Génération de questions à choix multiples thématiques à partir de bases de connaissances

Tanguy Raynaud

N° d'ordre NNT : 2019LYSES066

THESE DE DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de

L'Université Jean Monnet

Ecole Doctoral ED488

Sciences, Ingénierie et Santé

Discipline : Informatique

Soutenue publiquement le 28/02/2019 par

Tanguy RAYNAUD

Génération de Questions à Choix Multiples Thématiques à Partir de Bases de Connaissances

---

Devant le jury composé de :

Laurent Besacier, PR, *Université Grenoble Alpes*, Président

Brigitte Grau, PR, *Ecole Nationale Superieure d'Informatique pour l'Industrie et l'Entreprise*, Rapporteur

Catherine Faron-Zucker, MCF HDR, *Université Nice Sophia Antipolis*, Rapporteur

Frederique Laforest, PR, *INSA de Lyon*, Directrice de thèse

Julien Subercaze, MCF, *Université Jean Monnet*, Co-Directeur

Tanguy RAYNAUD

Génération de Questions à Choix Multiples Thématiques à Partir de Bases de Connaissances

Thèse en Informatique, Décembre 2018

Rapporteurs : Catherine Faron Zucker et Brigitte Grau

Examinateur : Laurent Besacier

Encadrants : Frederique Laforest et Julien Subercaze

Université Jean Monnet

École Doctorale ED488 Sciences, Ingénierie, Santé

Laboratoire Hubert Curien

Equipe Connected Intelligence

18 Rue Professeur Benoît Lauras

42000 Saint-Etienne

Résumé de la thèse

L'évaluation de connaissances à travers un support de questions à choix multiples est une méthode fiable et largement utilisée, y compris dans des contextes officiels, comme pour l'examen du code de la route. Cette méthode d'évaluation offre en effet de nombreux avantages, comme une égalité de notation entre les candidats, ou de façon plus pragmatique, une possibilité de correction automatique. L'émergence des MOOCs, des cours dispensés sous un format numérique, à contribué à accroître ce besoin d'évaluation automatique. Les travaux de cette thèse s'inscrivent ainsi dans ce contexte, en proposant une solution permettant de générer des questions thématiques, c'est à dire des questions centrées autour d'un thème prédéfini.

Les travaux présentés dans cette thèse utilisent des bases de connaissances comme sources de données pour générer automatiquement des questions à choix multiples thématiques. L'utilisation de bases de connaissances dans ce contexte pose ainsi un certain nombre de défis scientifiques qui constituent les contributions des travaux présentés :

— Les entités des bases de connaissances ne sont généralement pas explicitement corrélés à des thèmes. Cette thèse présente ainsi une méthode basée sur les méta-données de Wikipedia permettant d'identifier et de trier les entités de bases de connaissances en fonction de thèmes prédéfinis.

— Pour qu'une question soit intelligible, son énoncé doit être grammaticalement correct, et contenir suffisamment d'informations pour lever toute ambiguité quand-à la bonne réponse. Dans cette optique, nous avons introduit des modèles de questions permettant d'identifier des entités au sein de bases de connaissances, et de générer des énoncés en langage naturel.

— Dans une questions à choix multiples, les distracteurs (mauvaises réponses) sont aussi important que l'énoncé, de mauvais distracteurs rendant la question trop facile. Dans une dernière contribution, nous présentons la méthode utilisée pour sélectionner des distracteurs qui soient non seulement pertinents vis-à-vis de l'énoncé de la question, mais aussi de son contexte.

Abstract (English Version)

The use of multiple choice questions to assess knowledge is a reliable and widely used method, even in official contexts. Such a method offers many advantages, including equality of marking between candidates, or, more pragmatically, the possibility of automatic correction. With the emergence of MOOCs (courses delivered in a digital format), the need for automatic evaluation has increased. The scope of this thesis is part of this context, by proposing a solution that enables automatic thematic question generation.

The work presented in this thesis uses knowledge bases as data sources to automatically generate thematic multiple-choice questions. The use of knowledge bases in this context thus raises several scientific challenges that constitute the contributions of the presented work :

— Knowledge base entities are generally not explicitly correlated to themes. This thesis presents a method based on Wikipedia metadata to identify and sort knowledge base entities according to predefined themes.

— In order to be intelligible, a question must be grammatically correct, and must include enough information to remove any ambiguity about the correct answer. To that end, we have introduced question templates to identify entities within knowledge bases and generate natural language statements.

— In a multiple choice questions, distractors (wrong answers) are no less important than the statement. Wrong distractors are easilly discarded and affect the whole question difficulty. In a last contribution, we present the method used to select distractors that are not only relevant to the question's statement, but also to its context.

Remerciements

Bien peu de chercheurs peuvent prétendre avoir réalisé leurs travaux de doctorat seuls et coupés du monde, et je ne me compte définitivement pas parmi eux. En effet, une thèse reste avant tout une aventure humaine, et ne serait probablement pas possible sans qu'un certains nombre d'acteurs y participent, tant d'un point de vue professionnel que personnel. C'est pour cette raison que je tiens à remercier dans ce manuscrit un certain nombre de personnes qui m'ont aidé à faire de cette thèse un moment agréable.

