Analyse expérimentale

Le code et toutes les informations nécessaires à la reproduction de nos expériences peuvent être trouvés sur `https://gitlab.univ-nantes.fr/connes-v/order_inference`. Notons $\mathcal{X}$ l'ensemble de toutes les ressources, où l'ordre partiel est donné par les séries de ressources construites par les enseignants. Ainsi, deux ressources d'une même série seront *comparables*, et auront donc un ordre d'apprentissage cohérent connu dans la série. S'ils sont issus de séries différentes, ils seront dits *incomparables*. Ensuite, nous construisons l'ensemble $P(\mathcal{X})$ de toutes les paires de ressources ayant une vérité terrain connue (i.e : paire des ressources de même série). Par définition, chaque paire (r, r') de cet ensemble admet un ordre d'apprentissage cohérent connu $o(r, r')$ dans $\mathcal{X}$ (i.e : les séries de cours faites par les enseignants).

$$P(\mathcal{X})) = \{(x, y) \in \mathcal{X}^2 \mid x \text{ et } y \text{ sont comparables}\}$$

4.5.1 Prédire avec des informations contextuelles

Pour les deux premières tâches (tâche épisode agnostique et tâche paire agnostique), nous divisons notre ensemble $P(\mathcal{X}))$ comme suit. Nous tirons aléatoirement 80 % des ressources pour l'ensemble d'entraînement et 10 % pour chacun des ensembles de validation et de test ; nous désignons ces ensembles respectivement par X_T, X_V et X_S. Comme illustré dans la Figure 4.6, nous construisons ensuite quatre ensembles disjoints à partir de paires de ressources :

X_T^2 : composé des paires ressources de X_T, $X_T \times X_T$. Dans cet ensemble, les deux

ressources de chaque paire seront vues durant l'entraînement. Il servira d'ensemble d'entraînement pour nos modèles ;

X_S^2 : composé des paires ressources de X_S, $X_S \times X_S$. Dans cet ensemble, aucune des deux ressources ne sera vue durant l'entraînement. Il servira d'ensemble de test pour la tâche paire agnostique ;

X_{TV} : composé des paires ressources dont un élément provient de X_T et un de X_V, $X_{TV} = X_T \times X_V$ (produit cartésien de X_T et X_V). Dans cet ensemble, dans chaque paire, une seule des deux ressources sera vue durant l'entraînement. Il servira d'ensemble de validation pour la tâche paire agnostique ;

X_{TS} : composé des paires ressources dont un élément provient de X_T et un de X_S, $X_{TS} = X_T \times X_S$. Dans cet ensemble, dans chaque paire, une seule des deux ressources sera vue durant l'entraînement. Il servira d'ensemble de test pour la tâche épisode agnostique.

Notez que tous les ensembles X_T^2, X_S^2, X_{TV}, X_{TS} sont des sous-ensembles de $P(\mathcal{X})$.

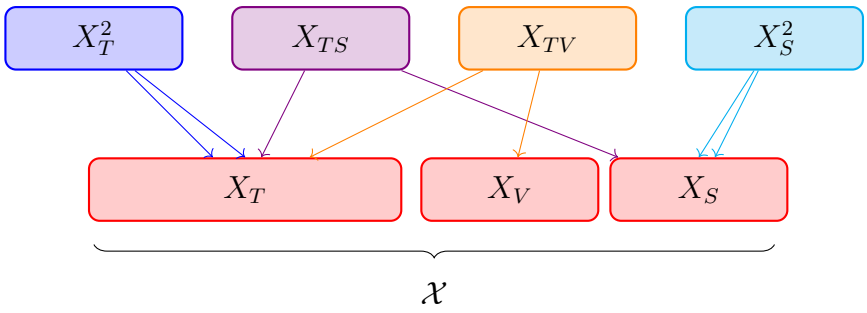


FIGURE 4.6 – Représentation de graphique du découpage entraînement, test, validation.

4.5.2 Prédire sans information contextuelle

Dans cette dernière tâche (série agnostique), nous ne savons rien de la série dont la paire testée a été extraite. Pour cela, nous divisons notre ensemble de données comme suit. Nous tirons au hasard 80 % des séries pour l'ensemble d'entraînement et 10 % pour l'ensemble de validation et l'ensemble de test. À partir de ce découpage, nous obtenons trois ensembles de paires : l'ensemble d'entraînement X_{TT}, l'ensemble de validation X_{VV} et l'ensemble de test X_{SS}. Nous attirons l'attention du lecteur sur le fait que les ensembles X_T^2 et X_{TT} sont différents, la contrainte supplémentaire par rapport à la tâche 2 sur cette tâche est qu'il n'existe pas de série pour laquelle des ressources appartiendraient à des

ensembles différents. Le même remarque s'applique respectivement au cas de X_S^2 et X_{SS} et à celui de X_V^2 et X_{VV}. Afin de prendre en compte les séries dispensées par un même enseignant, la division garantit que les séries d'un même enseignant se trouvent toujours dans des sous-ensembles différents.

4.5.3 Méta-paramètres

Comme pour les autres architectures neuronales, notre modèle et le modèle de référence héritent de plusieurs méta-paramètres qui doivent être réglés afin d'obtenir le meilleur apprentissage possible. Afin de les fixer, nous utilisons une approche de recherche par grille de valeurs et conservons le meilleur ensemble de méta-paramètres. Les paramètres du modèle ainsi sélectionnés servent ensuite à procéder à l'évaluation des performances à l'aide d'une validation croisée en 10 groupes. La validation croisée possède l'avantage de fournir des résultats plus robuste qu'une approche d'évaluation classique considérant une seule séparation arbitraire des ensembles de test, d'entrainement et de validation. Néanmoins, cette approche est rarement utilisée en conjonction avec des modèles neuronaux, cela s'explique par le fait que de nombreux modèles neuronaux profonds ont des temps d'entrainement ne permettant pas d'appliquer cette approche en pratique. Dans notre cas, nous travaillons sur un petit ensemble de données avec des modèles ayant peu de paramètres, cela permet d'obtenir un faible temps d'apprentissage des modèles (voir Tableau 4.3) et rend possible une évaluation grâce à une approche de validation croisée. L'époque retenue pour l'évaluation finale considère les meilleures performances sur l'ensemble de validation après 60 époques. Les meilleurs résultats obtenus dans nos expériences ont été obtenus avec l'architecture suivante : nous avons utilisé 4 couches LSTM dans notre RNN, le nombre de couches dans FNN a été fixé à 2, la dimension d'intégration d_e à 300, les fenêtres de contexte pour l'algorithme Doc2Vec à 5, la dimension cachée de FNN et de RNN a été fixée à $d_h^{FNN} = d_h^{RNN} = 100$, l'unité linéaire rectifiée (ReLU) a été utilisée comme fonction d'activation, un dropout général de 0.1 a été appliqué après chaque couche pendant l'étape de formation. L'optimisation des paramètres a été effectuée à l'aide de l'algorithme d'optimisation d'Adam avec un taux d'apprentissage de 10^{-4}, $\alpha = 0.9$, $\beta = 0.999$ suivant les directives de (Kingma and Ba, 2014). Sur la base de nos expérimentations, nous avons observé que cette configuration est meilleure ou équivalente à d'autres architectures. Pour la représentation continuousDoc2vec, nous avons choisi une taille de tronçons ($chunk\_size$)

de 1000 *mots* et un chevauchement (*overlap*) de 500 *mots*. [6]

Dans la section 4.6, les résultats sont présentés sur le modèle de référence défini dans la section 4.4.1, et notre TANN défini dans la section 4.4.2. Une extension possible de TANN est d'incorporer un mécanisme d'attention. Le mécanisme d'attention se matérialise par une matrice en charge de récupérer les différentes sorties du RNN et de les compresser dans un vecteur plat. Ainsi, au lieu de simplement utiliser la dernière sortie du RNN, il est possible d'agréger les informations obtenues tout au long du traitement de la séquence. Initialement, ce modèle a été introduit pour contrebalancer le phénomène d'évanescence du gradient observé dans les modèles récurrents. Le modèle obtenu est noté TANN+Att (un lecteur intéressé par les modalités d'incorporation d'un mécanisme d'attention sur une architecture RNN peut se référer à (Bahdanau et al., 2015)).

4.6 Résultats

Le tableau 4.1 nous donne le nombre d'exemples obtenus dans chaque ensemble pour chacune des tâches. On remarque que l'ensemble X_S^2 qui sera utilisé pour évaluer la tâche 2 est - par construction - nettement plus petit que les autres ensembles de validation.

		Nombre de paires
Tâches épisode et paire agnostique :	X_T^2	17263
	X_{TS}	4092
	X_S^2	251
	X_{TV}	4148
Tâche série agnostique :	X_{TT}	21753
	X_{SS}	2214
	X_{VV}	2553

TABLE 4.1 – Nombre de paires dans les différents ensembles (les tailles des ensembles peuvent légèrement varier en fonction du tirage aléatoire).

Pour chaque tâche, la précision d'un modèle M, de paramètre θ_M, sur un sous-ensemble spécifique S est calculée comme suit :

$$Précision(M, S) = \frac{\sum_{\forall (r,r') \in S} \lfloor \hat{o}(r, r' | \theta_M) \rceil = o(r, r')}{|S|}, \tag{4.10}$$

où $\hat{o}(r, r'|\theta_M) \in [0, 1]$ est une estimation de l'ordre binaire de vérité terrain $o(r, r')$ prédi pour l'entrée (r, r') par le modèle M.

6. 1000 mots correspondent approximativement à un discours de 8 à 10 minutes pour une présentation à débit de parole moyen, avec diapositives.

Le tableau 4.2 résume les résultats obtenus sur les différentes tâches.

Dans les deux premières tâches, on utilise des informations contextuelles sur la série pendant l'apprentissage ; lorsque le modèle TANNest utilisé pour prédire l'ordre de consommation d'une paire partiellement vue dans l'ensemble d'entraînement (X_{TS}), il obtient une moyenne de 80% de bonnes prédictions et surpasse le modèle de référence. Enfin, le modèle TANN et sa variante TANN+Att surpassent le modèle de référence pour la prédiction sur des paires provenant d'une série complètement nouvelle. En particulier, les séries proviennent de nouveaux enseignants et souvent d'un domaine non représenté dans le jeu de données d'entraînement. Malgré cela, le modèle obtient une précision de 69%. Ce résultat nous rend confiant sur la capacité du modèle à extrapoler un ordre cohérent dans un cadre général.

Nous remarquons également un écart important de performance entre les différentes itérations de la validation croisée et nous observons empiriquement que les meilleurs résultats de test sont également les meilleurs résultats de validation : l'initialisation aléatoire des poids est le facteur majeur pour expliquer cette variation. Comme suggéré dans la littérature des systèmes de recommandation (Zhao et al., 2019), il est possible de ne garder que les meilleurs modèles sur la validation pour une exploitation finale du modèle. Enfin, nous observons, comme discuté au début de la section 4.5, que l'extension classique du RNN, le mécanisme d'attention (TANN + Att), obtient une performance similaire au TANN tout en augmentant le nombre de paramètres. Pour cette raison, cette extension semble être moins pertinente dans le contexte de notre problème. Néanmoins, ce résultat reste à confirmer en présence d'un plus grand ensemble de données.

Tâche :	Avec information contextuelle						Agnostique		
Modèle :	X_{TS}			X_S^2			X_{SS}		
	Moy	Max	Min	Moy	Max	Min	Moy	Max	Min
Référence	0.76	0.82	0.69	**0.74**	0.79	0.66	0.63	0.70	0.54
TANN	**0.80**	0.84	0.76	0.71	0.80	0.65	**0.69**	0.77	0.61
TANN + Att	0.78	0.81	0.67	0.72	0.80	0.62	0.67	0.71	0.63

TABLE 4.2 – Précision de la validation croisée (10 groupes) pour les tâches au niveau des épisodes, des paires et des séries. Les écart-types observés sont de l'ordre de 0.05 points pour les modèles TANN et de l'ordre de 0.02 points pour les modèles de références.

Le tableau 4.3 résume les temps d'apprentissage des différents modèles. Dû au faible nombre de paramètres et aux petits ensembles de données, les modèles sont rapides à entraîner ($< 3h$) même sur des machines non spécialisées (sans GPU). Le modèle de ré-

férence ne considérant pas des données séquentielles est nettement plus rapide que les modèles TANN. L'ajout de l'attention ne rallonge pas significativement le temps d'entraînement des modèles, cela est en partie dû au fait que les séquences sont relativement courtes, de l'ordre de la vingtaine ($\approx 2h$ pour les conférences les plus longues).

	Temps d'entraînements (en secondes)			
Modèle :	Moy	Std	Max	Min
Référence	148	15	166	124
TANN	8157	516	8697	6311
TANN + Att	8317	846	8883	6335

TABLE 4.3 – Temps total d'apprentissage en validation croisée (10 groupes).

Les figures 4.7 à 4.10 reportent les valeurs de précision et d'erreur quadratique moyenne au cours de l'entrainement pour une architecture TANN. Les courbes d'apprentissages pour l'approche de référence et l'architecture TANN+att sont reportées en Annexe 4.8.

À l'époque zéro, les modèles commencent avec des résultats comparables à une approche aléatoire. C'est-à-dire une précision de 0.5, attendue sur une tâche de classification binaire. Au fur et en mesure de l'entraînement, les résultats s'améliorent sur les différents ensembles (augmentation de la précision, baisse de l'erreur quadratique moyenne). À partir, d'une certaine époque (autour de la 20ème époque dans notre cas) on observe un phénomène de sur-apprentissage : les résultats continuent à progresser sur l'ensemble d'entraînement, mais stagnent ou régressent sur les ensembles de validations. Autour de la 60ème époque, on observe que le modèle a complètement sur-appris les données avec une erreur quasi-nulle sur l'ensemble d'entrainement, complètement décorrélée de l'erreur sur l'ensemble de validation. Les écart-types observés sont logiquement plus grands sur les plus petits ensembles (en particulier X_V^2), mais restent relativement constants à partir de la 20ème époque sur les ensembles de validation. Le comportement décrit est conforme à une situation d'apprentissage classique.