Je tiens ainsi tout d'abord à remercier mes encadrants, Frédérique Laforest et Julien Subercaze, pour leur soutien et leurs conseils tout aux long de ces trois années.

Je remercie également Brigitte Grau et Catherine Faron-Zucker pour le temps qu'elles ont accordé à la relecture de ce manuscrit et pour les remarques pertinentes et constructives qui m'ont entre autre permis d'améliorer significativement celui-ci. J'adresse de la même manière mes remerciements à Laurent Besacier pour sa participation au jury.

Il est important de mentionner dans ces remerciement ceux qui furent mes collègues et amis pendant ces années de thèse, je veux bien sur parler des membres de l'ex équipe Satin. Merci donc à Jules, Pierre-Olivier, Abderrahmen et Syed pour ces bons moments passés en votre compagnie au quotidien.

Je porte une attention toute particulière à Pierre-Yves qui avec qui j'ai pu partager nombre de repas, qui n'auraient probablement pas eu la même saveur sans le soupçon d'humour qu'il a su apporter.

Je fait également une mention spéciale pour Damien, avec qui j'ai partagé un appartement et des bons moments pendant une bonne partie de cette thèse. C'est lui qui m'a également fait découvrir le Kung Fu, activitée qui m'a permis de travailler sur moi en compagnie de Emilie et Cécile.

Je remercie également tous mes amis lyonnais pour les soirées et Week ends passés en leur compagnie, tout particulièrement Sullivan et son canapé qui m'ont permis

de résider sur Lyon, mais également Leo, Mathieu et Boris pour les bons moments passés ensemble.

Je remercie également mes parents, et ma famille de façon générale, pour m'avoir toujours soutenu dans mes choix et poussé à travailler dès mon plus jeune âge. Cette thèse n'aurait en effet pas été possible sans leur soutien sans faille durant toutes ces années.

Ces remerciements ne seraient pas complets si je n'accordait pas ces quelques lignes à Sophie, qui à su me supporter pendant toutes ses années tout en m'apportant le soutien et le réconfort dont j'avais besoin au quotidien. J'en profite pour mentionner ses parents qui m'ont toujours réservé un accueil chaleureux.

Et pour finir je tiens également à remercier toutes les personnes que je n'ai pas mentionné dans ces quelques lignes mais qui ont contribué d'une façon ou d'une autre à cette aventure et à faire de moi quelqu'un de meilleur.

Table des matières

1 Introduction

1.1 Contexte

Les questionnaires à choix multiples sont largement utilisés, notamment dans les milieux éducatifs, pour vérifier l'acquisition de connaissances. Ils se présentent sous la forme d'une liste de questions, chacune associée à des propositions de réponses, habituellement au nombre de 4 (GRAESSER et WISHER, 2001). Parmi ces propositions de réponses, une ou plusieurs d'entre elles sont des réponses valides à la question, les autres sont des mauvaises réponses appelées distracteurs. L'utilisation de ces questionnaires pour lévaluation de connaissances dans un contexte éducatif offre un double intérêt : il permet de vérifier qu'un candidat a bien compris un élément du cours, mais également qu'il est capable de démêler le vrai du faux, et donc faire preuve d'esprit critique en écartant les mauvaises réponses.

La vérification d'acquis basée sur des questionnaires à choix multiples ne se limite pas aux milieux scolaires, mais est également largement répandu sur le Web. On trouve ainsi des questionnaires de ce type à la fin des cours de type MOOC (Massive Open Online Course) afin de permettre aux étudiants de vérifier qu'ils ont bien compris le contenu de la leçon, ou dans le cas contraire, de s'améliorer. C'est d'ailleurs dans ce contexte d'enseignement basé sur des MOOCs que se situe le projet européen Mooctab [1] qui a financé cette thèse. Les questionnaires à choix multiples sont également largement répandus dans des contexte officiels, en servant notamment de support pour l'examen du code de la route, ou des concours d'entrée en écoles d'ingénieur. On peut expliquer la popularité de ces questionnaires par plusieurs facteurs :

— **Les questions sont simples à concevoir :** En identifiant un concept clé au sein d'un paragraphe ou d'une leçon, il est facile d'interroger une personne sur des éléments relatifs à ce concept.

— **Les résultats sont simples à corriger :** Les bonnes réponses d'un questionnaire à choix multiples étant connues à l'avance, la correction est extrêmement simple. Elle est même souvent automatique, et cet argument contribue très largement à sa popularité.

1. Site Web du projet : `http://mooctab.com/`

— **La notation est équitable :** La notation est équitable dans la mesure où elle est déterministe et ne dépend pas du correcteur.

Pour toutes ces raisons, les questionnaires à choix multiples offrent un moyen idéal pour vérifier facilement et à moindre coût pour l'enseignant si un élève a assimilé une leçon, et dans quelle mesure. Les questionnaires à choix multiples portent généralement sur des faits simples ou des éléments précis et se limitent donc à la vérification des connaissances de la personne interrogée. Ils reflètent cependant difficilement sa capacité à mettre en application les concepts théoriques.