Le comportement observé dans le cas du modèle TANN se retrouve pour les modèles de référence et TANN+att (voir Annexe 4.8).

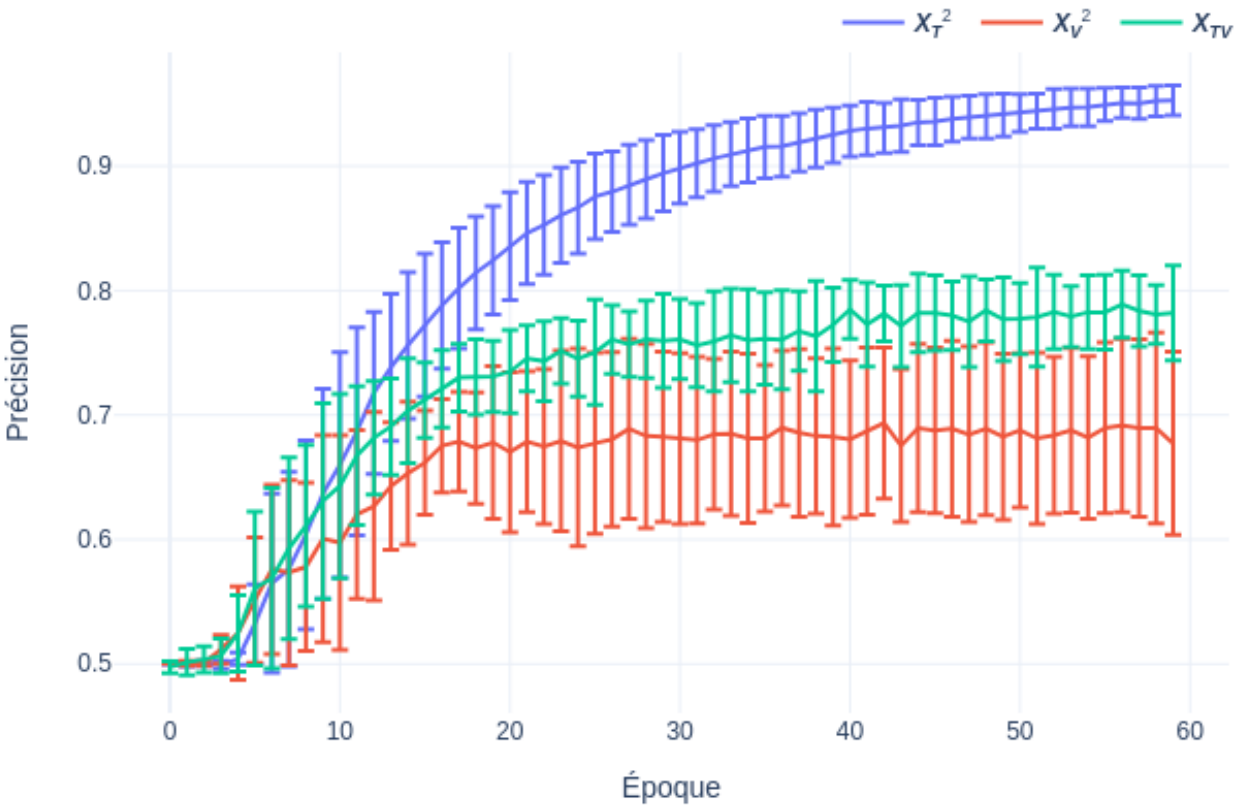


FIGURE 4.7 – Évolution de la précision durant l'entraînement sur les tâches 1 et 2 pour un modèle TANN.

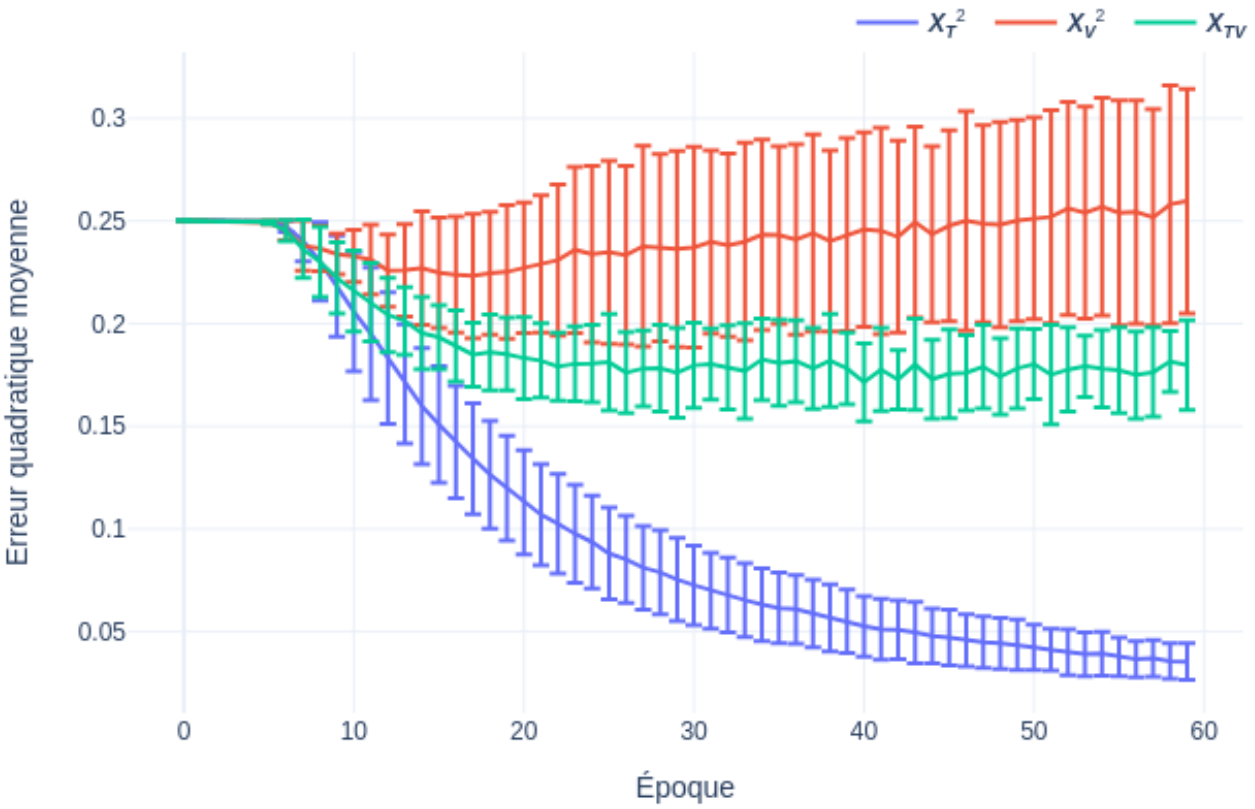


FIGURE 4.8 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur les tâches 1 et 2 pour un modèle TANN.

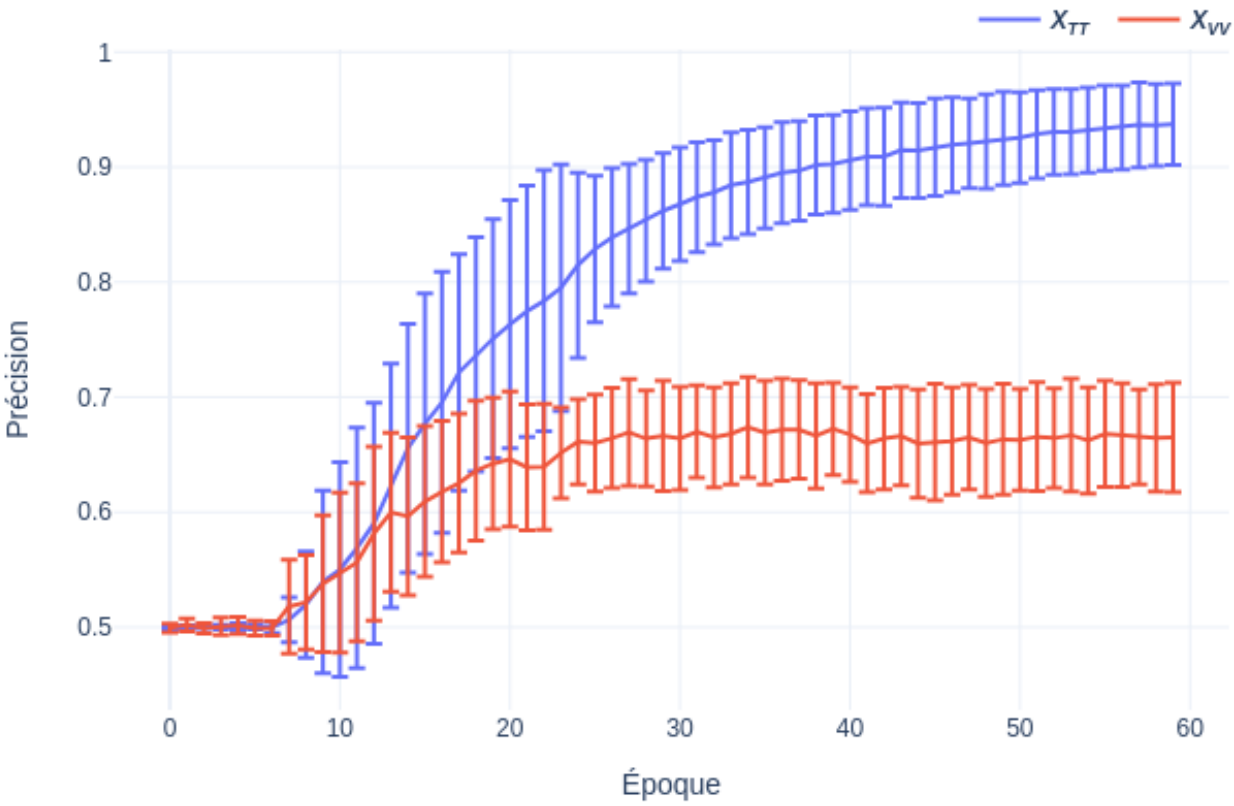


FIGURE 4.9 – Évolution de la précision durant l'entraînement sur la tâche 3 pour un modèle TANN.

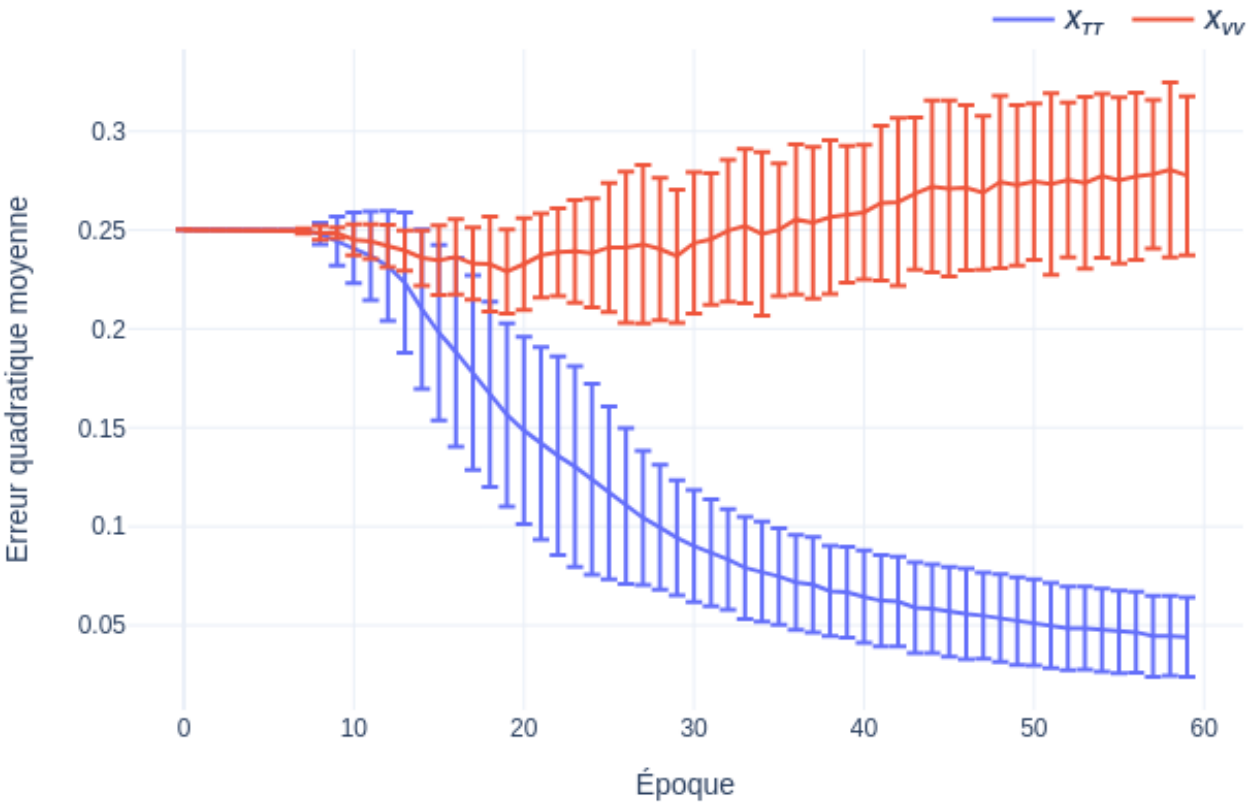


FIGURE 4.10 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur la tâche 3 pour un modèle TANN.