Bien que les questions à choix multiples offrent des facilités concernant leur création et leur correction, il est toutefois important de préciser que certains prérequis sont nécessaires pour assurer la qualité du contenu proposé :

— **L'énoncé** doit éviter d'utiliser des formulations ambigües,

— **La bonne réponse** et les distracteurs doivent être homogènes en terme de taille ou de contenu, et disposés de façon aléatoire,

— **Les distracteurs** doivent être crédibles, et donc influencer la personne interrogée dans le choix des réponses.

La nécessité des évaluations dans les milieux éducatifs a mené la communauté scientifique à s'intéresser aux problématiques de la génération automatique de questions. L'objectif est de mettre en place automatiquement un ensemble de questions en partant du contenu d'une leçon, de façon à réduire substantiellement le travail des enseignants, sans pour autant sacrifier la qualité des questions générées. Plusieurs approches proposent ainsi des solutions permettant de générer des questions de tous types, notamment des questions à choix multiples, mais aussi des questions ouvertes, des textes à trous, etc.

La génération automatique de questions à choix multiples pose cependant un certain nombre de contraintes à prendre en compte pour obtenir un résultat de qualité :

— **Cibler des éléments centraux pour générer la question :** Il est nécessaire d'identifier automatiquement au sein d'un texte ou d'un domaine les éléments pertinents qui peuvent faire l'objet d'une question.

— **Générer un énoncé en langage naturel :** L'énoncé doit valider les règles orthographiques et grammaticales de la langue dans laquelle il est généré.

— **Identifier la bonne réponse :** Il est indispensable de faire en sorte que la réponse automatiquement choisie soit effectivement une bonne réponse à la question posée.

— **Trouver un ensemble de distracteurs :** A partir des données disponibles, ou en se référant à des sources externes (dans le cas d'une leçon par exemple), un ensemble de distracteurs proches de la bonne réponse doivent être identifiés. Ces distracteurs ne doivent pas être des bonnes réponses à la question posée.

Parallèlement à l'intérêt scientifique pour le domaine de la génération automatique de questions, le domaine de l'informatique a connu l'émergence du Web sémantique, et l'apparition des bases de connaissances. Ces bases de connaissances se distinguent des bases de données relationnelles traditionnelles principalement dans la façon dont les données sont stockées. En effet, les bases de connaissances stockent les données sous la forme de graphes qui unissent entre elles un ensemble d'entités, à l'aide de relations. Ainsi, contrairement à une base de données relationnelles où les entités de même type possèdent systématiquement les mêmes attributs et relations, les bases de connaissances suivent des schémas permettant une plus grande expressivité. La manipulation des bases de connaissances, et la recherche d'informations au sein de ces dernières font ainsi l'objet d'un domaine scientifique à part entière. Le langage SPARQL y est notamment utilisé pour rechercher de l'information au sein de ces bases de connaissances, souvent exprimées en format RDF.

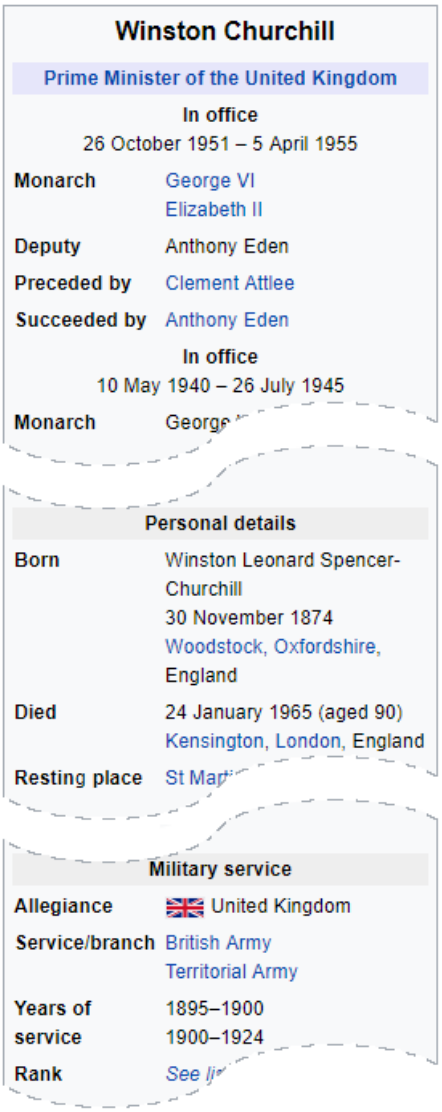
Les bases de connaissances regroupent de grandes quantités d'informations. La nature des bases de connaissances permet de stocker facilement des informations factuelles concernant une personne, un lieu, un événement ou tout autre entité réelle ou imaginaire. Ces informations sont stockées à l'aide de relations unissant entres elles des entités, ou unissant des entités à des littéraux. Concernant une personne, on peut ainsi trouver des relations telles que (<Personne>, *dateNaissance*, <date>) où la date sera un littéral, c'est-à-dire un élément composé de texte qui ne fera pas référence à une autre entité, et (<Personne>, *lieuNaissance*, <Lieu>) où le lieu fera référence à une autre entité de la base, comme une ville ou un pays. Ensemble, ces relations permettent de représenter sémantiquement la date et le lieu de naissance d'une personne. Certains faits sont cependant plus complexes à décrire, et nécessitent l'utilisation de nœuds intermédiaires pour en décrire le contenu. Prenons l'exemple d'un politicien élu pour un mandat présidentiel. Décrire le mandat avec une unique relation n'aurait aucun sens : (<Personne>, *electionPoste*, <Poste>) ou (<Personne>, *dateElection*, <date>). En effet, ce schéma ne permet d'attribuer à un politicien qu'un et un seul mandat au cours de sa carrière. Les nœuds intermédiaires sont ainsi utilisés pour résoudre ce problème, et permettre la description d'événements complexes. Dans notre exemple, cela se traduira par les relations suivantes : (<Personne>, *electionMandat*, <Mandat>), (<Mandat>, *dateDebut*, <date>) et (<Mandat>, *poste*, <Poste>) où <*Mandat*> désigne un nœud intermédiaire auquel sont rattachées toutes les informations sur un mandat donné. On peut alors, en ajoutant autant de nœuds que nécessaire, décrire les différents mandats d'un politicien au cours de sa carrière. Grâce à cette organisation sous forme de nœuds, on peut décrire avec une grande précision une grande quantité d'information au sein des bases de connaissances.

Afin de trouver des données permettant d'approvisionner massivement le contenu de bases de connaissances, plusieurs approches se sont basées sur Wikipedia [2]. Cette encyclopédie collaborative en ligne regroupe en effet un grand nombre d'informations, qui offrent l'avantage d'évoluer dans le temps en précision, en exactitude et en quantité. Des bases de connaissances comme *Yago*, *Freebase* ou *DBPedia* ont ainsi vu le jour en utilisant ces données comme source principale, les informations pouvant être issues soit des articles de Wikipedia à l'aide d'outils de traitement du langage, soit de données plus structurées, comme les cadres d'informations. La figure 1.1 montre un exemple partiel de cadre d'information issu de la page *Winston Churchill* de la version anglaise de Wikipedia. On peut y voir que les données sont formatées sous formes de triplets, où Winston Churchill est systématiquement le sujet de la relation. Ces triplets décrivent certains aspects de sa vie politique, personnelle ou militaire en présentant ces différentes informations avec un prédicat et un objet. Chaque objet peut être soit une référence à une autre entité (article) de Wikipedia, soit un littéral (date, label). Ces triplets sont donc des candidats idéaux à la création de bases de connaissances.

L'abondance et la structure de ces données les rendent intéressantes pour le traitement automatisé, et c'est notamment le cas pour le domaine de la génération automatique de questions. En effet, l'information contenue dans ces bases de connaissances est une candidate appropriée à la création automatique de questions, et tout particulièrement à la création de questions à choix multiples. Les données y sont structurées suivant des schémas précis, et sont ainsi comparables. Si on reprend l'exemple du mandat politique, il suffit d'identifier la relation *electionMandat* pour être en mesure de poser des questions telles que "*Quel politicien a été élu au poste de <Poste> en <date> ?*" ou "*En quelle année a été élu <Personne> au poste de <Poste> ?*", etc. On constate ainsi que pour une seule relation de la base de connaissances, on obtient une variété de questions possibles. Par ailleurs, la structure de la base de connaissances permet d'identifier des éléments ou des entités qui sont proches de celles choisies pour générer la question. Dans le contexte de questions à choix multiples, cela permet d'identifier des distracteurs pertinents. On peut proposer par exemple d'autres politiciens ayant des dates et des postes similaires à la bonne réponse.

Les bases de connaissances constituent donc une source de données avantageuse dans le contexte de la génération automatique de questions à choix multiples. L'approche présentée dans ce manuscrit a pour objectif d'utiliser cette connaissance pour générer des questions de qualité, en rapport avec une thématique déterminée à l'avance. Nous expliquons ainsi dans la section suivante les différents challenges liés à cette problématique, et les différents objectifs d'un générateur de questions à choix multiples thématiques.

2. `https://en.wikipedia.org`



Winston Churchill

Prime Minister of the United Kingdom

In office
26 October 1951 – 5 April 1955

Monarch	George VI Elizabeth II
Deputy	Anthony Eden
Preceded by	Clement Attlee
Succeeded by	Anthony Eden

In office
10 May 1940 – 26 July 1945

Monarch	George

Personal details

Born	Winston Leonard Spencer-Churchill 30 November 1874 Woodstock, Oxfordshire, England
Died	24 January 1965 (aged 90) Kensington, London, England
Resting place	St Mar

Military service

Allegiance	United Kingdom
Service/branch	British Army Territorial Army
Years of service	1895–1900 1900–1924
Rank	*See li*

Fig. 1.1.: Exemple de cadre d'information partiel issu de Wikipedia contenant des méta-données sur Winston Churchill

1.2 Motivation et objectifs

L'évaluation de connaissances n'est pas un concept transversal, mais spécifique au contenu d'une leçon ou d'un domaine que l'on souhaite évaluer. Les questions doivent donc cibler les acteurs et les éléments principaux du domaine, tout en laissant de côté ceux qui sont moins importants. De façon générale, s'agit donc d'identifier ces éléments importants, et de s'en servir pour générer des questions à choix multiples. C'est pourquoi l'objectif de cette thèse est de présenter un générateur de questions thématiques, i.e. un générateur capable d'identifier au sein d'une thématique donnée, les éléments les plus importants afin de créer des questions ciblées.