4.7 Discussion

Dans ce travail, nous avons choisi de construire un modèle générique capable de prédire quel que soit le domaine au lieu de construire un modèle spécifique par domaine. Ce choix est motivé par deux raisons principales. La première est que dans le cas du réseau mondial de REL, la catégorisation des ressources en domaines est rendue complexe par la nature de la construction de ce réseau. En effet, la diversité des origines, des domaines, des formats et des langues ne permet pas habituellement et systématiquement d'obtenir un étiquetage des ressources par domaine à travers les méta-données. Bien sûr, il existe de nombreux algorithmes de catégorisation qui pourraient remédier à cette carence (Devlin et al., 2019; Radford et al., 2019). Cependant, l'évaluation de leurs catégorisations dans un contexte tel que celui du réseau mondial des REL est une question de recherche en soi, qui dépasse le cadre de cette contribution. La deuxième raison est beaucoup plus pragmatique. À partir de notre expérimentation sur le corpus YaleOpenCourseware, nous avons observé que le modèle générique donne des résultats équivalents aux modèles spécifiques, voire bien meilleurs dans le cas de domaines faiblement dotés. Pour cette raison, nous pensons que l'utilisation d'un modèle générique permet un transfert d'apprentissage entre domaines. Ainsi, certaines informations sémantiques permettant d'ordonnancer les ressources seraient indépendantes du domaine ou du locuteur. Cependant, à ce stade, il est difficile de conclure, car le manque de ressources pour des domaines spécifiques ne nous permet pas d'évaluer correctement ce transfert d'apprentissage. Des expériences approfondies, en particulier sur des jeux de données émanant d'initiatives d'annotation participative, pour confirmer ce résultat constituent une perspective de recherche intéressante qui nécessite la confection d'un jeu de données plus diversifié et important.

De plus, le contexte général est celui d'un démarrage à froid : il n'y a pas assez d'éléments de données utilisateur pour (i) construire une solution basée sur l'activité de l'utilisateur, et (ii) utiliser l'activité de l'utilisateur pour effectuer des tests A/B.

Le but ultime de la recommandation pédagogique serait de construire un parcours d'apprentissage pour un utilisateur donné dans une variété de situations : ses objectifs peuvent être clairs ou non, le matériel qu'il souhaite inclure dans son expérience d'apprentissage peut être complètement ou partiellement inconnu. Ce parcours d'apprentissage devrait donc prendre en compte de nombreuses facettes du problème d'apprentissage. Une technologie capable de classer les ressources ne donnera pas, à elle seule, la réponse à cet objectif plus ambitieux, mais nous pensons que c'est un prérequis nécessaire pour réussir

dans cette tâche. Dans le projet X5GON, nous avons abordé les questions de difficulté, d'aléatoire, de personnalisation. Et les idées présentées dans ce travail ont déjà conduit à une implémentation sur la plateforme `x5learn.org` : un utilisateur est encouragé à parcourir les collections de REL et à construire une liste de lecture, puis à demander que les éléments de cette liste de lecture soient réorganisés de manière plus complète.

4.8 Conclusion

Pour conclure, nous avons proposé dans ce travail un nouveau cadre pour aborder le problème de la recommandation pédagogique en nous concentrant sur la nécessité d'une continuité satisfaisante dans le parcours d'apprentissage. Pour cela, nous avons conçu une nouvelle tâche d'apprentissage comme *proxy* pour ce problème qui tente de tirer profit de la continuité pédagogique implicite intégrée dans les séries de cours construites par l'enseignant. Nous avons ensuite fourni un nouvel ensemble de données libre et ouvert pour la recommandation pédagogique et évalué la nouvelle approche basée sur un réseau de neurones préservant la chronologie que nous avons spécialement conçue pour cette tâche par rapport à une base de référence TALN de pointe. L'évaluation démontre que notre modèle surpasse la ligne de base, en particulier lorsque nous lui demandons de prédire des paires de ressources complètement nouvelles, même en dehors de tout contexte vu pendant l'apprentissage. Dans ce contexte agnostique, une précision de 69% a été obtenue.

Nous espérons à l'avenir remplacer les outils actuels permettant de classer les REL dans les plateformes X5GON par des modèles construits à la suite de ce travail, et l'évaluer avec des tests A/B et une analyse de l'expérience utilisateur.

Conclusion

Dans ce manuscrit, nous nous sommes intéressés à la recommandation à visée pédagogique dans un cadre d'apprentissage non formel sur des données non structurées. Cette problématique s'inscrit dans un contexte stimulant d'expansion des pratiques d'apprentissage en ligne et dans une philosophie de l'éducation ouverte prônant une culture ouverte et une éducation pour tous (objectif de développement durable numéro 4 de l'UNESCO). L'objet au cœur de cette philosophie est la REL (Ressource Educatives Libres) ; cet acronyme désigne tout objet éducatif ouvert librement distribuable et réutilisable (Wiley, 2014). Par bien des aspects, cet objet révolutionne l'éducation et rend l'apprentissage plus accessible à tous (Clinton and Khan, 2019). Néanmoins, un des problèmes concerne la difficulté ressentie d'accès aux ressources, et en particulier pour les ressources provenant de petits acteurs et pour les ressources en langues peu dotées (Wang and Towey, 2017). Dans ce contexte, le projet X5gon a été lancé en 2018, il a eu pour but d'indexer et de connecter les REL dans un réseau unique et ouvert. Avec cet objectif X5gon souhaitait d'une part stimuler la recherche en fournissant des jeux de données de REL et des jeux de données de traces d'apprentissage, dont la précédente absence avait maintes fois été observée comme un frein aux avancées du domaine (Urdaneta-Ponte et al., 2021; Hernández-Blanco et al., 2019). D'autre part, il s'agissait de connecter les REL grâce en particulier à des moteurs de recherche de REL et des systèmes de recommandation de REL. Cette deuxième tâche fait donc l'objet de cette thèse et soulève une question de recherche particulière : comment recommander à large échelle dans le but de fournir la meilleure expérience d'apprentissage possible à l'apprenant, et ce, dans un contexte dans lequel les données sont non structurées et la situation d'apprentissage est non formelle ?

Pour répondre à cette problématique, les premières contributions que nous avons réalisées concernent la mise à disposition de jeux de données (Chapitre 2). Le premier, directement dérivé de notre travail dans le projet X5gon, se présente sous la forme d'une API à travers laquelle il est possible de récupérer le contenu, la transcription, les métadonnées ainsi qu'un ensemble de modèles dérivés de plus de 118 000 REL dans 8 différentes langues et provenant de 22 sites différents. Ce corpus est - à notre connaissance - le plus gros corpus de REL actuellement disponible ; de plus, il propose une diversité intéressante

en termes de langues et de variété qui fait souvent défaut à ce type de corpus.

Une des faiblesses observées de ce corpus concerne la faible qualité des méta-données et en particulier l'absence de structuration interne entre les ressources qui témoigneraient par exemple de liens de séquentialité ou de relations notions-exercices. Pour palier à cette absence, nous avons construit le corpus YaleOpenCourseware grâce à un travail d'extraction des données depuis le site `https://oyc.yale.edu/`. À la différence du corpus, X5GON ce jeu de données ne se démarque pas par sa taille (1058 REL) ou sa diversité (langue anglaise uniquement et cours de l'université de Yale malgré un large éventail de 15 domaines) mais par la richesse de ses méta-données. En particulier, le corpus est structuré en une liste de cours, eux-mêmes décomposés en séries de conférences. Cette relation de séquentialité entre les conférences offre des perspectives pour pouvoir évaluer les modèles d'ordonnancement pour des systèmes de recommandation à visée pédagogique qui sont d'une importance critique pour la tâche de recommandation (Granka et al., 2004).

En-dehors des jeux des données récoltées, le manuscrit s'articule autour de deux contributions principales :

1. la mise au point d'une méthode de modélisation des données séquentielles sous-forme d'automate temporisé et probabiliste. Nous avons développé un algorithme d'élagage de l'automate permettant de garantir la confidentialité des données. L'efficacité de l'approche est démontrée sur des données spécifiques provenant des traces d'apprentissage recueillies par X5GON (Chapitre 3).
2. la création de nouvelles tâches d'évaluation des modèles d'ordonnancement destinés à de la recommandation à visée pédagogique. Nous avons aussi développé un algorithme d'apprentissage permettant d'apprendre un ordre cohérent de consommation des ressources en s'appuyant sur des séries de ressources ordonnées par des experts, en l'occurrence les séries du corpus YaleOpenCourseware (Chapitre 4).

Concernant la première contribution, la méthode se base sur un nouveau type d'automates à états finis, à la fois temporisés et probabilistes (PTk-TA) introduits pour l'occasion. L'automate est appris depuis les données séquentielles grâce à une adaptation des techniques d'apprentissage d'automates k-testables incluant les durées et les probabilités. Enfin, l'automate est élagué afin de garantir la confidentialité : l'idée principale issue du domaine de la confidentialité différentielle est qu'aucune séquence des données ne doit avoir une importance trop forte sur la distribution des données générées. Le cas d'étude sur les données issues des traces d'apprentissage collectées durant le projet X5GON permet de conclure que même dans le cas d'un univers de symboles très large (ici les REL),

la méthode conserve les motifs court les plus fréquents dans l'ensemble de données. Ce dernier résultat est particulièrement intéressant pour le cas de la recommandation dans laquelle les jeux de données contiennent un univers de symboles très large et l'historique à court terme joue un rôle prépondérant.

Concernant la deuxième contribution, nous avons défini trois tâches de catégorisation binaire des ressources de difficulté croissante. Dans chaque tâche, il est question de prédire correctement l'ordre d'une paire de ressources sur la base de l'ordre observé dans une série réelle. Dans la première tâche (tâche 1), les ressources proviennent d'une série vue durant l'entraînement : en particulier, une et une seule des deux ressources de chaque paire est présente dans l'ensemble d'entraînement. Dans la deuxième tâche (tâche 2), les ressources proviennent toujours d'une série vue durant l'entraînement. Néanmoins, aucune des deux ressources constituant les paires de tests n'a été vue durant l'entraînement. Enfin, dans la troisième tâche (tâche 3), les ressources proviennent d'une nouvelle série complètement étrangère des données d'entraînement. Le modèle TANN que nous avons introduit est un modèle neuronal profond exposant une architecture siamoise récurrente. Le modèle prend en entrée deux REL et cherche à prédire un ordre de consommation cohérent pour ces REL. Contrairement, aux approches *état de l'art*, TANN cherche à exploiter la construction chronologique du discours pour inférer l'ordre. Pour ce faire, il utilise une représentation sémantique des ressources préservant la chronologie (*continuous-Doc2vec*). Les résultats sur le corpus YaleOpenCourseware montrent que le modèle TANN performe mieux qu'un modèle état de l'art sur les tâches 1 et 3. En particulier pour la tâche 3, même en présence d'une paire de ressources complètement nouvelle avec un locuteur et un domaine différent des données d'apprentissage, le modèle TANN reste capable d'inférer correctement l'ordre de 69% des paires de ressources. Ce résultat témoigne du fait que certaines informations sémantiques permettant d'ordonnancer les ressources seraient indépendantes du domaine ou du locuteur. Enfin, il permet d'être optimiste sur les possibilités d'application du modèle dans un contexte d'apprentissage multidomaine et multilocuteur. En particulier, le modèle développé est applicable en l'état comme modèle d'ordonnancement pour la tâche de recommandation pédagogique qui nous intéresse. Plus largement, cette contribution ouvre la voie aux développements de modèles d'ordonnancements spécifiques au cas de la recommandation à visée pédagogique en fournissant un cadre d'apprentissage et d'évaluation s'adaptant aux spécificités de ce contexte.

Durant ces dernières années, l'apprentissage en ligne est devenue une question de société de plus en plus importante. Cela est en grande partie dû au contexte spécifique

engendré par la pandémie de la COVID-19, mais l'intérêt pour la question ne semble pas être retombé. L'éducation représente une manne financière dans laquelle nombre d'acteurs privés sont impliqués. La marchandisation de l'éducation en ligne ne favorise pas toujours une égalité d'accès à l'apprentissage au cœur du mouvement de l'éducation ouverte. La philosophie portée par le projet X5GON était de créer une émulation dans les domaines de l'éducation et de la recherche pour les questions en lien avec l'éducation ouverte et en particulier l'apprentissage en ligne. L'ensemble des contributions que nous avons réalisées s'inscrit dans cette filiation. Aujourd'hui, les acteurs publics et en première ligne, l'UNESCO pousse dans le sens de l'éducation ouverte et font des REL un objet central de cette politique. En considérant la masse d'apprenants et de ressources à traiter, les méthodes automatiques et en particulier les méthodes basées sur l'apprentissage profond semblent les plus à même pour indexer, connecter et recommander les REL. Ces méthodes présentent deux lacunes majeures : premièrement, elles nécessitent un grand nombre d'exemples d'apprentissage pour être entraînées. Deuxièmement, elles fournissent des modèles difficilement interprétables susceptibles de conserver, voire d'exacerber, les biais des données. Nous pensons que la meilleure manière de combler ces lacunes est de fournir des jeux de données offrant la plus grande variété possible.

Dans un futur proche, notre premier objectif est donc d'améliorer la qualité et la richesse des corpus de données, en particulier à travers le projet Florilège et grâce à de l'annotation participative. Nous espérons pouvoir créer un corpus de REL francophones riche en méta-données. Ce corpus plus conséquent pourrait venir confirmer les résultats observés sur le corpus YaleOpenCoursewarehttps ://www.univ-nantes.fr/etudier-se-former/decouvrir-nos-formations/soutenir-son-doctorat et permettre d'explorer plus profondément les performances de l'approche TANN ou d'éventuelles approches concurrentes sur des aspects tels que l'importance du domaine, du niveau, de la langue ou du public cible. En outre, il pourrait également permettre d'explorer de nouvelles tâches comme la détection automatique du niveau d'une REL, de son type ou encore du public cible. Mentionnons aussi la détection automatique de REL qui permettrait à plus long terme d'obtenir une indexation automatique et non plus semi-automatique des REL corrigées par un système d'annotation participative.

Dans la continuité du projet X5GON nous espérons pérenniser le système d'indexation de REL et permettre son amélioration et son enrichissement. De plus, grâce aux plates-formes développées durant le projet, nous espérons recueillir suffisamment de traces d'apprentissage des utilisateurs. Cela permettra à terme de créer des jeux de données publics

et anonymisés de traces d'apprentissage en tirant profit de méthodes d'anonymisation comme celle présentée au Chapitre 3.

Annexes

Méthode d'anonymisation de traces d'apprentissage

Résultats additionnels

			compte	min	max	moy	écart	25%	50%	75%	90%
Données brutes			100236	<1s	18h57m	1h21m	1h57m	6m	30m	1h43m	4h3m
(RD)	k=1	ϵ=0.02	21749	>1s	13h56m	58m58s	1h33m	4m19s	21m	1h8m	2h55m
		ϵ=0.05	69253	>1s	16h32m	1h13m	1h49m	5m51s	27m	1h30m	3h38m
		ϵ=0.1	94169	>1s	17h1m	1h20m	1h56m	6m41s	30m	1h42m	4h
		ϵ=0.2	98495	<1s	17h1m	1h20m	1h57m	6m43s	30m	1h42m	4h1m
		ϵ=0.25	98721	<1s	17h1m	1h20m	1h57m	6m44s	30m	1h43m	4h1m
	k=2	ϵ=0.05	113	1s	12h15m	1h49m	2h11m	21m23s	1h4m	2h40m	4h22m
		ϵ=0.1	444	1s	12h15m	1h10m	1h44m	4m2s	26m25s	1h31m	3h36m
		ϵ=0.2	3553	<1s	12h15m	40m	1h21m	1m30s	9m52s	37m	1h53m
		ϵ=0.25	7413	<1s	12h15m	35m	1h11m	1m32s	9m12s	33m	1h38m
	k=3	ϵ=0.05	14	4s	5m3s	1m2s	1m35s	8s	17s	49s	3m29s
		ϵ=0.1	95	4s	5h29m	20m34s	53m3s	16s	2m27s	12m29s	46m7s
		ϵ=0.2	820	2s	11h55m	36m	1h26m	42s	5m55s	27m	1h35m
		ϵ=0.25	1551	2s	11h55m	33m	1h17m	41s	5m7s	25m	1h31m
	k=4	ϵ=0.05	14	4s	5m3s	1m2s	1m35s	8s	17s	49s	3m29s
		ϵ=0.1	82	4s	5h12m	19m6s	45m12s	16s	2m22s	13m25s	46m25s
		ϵ=0.2	643	2s	11h55m	28m	1h19m	34s	3m53s	19m	1h38s
		ϵ=0.25	1119	<1s	11h55m	23m	1h7m	29s	2m51s	15m	50m
	k=5	ϵ=0.05	14	4s	5m3s	1m2s	1m35s	8s	17s	49s	3m29s
		ϵ=0.1	82	4s	5h12m	19m6s	45m12s	16s	2m22s	13m25s	46m25s
		ϵ=0.2	640	2s	11h55m	27m1s	1h16m	33s	3m44s	18m51s	56m8s
		ϵ=0.25	1127	>1s	11h55m	23m38s	1h8m	29s	2m54s	15m11s	49m48s
(GD)	k=1	AE	100000	<1s	>2d	5h12m	5h28m	1h18m	3h31m	7h18m	12h22m
		ϵ=0.02	10000	<1s	1d19h	4h20m	4h36m	1h4m	2h53m	6h6m	10h16m
		ϵ=0.05	10000	<1s	2d17h	4h50m	5h7m	1h15m	3h15m	6h47m	11h22m
		ϵ=0.1	100000	<1s	2d2h	5h12m	5h25m	1h19m	3h30m	7h23m	12h19m
		ϵ=0.2	100000	<1s	>2d	5h12m	5h28m	1h18m	3h30m	7h19m	12h20m
		ϵ=0.25	100000	<1s	>2d	5d12h	5h27m	1h10m	3h31m	7h19m	12h17m
	k=2	AE	100000	<1s	>2d	3h	3h27m	35m	1h48m	4h11m	7h27m
		ϵ=0.05	10000	<1s	1d11h	3h26m	3h53m	35m1s	2h12m	4h58m	8h34m
		ϵ=0.1	10000	<1s	2d1h	2h27m	3h33m	14m31s	1h1m	3h16m	6h51m
		ϵ=0.2	100000	<1s	>3d	1h34m	2h27m	17m	50m	1h53m	3h38m
		ϵ=0.25	100000	<1s	>2d	1h25m	1h58m	16m	50m	1h50m	3h21m
	k=3	AE	10000	<1s	1d6h	2h19m	2h38m	29m49s	1h23m	3h14m	5h46m
		ϵ=0.05	10000	16s	32m15s	9m36s	4m45s	6m2s	8m59s	12m32s	16m8s
		ϵ=0.1	10000	<1s	1d1h	30m57s	1h17m	7m52s	13m49s	24m28s	50m33s
		ϵ=0.2	100000	<1s	>3d	1h7m	2h15m	13m	27m	57m	2h37m
		ϵ=0.25	100000	<1s	>2d	1h5m	2h3m	14m	29m	1h1m	2h25m
	k=4	AE	10000	<1s	1d1h	2h1m	2h23m	24m56s	1h10m	2h45m	5h5m
		ϵ=0.05	10000	18s	38m33s	11m16s	5m40s	7m0s	10m33s	14m42s	19m2s
		ϵ=0.1	10000	27s	5h36m	27m1s	34m34s	9m58s	16m21s	27m58s	52m4s
		ϵ=0.2	100000	3s	>2d	49m	1h50m	11m	21m	43m	1h31m
		ϵ=0.25	100000	<1s	>3d	42m	1h27m	11m	21m	42m	1h18m
	k=5	AE	10000	<1s	22h33m	1h48m	2h7m	21m52s	1h2m	2h27m	4h38m
		ϵ=0.05	10000	25s	39m18s	11m9s	5m40s	6m51s	10m23s	14m38s	18m47s
		ϵ=0.1	10000	10s	5h14m	25m28s	33m55s	9m12s	15m19s	25m57s	46m43s
		ϵ=0.2	10000	6s	1d20h	46m27s	1h37m	10m54s	21m8s	42m50s	1h29m
		ϵ=0.25	10000	<1s	1d3h	41m1s	1h16m	9m58s	20m52s	43m12s	1h21m

TABLE 4.4 – Distribution de durées des sessions pour les données brutes, non-élaguées et générées.

			count	min	max	moy	écart	25%	50%	75%	90%
Données brutes			100236	4	2609	21.31	31.81	6	12	23	46
(RD)	k=1	ϵ=0.02	21749	4	359	18.64	22.279	6	11	22	40
		ϵ=0.05	69253	4	629	19.18	24.74	6	11	22	41
		ϵ=0.1	94169	4	1513	20.69	28.69	6	11	23	45
		ϵ=0.2	98495	4	1513	21.12	30.36	6	12	23	46
		ϵ=0.25	98721	4	1513	21.16	30.47	6	12	23	46
	k=2	ϵ=0.05	113	4	27	6.67	3.76	4	6	8	10
		ϵ=0.1	444	4	41	6.76	4.56	4	5	7	11.7
		ϵ=0.2	3553	4	89	8.46	6.81	5	6	10	15
		ϵ=0.25	7413	4	124	9.12	7.8	5	6	10	17
	k=3	ϵ=0.05	14	4	4	4	0	4	4	4	4
		ϵ=0.1	95	4	8	4.77	0.98	4	4	5	6
		ϵ=0.2	820	4	61	5.91	4.77	4	5	6	8
		ϵ=0.25	1551	4	61	6.09	4.20	4	5	6	9
	k=4	ϵ=0.05	14	4	4	4	0	4	4	4	4
		ϵ=0.1	82	4	6	4.56	0.75	4	4	5	6
		ϵ=0.2	643	4	61	5.75	4.84	4	5	6	8
		ϵ=0.25	1119	4	61	5.66	4.02	4	5	6	8
	k=5	ϵ=0.05	14	4	4	4	0	4	4	4	4
		ϵ=0.1	82	4	6	4.56	0.75	4	4	5	6
		ϵ=0.2	640	4	61	5.75	4.85	4	5	6	8
		ϵ=0.25	1127	4	61	5.68	4.02	4	5	6	8
(GD)	k=1	AE	100000	1	207	22.43	21.69	7	16	31	51
		ϵ=0.02	10000	1	171	19.55	19.23	6	14	27	44
		ϵ=0.05	10000	1	246	20.27	19.90	6	14	28	46
		ϵ=0.1	10000	1	215	21.82	21.15	7	15	30	50
		ϵ=0.2	100000	1	215	21.81	21.51	7	15	30	50
		ϵ=0.25	100000	1	192	21.92	21.31	7	15	30	49
	k=2	AE	100000	1	191	21.72	21.49	7	15	30	50
		ϵ=0.05	10000	1	56	6.73	6.21	2	5	9	15
		ϵ=0.1	10000	1	68	6.78	6.23	3	5	8	15
		ϵ=0.2	100000	1	82	8.42	7.61	4	6	11	18
		ϵ=0.25	100000	1	112	9.16	8.47	4	7	12	20
	k=3	AE	100000	2	217	21.39	20.43	7	15	29	48
		ϵ=0.05	10000	4	4	4	0	4	4	4	4
		ϵ=0.1	10000	2	32	4.77	1.82	4	4	5	6
		ϵ=0.2	100000	2	61	5.91	4.75	4	5	6	9
		ϵ=0.25	100000	2	61	6.08	4.26	4	5	7	10
	k=4	AE	100000	3	180	21.22	19.71	7	15	28	46
		ϵ=0.05	10000	4	4	4	0	4	4	4	4
		ϵ=0.1	10000	4	6	4.56	0.75	4	4	5	6
		ϵ= 0.2	100000	3	61	5.82	4.94	4	5	6	8
		ϵ= 0.25	100000	3	61	5.67	4.05	4	5	6	8
	k=5	AE	10000	4	227	21.31	19.90	7	15	28	47
		ϵ=0.05	10000	4	4	4	0	4	4	4	4
		ϵ=0.1	10000	4	6	4.55	0.75	4	4	5	6
		ϵ=0.2	10000	4	61	5.71	4.31	4	5	6	8
		ϵ=0.25	10000	4	61	5.69	3.71	4	5	6	8

TABLE 4.5 – Distribution de longueurs des sessions pour les données brutes, non-élaguées et générées.

			$\mathcal{N}$=20	$\mathcal{N}$=40	$\mathcal{N}$=60	$\mathcal{N}$=80	$\mathcal{N}$=100
Données générées	k=1	AE	0	0	0	0	0
		ϵ=0.02	0.05	0.05	0.10	0.11	0.09
		ϵ=0.05	0	0	0	0	0
		ϵ=0.1	0	0	0	0	0
		ϵ=0.2	0	0.02	0.01	0.01	0.01
		ϵ=0.25	0	0	0	0	0
	k=2	AE	0.65	0.62	0.68	0.66	0.61
		ϵ=0.05	0	0	0	0	0
		ϵ=0.1	0	0	0	0	0
		ϵ=0.2	0.20	0.25	0.25	0.22	0.22
		ϵ=0.25	0.40	0.30	0.31	0.28	0.28
	k=3	AE	0.25	0.45	0.48	0.48	0.51
		ϵ=0.05	0	0	0	0	0
		ϵ=0.1	0	0	0	0	0
		ϵ=0.2	0	0	0.10	0.08	0.07
		ϵ=0.25	0.25	0.22	0.28	0.25	0.25
	k=4	AE	0.25	0.32	0.43	0.50	0.49
		ϵ=0.05	0	0	0	0	0
		ϵ=0.1	0	0	0	0	0
		ϵ=0.2	0.10	0.12	0.10	0.10	0.09
		ϵ=0.25	0.10	0.12	0.11	0.10	0.10
	k=5	AE	0.55	0.70	0.71	0.72	0.74
		ϵ=0.05	0	0	0	0	0
		ϵ=0.1	0	0	0	0	0
		ϵ=0.2	0.10	0.12	0.10	0.10	0.09
		ϵ=0.25	0.10	0.12	0.11	0.10	0.10

TABLE 4.6 – Taux de vrais positifs dans $\mathcal{N}$ motifs les plus fréquents. Les valeurs sont reportés pour une taille d'échantillon des données générées de 10000, soit 10 fois non que le jeu de données réel cela peut altérer négativement les performances observés.

Prédiction d'ordre sur des séquences d'apprentissages

Résultats additionnels

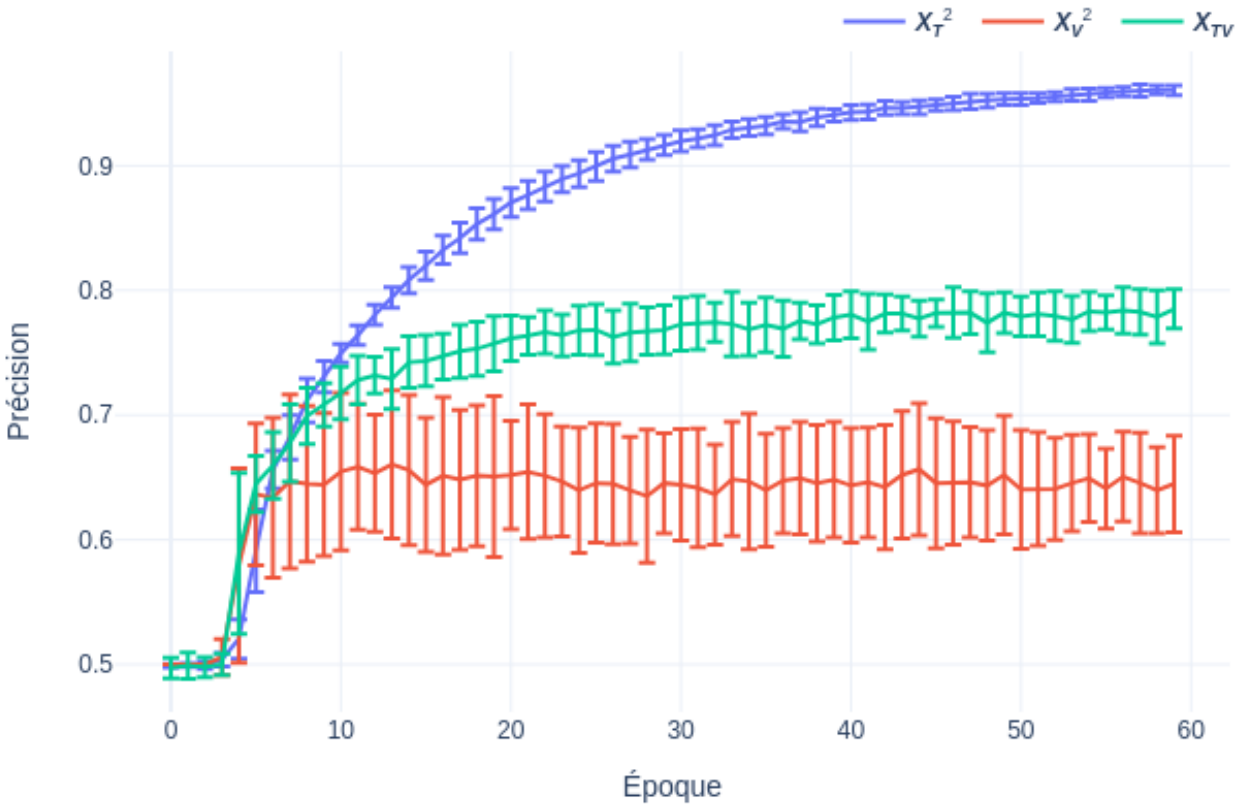


FIGURE 4.11 – Évolution de la précision durant l'entraînement sur les tâches 1 et 2 pour un modèle TANN avec attention.