Le format et l'abondance des données stockées au sein des bases de connaissances font de ces dernières des candidates idéales pour chercher des informations qui pourront être utilisées pour générer ces questions. Cependant, plusieurs limites doivent être considérées pour utiliser ces bases de connaissances comme support à la génération de questions thématiques :

1. **La notion de thème au sein des bases de connaissances :** Les entités des bases de connaissances sont décrites par les relations qui les composent, et par les méta-données qui décrivent leur nature. Cependant, il n'existe pas dans ces bases de connaissances de données explicites permettant de dire, par exemple, si une entité de type *personne* est plutôt liée à la thématique de l'histoire, ou à celle de la politique. Utiliser les données de bases de connaissances dans un contexte thématique implique ainsi nécessairement d'utiliser des informations externes à ces dernières pour y introduire la notion de thème.
2. **La validité des énoncés :** La génération des énoncés de questions, et la génération de phrases en langage naturel de façon générale, est un procédé complexe qui malgré une forte attention de la communauté scientifique, peine à donner des résultats satisfaisants. La génération des énoncés de questions, que ce soit dans leur formulation ou leur validité grammaticale ou orthographique, reste un problème central de la génération de questions.
3. **Respecter la complétude des faits :** Si un humain peut facilement identifier un ensemble d'éléments minimums nécessaires à la validité d'une question, l'automatisation de cette tâche reste un problème complexe. Si on reprend l'exemple du mandat politique, un humain peut facilement déterminer que la question "*Quel politicien a été élu au poste de <Poste> ?*" ne contient pas assez d'informations pour que l'on puisse y répondre de façon certaine. Il en va de même pour identifier les informations superflues d'une question, avec par exemple, la question "*Quel politicien né à <Lieu> en <date> a été élu au poste de <Poste> en <date> ?*". Il n'est en effet pas nécessaire d'énumérer toute la biographie d'une personne pour formuler une question, voire cela peut avoir l'effet inverse en donnant trop d'indications au lecteur. Pour que la question soit valide, il faut ainsi y insérer une quantité suffisante, nécessaire et sans excédent d'information.

Ces limites compliquent considérablement l'utilisation des bases de connaissances comme support thématique à la génération de questions à choix multiples. La résolution de ce problème implique ainsi nécessairement d'apporter des solutions intermédiaires au problème des données thématiques dans les bases de connaissances, ainsi qu'au problème de la génération en langage naturel des énoncés.

Le problème de **l'absence de thème au sein des bases de connaissances** repose sur le manque de méta-données permettant de catégoriser les entités de la base avec un ensemble de thèmes. Un de nos objectifs est donc de mettre en place un

procédé permettant d'identifier le degré d'appartenance des entités de la base à chaque thème d'un ensemble de thèmes prédéfinis, et de stocker ces données afin de pouvoir les réutiliser par la suite dans notre générateur de questions thématiques.

En ce qui concerne la **validité des énoncés** et la **complétude des faits**, nous fixons comme objectif commun la génération d'un énoncé qui soit à la fois valide grammaticalement et orthographiquement, et synthétique mais complet dans sa formulation, de façon à lever toute ambiguité.

La motivation finale de cette thèse est donc de proposer une solution permettant de générer des questions à choix multiples relatifs à un thème choisi a priori, en utilisant les données structurées des bases de connaissances. Ces questions doivent avoir une qualité comparable à des questions similaires qui seraient créées par des humains. La solution proposée devra offrir une solution à l'ensemble des problématiques scientifiques énoncées ci-dessus. Parallèlement, la solution recherchée devra répondre à un certain nombre de contraintes opérationnelles permettant une utilisation pratique. Ces objectifs peuvent-être décrits de la façon suivante :

1. **Un générateur thématique :** Une solution permettant de choisir en premier lieu la thématique à traiter, puis de générer des questions relatives à cette thématique.
2. **Des énoncés de questions de qualité :** Les énoncés devront valider les critères énoncés ci-dessus, et donc être valides grammaticalement et orthographiquement, tout en étant synthétiques et complets dans leurs formulations.
3. **Des bonnes réponses valides :** La bonne réponse associée à chaque question doit valider l'énoncé de la question.
4. **Des distracteurs crédibles :** Les mauvaises réponses associées à la question doivent être proches de la bonne réponse et pertinentes dans la thématique choisie, tout en étant effectivement fausses. Les distracteurs ne doivent donc pas valider l'énoncé de la question.
5. **Un générateur fonctionnel :** Une solution fonctionnelle de bout en bout permettant de générer des questions automatiquement, au sein desquels seront clairement identifiables l'énoncé, la bonne réponse et les distracteurs.