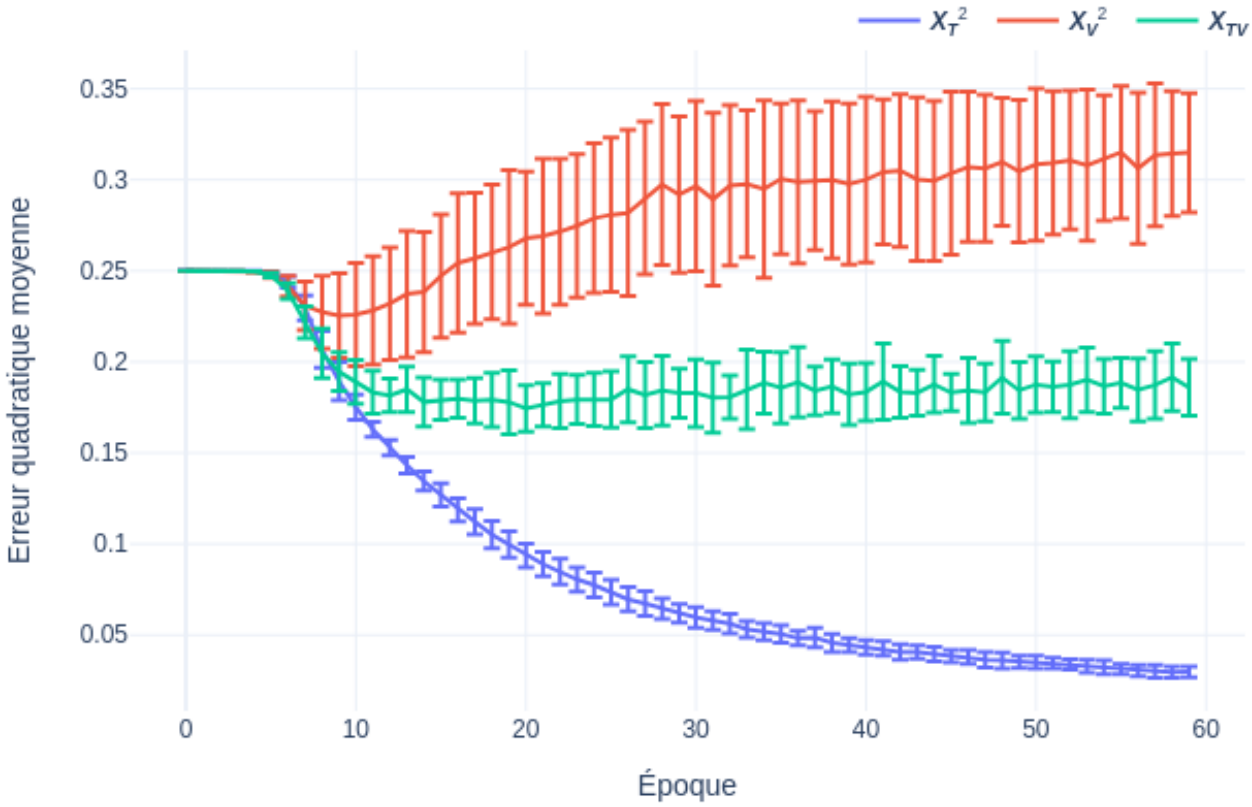


FIGURE 4.12 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur les tâches 1 et 2 pour un modèle TANN avec attention.

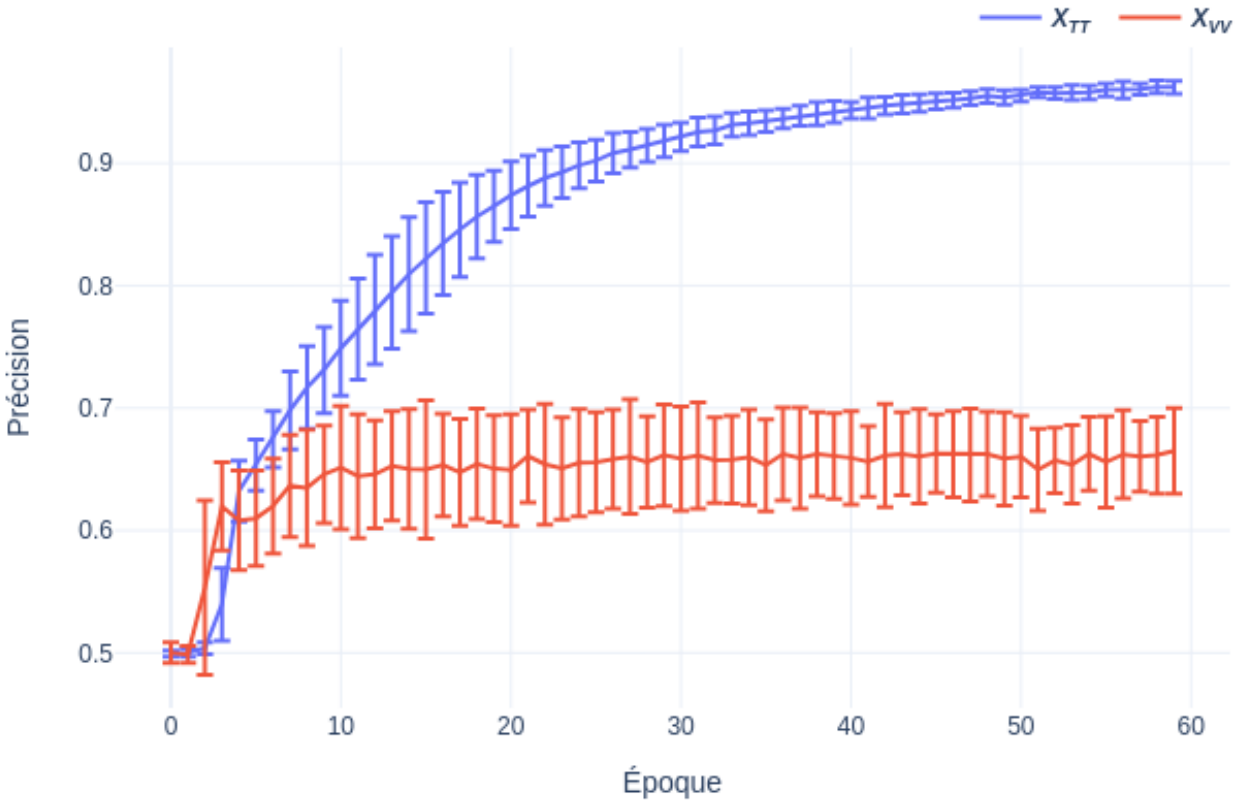


FIGURE 4.13 – Évolution de la précision durant l'entraînement sur la tâche 3 pour un modèle TANN avec attention.

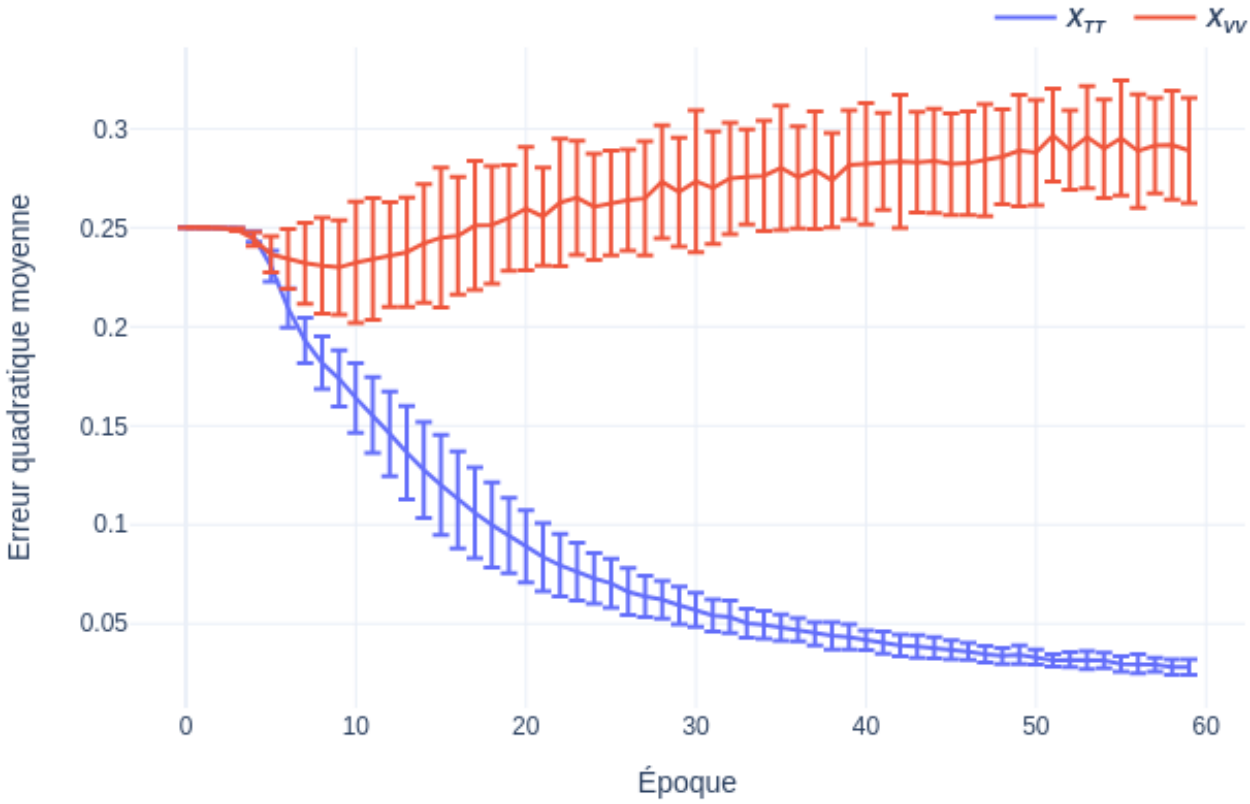


FIGURE 4.14 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur la tâche 3 pour un modèle TANN avec attention.

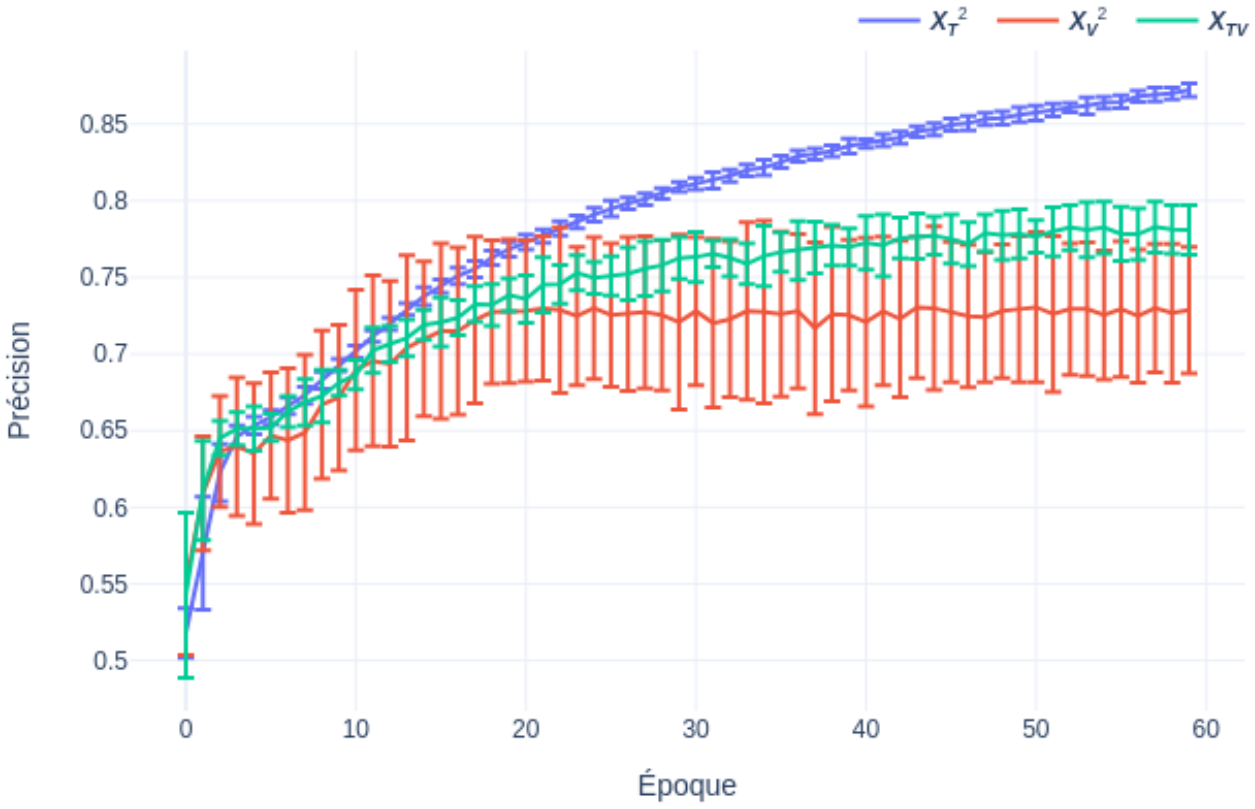


FIGURE 4.15 – Évolution de la précision durant l'entraînement sur les tâches 1 et 2 pour un modèle "référence".