1.3 Contributions

Dans cette thèse, nous présentons un générateur de questions à choix multiples thématiques à partir de bases de connaissances. La mise en place de cette solution s'est heurtée à un certains nombre de problématiques scientifiques. Ces problématiques ont été étudiées et résolues séparément, puis intégrées à notre générateur de questions à choix multiples.

Nous pouvons ainsi décrire ces contributions de la façon suivante :

1. **Des bases de connaissances thématiques :** Les bases de connaissances ne contiennent nativement pas de notion de thème (histoire, arts, sports...). Il est par exemple impossible de dire si l'entité *France* se rapproche plutôt du thème de la géographie, ou de celui de l'histoire, ni dans quelle mesure, à moins que ces informations ne soient explicites dans la base. Cette première contribution propose d'exploiter les méta-données de Wikipedia, et plus particulièrement sa structure en graphes, pour introduire la notion de thème dans les bases de connaissances. Cette contribution concerne ainsi tout particulièrement les bases de connaissances dérivées de Wikipedia, comme DBpedia, Yago, Freebase, au sein desquelles nous sommes en mesure de définir un ensemble de thèmes, et d'identifier quelles entités sont liées à chacun des thèmes, et dans quelle mesure (classement des entités au sein des thèmes). Nous avons publié une démonstration permettant de juger de l'utilité de cette contribution dans la conférence internationale CIKM (*International Conference on Information and Knowledge Management*) (RAYNAUD et al., 2018a).
2. **Des modèles de questions :** Le problème de la génération de phrases en langage naturel est un problème récurrent de traitement du langage, qui s'applique particulièrement dans le domaine de la génération de questions, où il est nécessaire de générer un énoncé pour chacune des questions. Dans cette optique, nous avons défini dans cette thèse des *modèles de questions*, destinés à faciliter la génération de questions à choix multiples en utilisant des bases de connaissances comme sources de données . Les modèles de questions contiennent ainsi les informations nécessaires à la génération de l'énoncé en langage naturel, ainsi que les informations nécessaires à l'identification au sein d'une base de connaissances des entités qui seront insérées dans cet énoncé, ou utilisées en tant que mauvaises réponses. Ces modèles offrent également l'avantage de pouvoir inclure autant d'entités que nécessaire dans la question. Nous présentons également, en complément de cette contributions, les méthodes originales que nous avons mises en place pour approvisionner ces modèles de questions.
3. **Un générateur de questions thématiques :** Avec cette dernière contribution, nous utilisons la combinaison des modèles de questions définis précédemment avec le contenu thématique identifié au sein des bases de connaissances pour former des questions à choix multiples thématiques. En effet, notre générateur de questions est en mesure d'identifier au sein de la base de connaissances des entités en relation avec un thème prédéfini, de les combiner avec un modèle de questions pour former l'énoncé de la question, et d'en sélectionner d'autres, en lien avec la bonne réponse, qui formeront des distracteurs pertinents.

La combinaison de ces trois contributions a ainsi permis d'obtenir un générateur de questions à choix multiples thématiques à partir de bases de connaissances.

Ce générateur, innovant dans son fonctionnement et dans sa modularité, est une contribution en soi. L'ensemble du procédé a fait l'objet d'une publication dans la conférence internationale WI (*Web Intelligence*) (RAYNAUD et al., 2018b).

1.4 Organisation de la thèse

Ce manuscrit de thèse décrit les différentes contributions présentées ci-dessus. Le manuscrit s'organise de la façon suivante.

Nous présentons dans un premier chapitre, *État de l'art*, les travaux principaux existant dans le domaine de la génération automatique de questions. Nous abordons ensuite plus en détails les approches concernant la génération de questions à choix multiples, qui sont au centre de ce manuscrit. Ce premier chapitre détaille également les approches existant dans le domaine de l'extraction de thèmes dans des bases de connaissances, et dans le domaine de la génération automatique de langage naturel, nécessaire ici pour générer les énoncés des questions.

Nous détaillons ensuite dans les trois chapitres suivants les différentes contributions réalisées au cours de cette thèse :

Le second chapitre présente comment nous avons introduit *la notion de thème dans les bases de connaissances*. Nous expliquons dans ce chapitre la méthode originale que nous avons mise en place pour utiliser les méta-données de Wikipedia dans le but de créer et approvisionner des thèmes utilisables dans toutes les bases de connaissances dérivées de l'encyclopédie.

Le troisième chapitre présente les *modèles de questions* que nous avons définis pour simplifier la création de questions à choix multiples. Les modèles proposés dans cette thèse permettent d'associer de manière originale des énoncés avec des sous-graphes de bases de connaissances. Cette définition propose une alternative intéressante aux problèmes récurrents du domaine de la génération automatique de questions, notamment au regard de la complétude des énoncés et de la gestion des paraphrases.

Le quatrième chapitre, *Construction des questions à choix multiples*, détaille les procédés utilisés pour combiner modèles de questions et thèmes dans l'optique de produire des questions à choix multiples thématiques. Nous détaillons ainsi dans cette section notre procédé original de génération en langage naturel des énoncés à partir d'arbres syntaxiques, ainsi que les méthodes utilisées pour identifier des distracteurs pertinents pour la question mais aussi pour la thématique.

Dans un cinquième chapitre, *Expérimentations*, nous détaillons les expériences et les évaluations réalisées pour tester la fiabilité de notre générateur de questions et la qualité des résultats obtenus. Nous y détaillons ainsi l'évaluation manuelle qui a été réalisée pour juger de la qualité des questions générées, ainsi que les conditions de cette évaluation et ses résultats.

Nous terminons ce manuscrit par un chapitre de *Conclusion*, qui revient sur les contributions, synthétise les résultats obtenus, et présente quelques perspectives au travail proposé.

2 État de l'art

Introduction

La génération de questions à choix multiples thématiques est un procédé complexe qui soulève un certain nombre de problématiques scientifiques auxquelles il n'existe à l'heure actuelle pas de solutions satisfaisantes. Nous présentons ainsi dans ce premier chapitre les travaux réalisés par la communauté scientifique relatifs au domaine de la génération automatique de questions, ainsi qu'aux domaines de la thématisation d'entités et de la génération en langage naturels. Nous présentons ainsi pour chacune de ces parties les travaux existants, ainsi que leur limites.

Ce chapitre s'organise de la façon suivante : La section 2.1 présente de façon générale les approches existantes dans le domaine de la génération de questions. La section 2.2 détaille plus particulièrement la gestion des distracteurs dans les approches de génération de questions à choix multiples. La section 2.3 présente la façon dont la difficulté des questions générées est gérée dans les approches existantes. La section 2.4 présente les différentes modes opératoires mis en place pour évaluer la qualité des questions générées automatiquement. La section 2.5 présente les approches existantes dans le domaine de la *génération de phrases en langage naturel*, nécessaire ici pour générer l'énoncé des questions. Ce chapitre se termine avec la section 2.6 qui présente les approches et les techniques utilisées dans l'état de l'art pour gérer la notion de thème.

2.1 Génération de Questions

La génération automatique de questions est la tâche qui consiste à créer un environnement propice à l'évaluation des connaissances d'une personne. Elle ne se limite ainsi pas à la stricte génération de l'énoncé de la question, mais s'étend également à l'ensemble des composants qui sont utiles à la question, comme la bonne et les mauvaises réponses, dans le cas de questions à choix multiples. Pour définir précisément la tâche consistant à générer automatiquement des questions, nous citons la définition introduite par (RUS et al., 2008) et reprise par (PIWEK et BOYER, 2012) :

"*La génération de questions est la tâche qui consiste à générer automatiquement des questions à partir de différentes sources de données, dont notamment du texte, des bases de données, ou des bases de connaissances. La génération de questions est considérée comme une tâche linguistique basée sur les 4 étapes suivantes : (1) quand poser la question, (2) sur quoi porte la question, c'est-à-dire la sélection du contenu, (3) déterminer le type de question, et (4) construire la question*".

La génération automatique de questions est un domaine actif, car il offre de nombreuses applications, particulièrement dans le milieu éducatif, dans le cadre de l'évaluation de connaissances d'étudiants(LE et al., 2014). On peut générer des questions de différentes natures, en fonction du besoin recherché lors de l'évaluation :

- **Les questions ouvertes :** Il s'agit de questions qui nécessitent une réponse complexe de la part de la personne interrogée. Elles permettent d'évaluer de façon précise la compréhension d'un utilisateur sur un domaine donné. Ces questions sont difficilement compatibles avec des systèmes de correction automatique (exemple : "*Why is Paris the capital of France*").
- **Les questions Oui/Non :** Ces questions n'admettent que deux possibilités de réponses : *oui/non* (ou *vrai/faux*). On trouve ainsi deux types de questions Oui/Non : (i) une phrase interrogative commençant généralement par *Is + subject + verb*, à laquelle les utilisateurs répondront par *yes* ou *no* (exemple : "*Is Buckingham Palace located in London City ?*"), (ii) une phrase affirmative, et l'utilisateur doit déterminer si elle est vrai ou fausse (exemple : "*The french country is bordered by exactly 8 different countries*". *True or False ?*")
- **Les textes à trous :** Ce type de questions est généralement constitué de phrases affirmatives desquelles un mot (ou ensemble de mots) a été retiré. L'utilisateur répond à la question en déterminant le mot manquant. Si le mot manquant est proposé parmi plusieurs autres possibilités, on se situe dans le cas de figure du questionnaire à choix multiple, détaillé ci-dessous. Dans les autres cas, l'utilisateur devra s'appuyer sur la structure et les éléments de la phrase pour déterminer le mot manquant (exemple : "*In 60 BC, Pompey, Crassus and* _____ *formed a political alliance called Triumvirate*").

— **Les questionnaires à choix multiples :** Les questionnaires à choix multiples offrent à la personne interrogée plusieurs propositions de réponses. L'énoncé des questions admet ainsi généralement une réponse courte, souvent composée d'un mot ou d'un groupe de mots. En anglais, on parle de *WH-questions*, étant donné que les pronoms interrogatifs commencent généralement par WH : Who (qui), Where (où), When (Quand), Which (Quel), etc. (Exemple : "*Where was John Lennon born ?*" *A. England, B.Scotland, C. USA, D. Canada*).