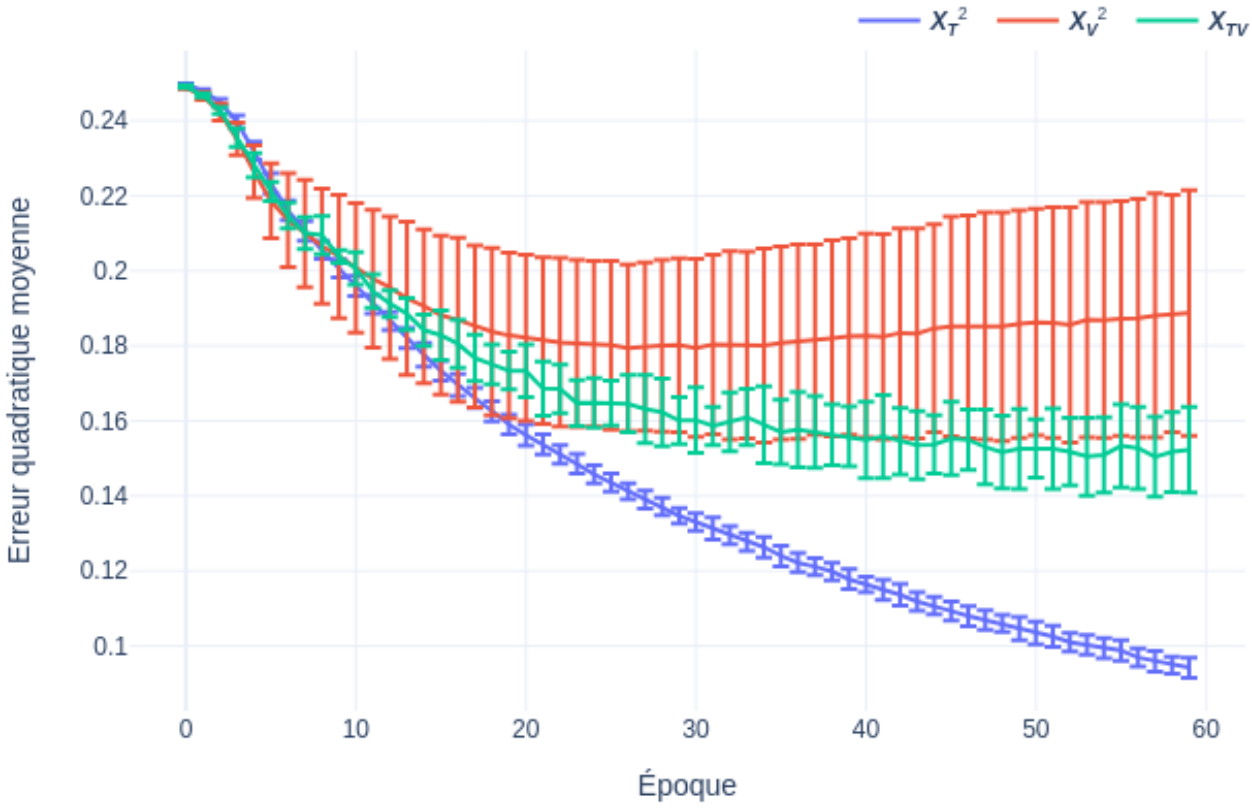


FIGURE 4.16 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur les tâches 1 et 2 pour un modèle "référence".

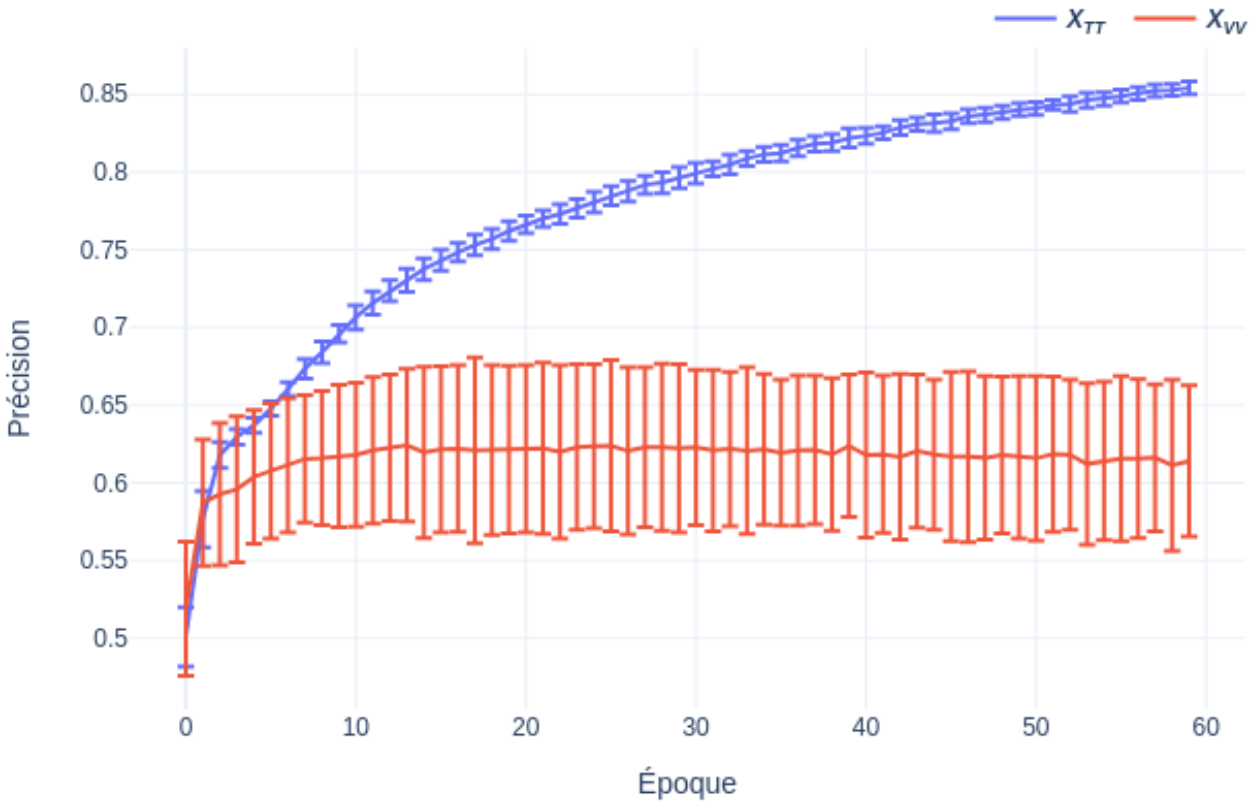


FIGURE 4.17 – Évolution de la précision durant l'entraînement sur la tâche 3 pour un modèle "référence".

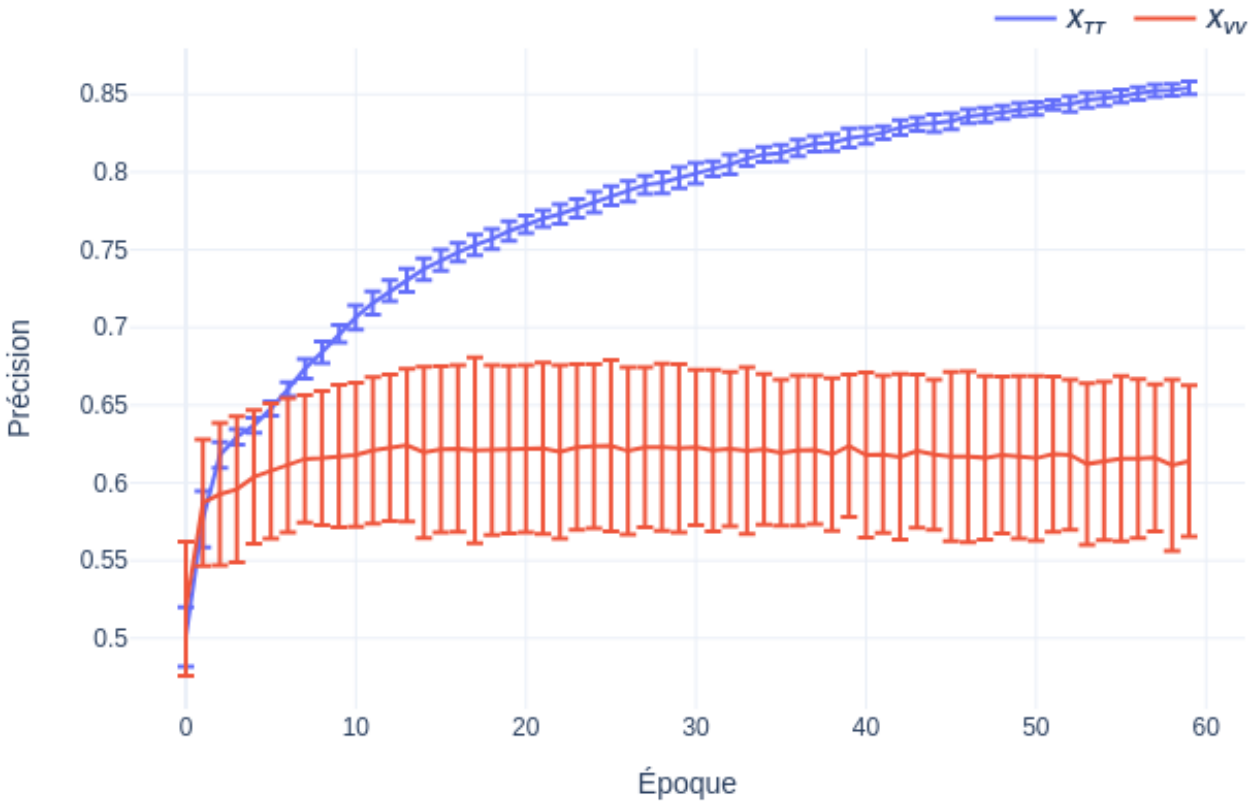


FIGURE 4.18 – Évolution de l'erreur quadratique moyenne durant l'entraînement sur la tâche 3 pour un modèle "référence".

BIBLIOGRAPHIE

Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. 2019. Privacy preserving synthetic data release using deep learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 510–526. Springer International Publishing.

Kumar Abhinav, Venkatesh Subramanian, Alpana Dubey, Padmaraj Bhat, and Aditya Divakaruni Venkat. 2018. Lecore : A framework for modeling learner's preference. In *EDM*.

Charu C Aggarwal. 2016a. Model-based collaborative filtering. In *Recommender systems*, pages 71–138. Springer.

Charu C Aggarwal. 2016b. Neighborhood-based collaborative filtering. In *Recommender systems*, pages 29–70. Springer.

Charu C Aggarwal and Srinivasan Parthasarathy. 2001. Mining massively incomplete data sets by conceptual reconstruction. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 227–232.

Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. 2007. Open user profiles for adaptive news systems : help or harm ? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20.

Fatima N Al-Aswadi, Huah Yong Chan, and Keng Hoon Gan. 2020. Automatic ontology construction from text : a review from shallow to deep learning trend. *Artificial Intelligence Review*, 53(6) :3901–3928.

Joachim Allgaier. 2019. Science and Environmental Communication via Online Video : Strategically Distorted Communications on Climate Change and Climate Engineering on YouTube. *Frontiers in Communication*, 4 :36.

Uri Alon and Eran Yahav. 2020. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.

Rajeev Alur and David L Dill. 1994. A theory of timed automata. *Theoretical computer science*, 126(2) :183–235.

Tel Amiel and Tiago Chagas Soares. 2016. Identifying tensions in the use of open licenses in oer repositories. *The International Review of Research in Open and Distributed Learning*, 17(3).

Deepa Anand and Kamal K Bharadwaj. 2011. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. *Expert systems with applications*, 38(5) :5101–5109.

Chris Anderson. 2006. *The long tail : Why the future of business is selling less of more.* Hachette UK.

Gary Anthes. 2013. Deep learning comes of age. *Communications of the ACM*, 56(6) :13–15.

Daniel Ewell Atkins, John Seely Brown, and Allen L Hammond. 2007. *A review of the open educational resources (OER) movement : Achievements, challenges, and new opportunities*, volume 164. Creative common Mountain View.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015.*

Marko Balabanović and Yoav Shoham. 1997. Fab : content-based, collaborative recommendation. *Communications of the ACM*, 40(3) :66–72.

Sudipto Banerjee and Anindya Roy. 2014. *Linear algebra and matrix analysis for statistics*, volume 181. Crc Press Boca Raton, FL, USA.

Himani Bansal, Gulshan Shrivastava, Gia Nhu Nguyen, and Loredana-Mihaela Stanciu. 2018. *Social network analytics for contemporary business organizations.* IGI Global.

Ana Belén Barragáns-Martínez, Enrique Costa-Montenegro, Juan C Burguillo, Marta Rey-López, Fernando A Mikic-Fonte, and Ana Peleteiro. 2010. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22) :4290–4311.

Mustapha Baziz, Mohand Boughanem, and Nathalie Aussenac-Gilles. 2004. The use of ontology for semantic representation of documents. In *The 2nd Semantic Web and Information Retrieval Workshop (SWIR), SIGIR*, pages 38–45.

Punam Bedi, Pooja Vashisth, Purnima Khurana, et al. 2013. Modeling user preferences in a hybrid recommender system using type-2 fuzzy sets. In *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.

Joeran Beel, Marcel Genzmehr, Stefan Langer, Andreas Nürnberger, and Bela Gipp. 2013a. A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In *Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation*, pages 7–14.

Joeran Beel, Stefan Langer, Andreas Nürnberger, and Marcel Genzmehr. 2013b. The impact of demographics (age and gender) and other user-characteristics on evaluating recommender systems. In *International Conference on Theory and Practice of Digital Libraries*, pages 396–400. Springer.

Olga Maria Belikov and Robert Bodily. 2016. Incentives and barriers to oer adoption: A qualitative analysis of faculty perceptions. *Open Praxis*, 8(3) :235–246.

Robert Bell, Yehuda Koren, and Chris Volinsky. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104.

Robert M Bell and Yehuda Koren. 2007a. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2) :75–79.

Robert M Bell and Yehuda Koren. 2007b. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 43–52. IEEE.

Richard Bellman. 1966. Dynamic programming. *Science*, 153(3731) :34–37.

Geert Jan Bex, Frank Neven, Thomas Schwentick, and Karl Tuyls. 2006. Inference of concise dtds from xml data. In *Proceedings of the 32nd international conference on Very large data bases*, pages 115–126.

Daniel Billsus and Michael Pazzani. 1997. Learning probabilistic user models. In *UM97 Workshop on Machine Learning for User Modeling*.

Daniel Billsus and Michael J Pazzani. 2000. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2) :147–180.

Daniel Billsus, Michael J Pazzani, and James Chen. 2000. A learning agent for wireless news access. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 33–36.

Antoine Blanchard, Stéphane Debove, Pleen Le Jeune, David Louapre, and Tania Louis. 2018. Que sait-on des vidéastes de science sur YouTube ? `https://www.amcsti.fr/fr/bulletin/sait-on-videastes-de-science-youtube/`. Le Bulletin de l'AMCSTI.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan) :993–1022.

JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 26 :225–238.

Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5 :135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker ? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. 2016. Casmos : A framework for learning candidate selection models over structured queries and documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 441–450.

G. Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of GSCL*, volume 30.

Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant Wikipedia concepts. *Proceedings of SiKDD*.

Matthias Braunhofer, Mehdi Elahi, Mouzhi Ge, and Francesco Ricci. 2014. Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In *International Conference on Learning and Collaboration Technologies*, pages 105–116. Springer.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7) :107–117.

A. Broder, S. Glassman, M. Manasse, and G. Zweig. 1997. Syntactic clustering of the web. In *6th International World Wide Web Conference*, pages 393–404.

Robin Burke. 2002. Hybrid recommender systems : Survey and experiments. *User modeling and user-adapted interaction*, 12(4) :331–370.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334) :183–186.

Roc'io Cañamares and Pablo Castells. 2018. Should i follow the crowd ? a probabilistic analysis of the effectiveness of popularity in recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 415–424.

Erion Çano and Maurizio Morisio. 2017. Hybrid recommender systems : A systematic literature review. *Intelligent Data Analysis*, 21(6) :1487–1524.

Alket Cecaj, Marco Mamei, and Franco Zambonelli. 2016. Re-identification and information fusion between anonymized CDR and social network data. *J. Ambient Intell. Humaniz. Comput.*, 7(1) :83–96.

Sonny Han Seng Chee, Jiawei Han, and Ke Wang. 2001. Rectree : An efficient collaborative filtering method. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 141–151. Springer.

Hung-Hsuan Chen, Chu-An Chung, Hsin-Chien Huang, and Wen Tsui. 2017. Common pitfalls in training and evaluating recommender systems. *ACM SIGKDD Explorations Newsletter*, 19(1) :37–45.

Minmin Chen. 2017. Efficient Vector Representation for Documents through Corruption. *arXiv e-prints*, page arXiv :1707.02377.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464.

Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 638–649.

Wei Chen, Zhendong Niu, Xiangyu Zhao, and Yi Li. 2014. A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web*, 17(2) :271–284.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

Paras Chopra. 2010. The ultimate guide to a/b testing. *Smashing Magazine*, 119.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824.

Davide Cirillo, Silvina Catuara-Solarz, Czuee Morey, Emre Guney, Laia Subirats, Simona Mellino, Annalisa Gigante, Alfonso Valencia, María José Rementeria, Antonella Santuccione Chadha, et al. 2020. Sex and gender differences and biases in artificial intelligence for biomedicine and healthcare. *NPJ digital medicine*, 3(1) :1–11.

Virginia Clinton and Shafiq Khan. 2019. Efficacy of open textbook adoption on learning performance and course withdrawal rates: A meta-analysis. *AERA Open*, 5(3) :2332858419872212.

Frank Coffield, Kathryn Ecclestone, Elaine Hall, and David Moseley. 2004. Learning styles and pedagogy in post-16 learning : A systematic and critical review.

François Coste. 2016. Learning the language of biological sequences. In *Topics in Grammatical Inference*, pages 215–247. Springer.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2014. Training deep neural networks with low precision multiplications. *arXiv e-prints*, page arXiv :1412.7024.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198. ACM.

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46.

Liang-Zhong Cui, Fu-Liang Guo, and Ying-jie Liang. 2018. Research overview of educational recommender systems. In *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, pages 1–7.

Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, 39(2) :1–49.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress ? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109.

M Daniluk, T Rocktäschel, J Welbl, and S Riedel. 2019. Frustratingly short attention spans in neural language modeling. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*, volume 5. International Conference on Learning Representations (ICLR).

Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2014. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 1(11) :1–21.

Peteris Daugulis. 2011. A note on a generalization of eigenvector centrality for bipartite graphs and applications. *Networks*, 59(2) :261–264.

Luis M De Campos, Juan M Fernández-Luna, Juan F Huete, and Miguel A Rueda-Morales. 2010. Combining content-based and collaborative recommendations : A hybrid approach based on bayesian networks. *International journal of approximate reasoning*, 51(7) :785–799.

Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Pierpaolo Basile. 2008. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 163–170.

Colin De La Higuera. 2010. *Grammatical Inference : learning automata and grammars*. Cambridge University Press.

Dennis DeCoste. 2006. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd international conference on Machine learning*, pages 249–256.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6) :391–407.

Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. 2007. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3) :217–255.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 189–198.

Pedro Domingos and Michael Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2) :103–130.

Robin Donaldson, David Nelson, and Eric Thomas. 2012. 2012 florida student textbook survey. Technical report, Tallahassee, Florida.

Hendrik Drachsler, Toine Bogers, Riina Vuorikari, Katrien Verbert, Erik Duval, Nikos Manouselis, Guenter Beham, Stephanie Lindstaedt, Hermann Stern, Martin Friedrich, et al. 2010. Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. *Procedia Computer Science*, 1(2) :2849–2858.

Hendrik Drachsler, Katrien Verbert, Olga C Santos, and Nikos Manouselis. 2015. Panorama of recommender systems to support learning. In *Recommender systems handbook*, pages 421–451. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284. Springer Berlin Heidelberg.

Cynthia Dwork and Aaron Roth. 2013. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4) :1–277.

Nikk Effingham. 2013. *An introduction to ontology*. John Wiley & Sons.

Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie : A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 World Wide Web Conference*, pages 1775–1784. International World Wide Web Conferences Steering Committee.

Michael D Ekstrand, Michael Ludwig, Joseph A Konstan, and John T Riedl. 2011. Rethinking the recommender research ecosystem : reproducibility, openness, and lenskit. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 133–140.

Mehdi Elahi, Francesco Ricci, and Neil Rubens. 2014. Active learning in collaborative filtering recommender systems. In *International Conference on Electronic Commerce and Web Technologies*, pages 113–124. Springer.

Tal Feldman and Ashley Peake. 2021. End-to-end bias mitigation: Removing gender bias in deep learning.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3) :355–369.

Benjamin Fung, ke Wang, Rui Chen, and Philip Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42.

Veerraju Gampala, Jaideep Vallapuneni, Pavan Kumar Ande, Ravindra Kumar Indurthi, and Nichenametla Rajesh. 2021. Comparative study on telugu text classification using machine learning and deep learning models. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1393–1398.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16) :E3635–E3644.

Mustansar Ghazanfar and Adam Prugel-Bennett. 2010. Building switching hybrid recommender system using machine learning classifiers and collaborative filtering. *IAENG International Journal of Computer Science*, 37(3).

Daniela Godoy and Analia Amandi. 2008. Hybrid content and tag-based profiles for recommendation in collaborative tagging systems. In *2008 Latin American Web Conference*, pages 58–65. IEEE.

Nadav Golbandi, Yehuda Koren, and Ronny Lempel. 2011. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 595–604.

David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12) :61–70.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig : Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614.

Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 133–138. IEEE.

M. Gori and A. Pucci. 2007. Itemrank : a random-walk based scoring algorithm for recommender engines. In *IJCAI Conference*, pages 2766–2771,.

Laura A Granka, Thorsten Joachims, and Geri Gay. 2004. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479.

Philip J Guo and Katharina Reinecke. 2014. Demographic differences in how students navigate through MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 21–30. ACM.

Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. Wtf : The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 505–514. ACM.

Prakhar Gupta, Matteo Pagliardini, and Martin Jaggi. 2019. Better word embeddings by disentangling contextual n-gram information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 933–939, Minneapolis, Minnesota. Association for Computational Linguistics.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR.

Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Bhat S Harish, Devanur S Guru, and Shantharamu Manjunath. 2010. Representation and classification of text documents : A brief review. *IJCA, Special Issue on RTIPPR (2)*, pages 110–119.

Donna Harman. 1992. Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10.

Zellig S. Harris. 1954. Distributional Structure. *WORD*, 10(2-3) :146–162.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer.

Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems : A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56 :9–27.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM.

Christina Hendricks, Stefan A. Reinsberg, and Georg W Rieger. 2017. The adoption of an open textbook in a large physics course: An analysis of cost, outcomes, use, and perceptions. *The International Review of Research in Open and Distributed Learning*, 18(4).

Jon Herlocker, Joseph A Konstan, and John Riedl. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4) :287–310.

Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :5–53.

Antonio Hernández-Blanco, Boris Herrera-Flores, David Tomás, and Borja Navarro-Colorado. 2019. A Systematic Review of Deep Learning Approaches to Educational Data Mining. *Complexity*, 2019 :1–22.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8) :1735–1780.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1) :89–115.

Jiri Hron, Karl Krauth, Michael I Jordan, and Niki Kilbertus. 2020. Exploration in two-stage recommender systems. *arXiv preprint arXiv :2009.08956*.

Z. Huang, H. Chen, and D. Zheng. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1) :116–142,.

Z. Huang, X. Li, and H. Chen. 2005. Link prediction approach to collaborative filtering. In *ACM/IEEE-CS joint conference on Digital libraries*, page 141–142.

Gordon Hughes. 1968. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1) :55–63.

Mohammed E Ibrahim, Yanyan Yang, David L Ndzi, Guangguang Yang, and Murtadha Al-Maliki. 2018. Ontology-based personalized course recommendation framework. *IEEE Access*, 7 :5180–5199.

Yuki Ichimura and Katsuaki Suzuki. 2017. Dimensions of MOOCs for quality design : Analysis and synthesis of the literature. *International Journal*, 11(1) :42–49.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to train bert with an academic budget.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Jansche. 2003. Parametric models of linguistic count data. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 288–295.

Rajiv S Jhangiani, Farhad N Dastur, Richard Le Grand, and Kurt Penner. 2018. As good or better than commercial textbooks: Students' perceptions and outcomes from using open digital and open print textbooks. *The Canadian Journal for the Scholarship of Teaching and Learning*, 9(1).

Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model.

Thorsten Joachims. 1998. Text categorization with support vector machines : Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 103–112.

Martijn Kagie, Michiel Van Wezel, and Patrick JF Groenen. 2011. Map based visualization of product catalogs. In *Recommender Systems Handbook*, pages 547–576. Springer.

Antonios Karatzoglou, Adrian Jablonski, and Michael Beigl. 2018. A seq2seq learning approach for modeling semantic trajectories and predicting the next location. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 528–531.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns : Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoust. Speech Signal Process.*, 35 :400–401.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 941–951, Berlin, Germany. Association for Computational Linguistics.

Nikhil Ketkar. 2017. Stochastic gradient descent. In *Deep learning with Python*, pages 113–132. Springer.

Diederik Kingma and Jimmy Ba. 2014. Adam : A method for stochastic optimization. *International Conference on Learning Representations*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.

Janet Kolodner. 2014. *Case-based reasoning*. Morgan Kaufmann.

Joseph A Konstan and Gediminas Adomavicius. 2013. Toward identification and adoption of best practices in algorithmic recommender systems research. In *Proceedings of the international workshop on Reproducibility and replication in recommender systems evaluation*, pages 23–28.

Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37.

Y. Koren and Koren. 2008. Factorization meets the neighborhood : a multifaceted collaborative filtering model. *ACM KDD Conference*, 4(1) :426–434,.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4).

Walid Krichene, Nicolas Mayoraz, Steffen Rendle, Li Zhang, Xinyang Yi, Lichan Hong, Ed Chi, and John Anderson. 2019. Efficient training on very large corpora via gramian estimation. In *International Conference on Learning Representations*.

Matevž Kunaver, Tomaž Požrl, Matevž Pogačnik, and Jurij Tasič. 2007. Optimisation of combined collaborative recommender systems. *AEU-International Journal of Electronics and Communications*, 61(7) :433–443.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Aristomenis S Lampropoulos, Paraskevi S Lampropoulou, and George A Tsihrintzis. 2012. A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. *Multimedia Tools and Applications*, 59(1) :241–258.

Amy N Langville, Carl D Meyer, Russell Albright, James Cox, and David Duling. 2006. Initializations for the nonnegative matrix factorization. In *Proceedings of the twelfth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 23–26. Citeseer.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553) :436–444.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324.

D. Lemire and A. Maclachlan. 2005. Slope one predictors for online rating-based collaborative filtering. In *SIAM Conference on Data Mining*.

Jerome Y Lettvin, Humberto R Maturana, Warren S McCulloch, and Walter H Pitts. 1959. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11) :1940–1951.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the association for computational linguistics*, 3 :211–225.

David D Lewis and Marc Ringuette. 1994. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93.

Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. 2015. MOOC video interaction patterns : What do they tell us ? In *Design for teaching and learning in a networked world*, pages 197–210. Springer.

Ruizhi Liao, Daniel Moyer, Miriam Cha, Keegan Quigley, Seth Berkowitz, Steven Horng, Polina Golland, and William M. Wells. 2021. Multimodal representation learning via maximization of local mutual information.

D. Liben-Nowell and J. Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7) :1019–1031,.

Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4) :2065–2073.

Chenxi Lin, Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Benyu Zhang, and Jian Wang. 2007. Collaborative filtering using cluster-based smoothing. US Patent App. 11/377,130.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1) :76–80.

Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4) :309–317.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1150–1160, New York, NY, USA. Association for Computing Machinery.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2020. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pages 463–473.

Danilo Mandic and Jonathon Chambers. 2001. *Recurrent neural networks for prediction : learning algorithms, architectures and stability*. Wiley.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Antonio Maratea, Alfredo Petrosino, and Mario Manzo. 2013. Generation of description metadata for video files. In *Proceedings of the 14th International Conference on Computer Systems and Technologies*, pages 262–269.

Jonathan Mayer, Patrick Mutchler, and John C. Mitchell. 2016. Evaluating the privacy properties of telephone metadata. *Proceedings of the National Academy of Sciences*, 113(20) :5536–5541.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48.

Robert McNaughton and Seymour A. Papert. 1971. *Counter-Free Automata (M.I.T. Research Monograph No. 65)*. The MIT Press.

N. Meinshausen. 2013. Sign-constrained least squares estimation for high-dimensional regression. *Electronic Journal of Statistics*, 7 :607–1631,.

Rada Mihalcea and Andras Csomai. 2007. Wikify ! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv :1301.3781 [cs]*. ArXiv : 1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning-based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3).

Juan Daniel Valor Miró, Pau Baquero-Arnal, Jorge Civera, Carlos Turró, and Alfons Juan. 2018. Multilingual videos for MOOCs and OER. *J. Educ. Technol. Soc.*, 21(2) :1–12.

Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20.

Miquel Montaner, Beatriz Lopez, and Josep Lluis De La Rosa. 2003. A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4) :285–330.

Raymond J Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204.

Mohammad-Hossein Nadimi-Shahraki and Mozhde Bahadorpour. 2014. Cold-start problem in collaborative recommender systems : Efficient methods based on ask-to-rate technique. *Journal of computing and information technology*, 22(2) :105–113.

Arvind Narayanan and Vitaly Shmatikov. 2006. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles : a survey. *Scientometrics*, 117(3) :1931–1990.

Malvina Nissim, Rik van Noord, and Rob van der Goot. 2020. Fair is better than sensational : : Man is to doctor as woman is to doctor. *Computational Linguistics*, 46(2) :487–497.

José M Noguera, Manuel J Barranco, Rafael J Segura, and Luis Martínez. 2012. A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215 :37–52.

Mark O'Connor and Jon Herlocker. 1999. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR workshop on recommender systems*, volume 128. Citeseer.

Eli Pariser. 2011. *The filter bubble : What the Internet is hiding from you.* Penguin UK.

Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Torben Pedersen. 2005. *HTTPS, Secure HTTPS*, pages 268–269. Springer US, Boston, MA.

J. Pei, Jiawei Han, B. Mortazavi-Asl, Helen Pinto, Q. Chen, U. Dayal, and M. Hsu. 2001. Prefixspan, : mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings 17th International Conference on Data Engineering*, pages 215–224.

Xi Peng. 2020. A comparative study of neural network for text classification. In *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, pages 214–218.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Maria Perez-Ortiz, Claire Dormann, Yvonne Rogers, Sahan Bulathwela, Stefan Kreitmayer, Emine Yilmaz, Richard Noss, and John Shawe-Taylor. 2021. X5learn: A personalised learning companion at the intersection of ai and hci. In *26th International Conference on Intelligent User Interfaces - Companion*, IUI '21 Companion, page 70–74, New York, NY, USA. Association for Computing Machinery.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation : a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1) :1–15.

Marcelo OR Prates, Pedro H Avelar, and Luis C Lamb. 2020. Assessing gender bias in machine translation : a case study with google translate. *Neural Computing and Applications*, 32(10) :6363–6381.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, pages 157–163.

Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. 2021. Sinr : Fast computing of sparse interpretable node representations is not a sin ! In *Advances in Intelligent Data Analysis XIX*, pages 325–337, Cham. Springer International Publishing.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8) :9.

Anand Rajaraman and Jeffrey David Ullman. 2011. Mining of massive datasets : Data mining (ch01). *Min. Massive Datasets*, 18 :114–142.

Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. 2002. Getting to know you : learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM Conference on Recommender Systems*, pages 240–248.

Paul Resnick and Hal R Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3) :56–58.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Bernhard Rieder, Ariadna Matamoros-Fernández, and Òscar Coromina. 2018. From Ranking Algorithms to 'Ranking Cultures' Investigating the Modulation of Visibility in YouTube Search Results. *Convergence*, 24(1) :50–68.

Christopher Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. 2016. Linking users across domains with location data : Theory and validation. WWW '16. International World Wide Web Conferences Steering Committee.

Stephen E. Robertson and Karen Sparck Jones. 1988. *Relevance Weighting of Search Terms*, page 143–160. Taylor Graham Publishing, GBR.

Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1) :1–9.

Cañamares Roc'io, Pablo Castells, and Alistair Moffat. 2020. Offline evaluation options for recommender systems. *Information Retrieval*, 23(4) :387–410.

Marcos Wander Rodrigues, Seiji Isotani, and Luiz Enrique Zárate. 2018. Educational data mining: A review of evaluation process in the e-learning. *Telematics and Informatics*, 35(6) :1701–1717.

José Antonio Sánchez Rodriguez, Jui-Chieh Wu, and Mustafa Khandwawala. 2020. Two-stage session-based recommendations with candidate rank embeddings. In *Fashion Recommender Systems*, pages 49–66. Springer.

James Rogers and Geoffrey K Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3) :329–342.

Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. 2015. Active learning in recommender systems. In *Recommender systems handbook*, pages 809–846. Springer.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088) :533–536.

Alan Said and Alejandro Bellogin. 2014. Comparative recommender system evaluation : benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 129–136.

Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Recommender systems for large-scale e-commerce : Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324. Citeseer.

Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer.

D.T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D.E. Pritchard. 2014. Who does what in a massive open online course ? *Communications of the ACM*, 57(4) :58–65.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1) :1–47.

Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. 2016. Hierarchical attention networks. *CoRR, abs/1606.02393*.

Pierre Sermanet, Soumith Chintala, and Yann LeCun. 2012. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3288–3291. IEEE.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382.

Matthew A Shapiro and Han Woo Park. 2015. More than entertainment : YouTube and public responses to the science of global warming and climate change. *Social Science Information*, 54(1) :115–145.

Sagar Sharma and Simone Sharma. 2017. Activation functions in neural networks. *Towards Data Science*, 6(12) :310–316.

Martin Shepperd and Michelle Cartwright. 2001. Predicting with sparse data. *IEEE Transactions on Software Engineering*, 27(11) :987–998.

Beerud Sheth and Pattie Maes. 1993. Evolving agents for personalized information filtering. In *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*, pages 345–352. IEEE.

Abdulhadi Shoufan and Fatma Mohamed. 2017. On the likes and dislikes of youtube's educational videos: A quantitative study. In *Proceedings of the 18th Annual Conference on Information Technology Education*, SIGITE '17, page 127–132, New York, NY, USA. Association for Computing Machinery.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv :1703.00810*.

William Simão de Deus and Ellen Francine Barbosa. 2020. The use of metadata in open educational resources repositories: An exploratory study. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 123–132.

Barry Smyth and Paul Cotter. 2000. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3) :53–59.

Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 445–454.

Ayse Saliha Sunar, Erik Novak, and Dunja Mladenic. 2020. Users' learning pathways on cross-site open educational resources. In *CSEDU (2)*, pages 84–95.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *ICML*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Latanya Sweeney. 2013. Discrimination in online ad delivery. *Communications of the ACM*, 56(5) :44–54.

Julian Szymański and Maciej Naruszewicz. 2019. Review on wikification methods. *AI Communications*, 32(3) :235–251.

Frédéric Tantini, Alain Terlutte, and Fabien Torre. 2010. Sequences classification by least general generalisations. In *International Colloquium on Grammatical Inference*, pages 189–202. Springer.

Andrew H Turpin and William Hersh. 2001. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 225–231.

María Cora Urdaneta-Ponte, Amaia Mendez-Zorrilla, and Ibon Oleagordia-Ruiz. 2021. Recommendation systems for education: Systematic review. *Electronics*, 10(14).

Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. 2009. Dimensionality reduction : a comparative. *J Mach Learn Res*, 10(66-71) :13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

S. Vucetic and Z. Obradovic. 2005. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*, 7(1) :1–22,.

Boya Wang, Jianqing Xu, Junbao Li, Cong Hu, and Jeng-Shyang Pan. 2017. Scene text recognition algorithm based on faster rcnn. In *2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS)*, pages 1–4. IEEE.

Congcong Wang, Paul Nulty, and David Lillis. 2020. A comparative study on word embeddings in deep learning for text classification. In *Proceedings of the 4th International*

Conference on Natural Language Processing and Information Retrieval, NLPIR 2020, page 37–46, New York, NY, USA. Association for Computing Machinery.

Tianchong Wang and Dave Towey. 2017. Open educational resource (oer) adoption in higher education: Challenges and strategies. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 317–319.

Markus Weimer, Alexandros Karatzoglou, Quoc Le, and Alex Smola. 2007. Cofirank-maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems, 21st Annual Conference on Neural Information Processing Systems 2007*, pages 222–230.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

David Wiley. 2014. *An Open Education Reader*. no.

Ian H Witten and Timothy C Bell. 1991. The zero-frequency problem : Estimating the probabilities of novel events in adaptive text compression. *Ieee transactions on information theory*, 37(4) :1085–1094.

Chris Wong. 2018. Sequence based course recommender for personalized curriculum planning. In *International Conference on Artificial Intelligence in Education*, pages 531–534. Springer.

Lili Wu, Sam Shah, Sean Choi, Mitul Tiwari, and Christian Posse. 2014. The browsemaps : Collaborative filtering at linkedin. *RSWeb@ RecSys*, 1271.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. 2018. Word mover's embedding: From Word2Vec to document embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4524–4534, Brussels, Belgium. Association for Computational Linguistics.

Bin Xu, Jiajun Bu, Chun Chen, and Deng Cai. 2012. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30.

Sheng-Yuan Yang and Chun-Liang Hsu. 2010. A new ontology-supported and hybrid recommending information system for scholars. In *2010 13th International Conference on Network-Based Information Systems*, pages 379–384. IEEE.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics : human language technologies*, pages 1480–1489.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 269–277.

Hilmi Yildirim and Mukkai S Krishnamoorthy. 2008. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 131–138.

Takashi Yokomori and Satoshi Kobayashi. 1998. Learning local languages and their application to dna sequence analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 10 :1067–1079.

Kyung-Hyan Yoo and Ulrike Gretzel. 2011. Creating more credible and persuasive recommender systems : The influence of source characteristics on recommender system evaluations. *Recommender systems handbook*, pages 455–477.

Chi-Chih Yu, Toru Yamaguchi, and Yasufumi Takama. 2013. A hybrid recommender system based non-common items in social media. In *2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (iCAST 2013 & UMEDIA 2013)*, pages 255–261. IEEE.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact : Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180.

Andrew Zhai, Dmitry Kislyuk, Yushi Jing, Michael Feng, Eric Tzeng, Jeff Donahue, Yue Li Du, and Trevor Darrell. 2017. Visual discovery at pinterest. In *Proceedings of the 26th*

International Conference on World Wide Web Companion, pages 515–524. International World Wide Web Conferences Steering Committee.

Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. 2014. Addressing cold start in recommender systems : A semi-supervised co-training algorithm. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 73–82.

Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. 2006. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 549–553. SIAM.

Yi Zhang, Jamie Callan, and Thomas Minka. 2002. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88.

Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next : a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51. ACM.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv :1511.08630*.

Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 315–324.

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

Titre : Recommandation de Ressources Educatives Libres dans le projet X5GON

Mot clés : Ressources Educatives Libres, Education Ouverte, Système de recommandation

Résumé : Ces dernières années, les pratiques d'apprentissage en ligne n'ont cessé de croitre, la pandémie mondiale du COVID-19 a encore accéléré cette tendance. Pour atteindre l'objectif de développement durable numéro 4 : « l'éducation de qualité et tout au long de la vie », l'UNESCO fait de l'apprentissage en ligne et des REL (Ressources Educatives Libres) les aspects centraux de cette politique. Dans un contexte où le nombre de ressource et d'utilisateur est pléthorique, des algorithmes de recommandation de contenu semblent indispensables pour guider les apprenants à travers les ressources. Néanmoins, l'emploi de la recommandation à des fins pédagogiques soulève des problématiques spécifiques non étudiées jusqu'alors. De plus, le manque de jeux de données libres disponibles complexifie l'évaluation et la comparaison des approches et ne permet pas l'emploi de méthodes gourmandes en données qui semblent pourtant les plus prometteuses. Dans ce document, nous nous intéressons à la problématique de la recommandation à visée pédagogique à large-échelle et dans un contexte éducationnel non-formel où les données sont non structurées. En particulier, nous explorerons la question d'un ordre satisfaisant de consultation des ressources ainsi que celle de mise à disposition de jeux de données libres pour cette tâche.

Title: Recommendation of Open Educational Resources in the X5GON project

Keywords: Open Educative Ressources, Technologies to Enhance Learning, Educationnal Data Mining

Abstract: In recent years, e-learning practices have continued to grow, with the global pandemic of COVID-19 further accelerating this trend. In order to achieve the sustainable development goal number 4: "quality education throughout life", UNESCO makes online learning and OERs (Open Educational Resources) the central aspects of this policy. In a context where the number of resources and users is plethoric, content recommendation algorithms seem indispensable to guide learners through the resources. Nevertheless, the use of recommendation for pedagogical purposes raises specific issues that have not been studied so far. Moreover, the lack of available free datasets makes it difficult to evaluate and compare approaches and does not allow the use of data-driven methods that seem to be the most promising. In this paper, we focus on the problem of pedagogical recommendation on a large scale and in a non-formal educational context where the data is unstructured. In particular, we will explore the question of a satisfactory order of consultation of resources as well as that of making available free datasets for this task.