La génération de questions à choix multiples constitue une tâche particulière, dans la mesure où elle combine deux parties distinctes : la génération de l'énoncé de la question, et la génération des mauvaises réponses qui l'accompagnent, appelées *distractors*. Certaines approchent regroupent ces deux tâches et seront détaillées dans cette section. D'autres approchent se spécialisent sur la tâche de génération de distracteurs, et seront détaillées dans une section spécifique (section 2.2).

En plus du type de question généré, on peut également distinguer les approches de génération automatique de questions en fonction de la source de données utilisée. On distingue ainsi les questions générées en se basant sur des données issues de corpus de textes, des questions générées à partir de bases de données en graphes, autrement appelées bases de connaissances.

(Rakangor et Ghodasara, 2015) et (Vibhandik et Samant, 2014) nous proposent des revues de l'état de l'art sur le domaine de la génération automatique de questions. Divers systèmes y sont comparés, notamment sur les méthodologies, les types de questions générées et les évaluations.

2.1.1 La génération de questions à partir de textes

Pour générer des questions, les systèmes automatisés ont besoin de sources de données afin d'identifier des éléments qui formeront le fond des questions. La source de données est souvent un texte, ou un ensemble de textes. Cette approche doit sa popularité au fait que, notamment dans les milieux éducatifs mais également sur la grande majorité des supports informatifs, la connaissance est présentée sous la forme d'un texte (la leçon), qui contient l'ensemble des informations que l'auteur considère comme importantes. Ainsi, afin de permettre aux générateurs de questions d'extraire ces informations depuis le texte, ces systèmes font généralement appel au domaine du *Traitement Automatique du Langage* (TAL), permettant d'analyser un ensemble de phrases d'un langage donné, et d'en comprendre l'agencement et le sens des mots qui la composent.

(Guo et al., 2016) ont proposé *Questimator*, un générateur de questions à choix multiples thématique se basant sur des articles de Wikipedia. A partir d'un thème

donné, le système génère un énoncé se présentant sous la forme d'un texte à trou. Plusieurs réponses pouvant convenir sont proposées à l'utilisateur. Ces réponses se présentent sous la forme de portions de phrases qui complètent l'énoncé. Questimator a la particularité d'exploiter la structure de Wikipedia pour identifier et ordonner des articles similaires, desquels sont issus les distracteurs associés aux questions.

On trouve également les travaux de (NARENDRA et al., 2013), qui proposent des questions sous forme de textes à trous, auxquels sont associés des propositions (bonne réponse + distracteurs). Les questions sont générées à partir d'un document donné, duquel sont extraits les résumés des phrases et les mots importants grâce à différents outils de traitement automatique du langage : ajout de tags POS (FINKEL et al., 2005), reconnaissance d'entités (TOUTANOVA et al., 2003), analyse de la structure (KLEIN et MANNING, 2003), et résumé de phrases (AGARWAL et MANNEM, 2011). Ces outils permettent d'identifier au sein du document des phrases importantes, et au sein de ces dernières, des mots-clés à supprimer pour créer les questions à trous.

(PANDEY et RAJESWARI, 2013) proposent de générer des questions respectant des contraintes liées à des niveaux scolaires, en se basant sur la taxonomie de Bloom (FOREHAND, 2010). Ils proposent pour cela un système reposant sur plusieurs agents travaillant parallèlement. Ces agents sont chargés d'extraire des mots-clés d'un document, de les traiter, et de générer les questions sur la base de modèles d'énoncés auxquels sont ajoutés ultérieurement les entités issues du document. Cependant, ces travaux n'intègrent pas à ce stade de système de classement des entités, permettant de les corréler avec un niveau d'éducation donné.

(LIU et al., 2012) proposent *G-ask*, un système de génération de questions à partir de documents décomposés sous la forme d'ensembles de phrases. Parmi ces phrases, seules les citations sont sources pour la génération de question. En effet, les auteurs du papier se basent sur les citations textuelles (POWLEY et DALE, 2007), à partir desquelles ils sont capables d'identifier, à partir d'outils de reconnaissance d'entités (RATINOV et ROTH, 2009), les éléments qui formeront l'objet de la question. Ensuite, avec un système de modèles de questions dépendant du contenu de la phrase, une question est générée. Ce système ne permet de produire que des questions ouvertes. G-ask ne s'intéresse donc pas aux réponses qui peuvent y être apportées. La table 2.1 liste les modèles utilisés par G-ask pour générer les questions, en se basant sur le contenu de la citation choisie.

(AWAD et DAHAB, 2014) ont mis en place un système capable de générer différents types de questions à partir de documents annotés dans cette optique. Ainsi, plusieurs types d'énoncés peuvent être générés, comme des textes à trous, ou des WH-questions (*Who, where, when,* etc.). Afin de n'autoriser systématiquement qu'une seule bonne

Rule	Category	Question Template
1	Opinion	Why +subject_auxiliary_inversion() ? What evidence is provided by +subject+ to prove the opinion ? Do any other scholars agree or disagree with +subject+ ?
2	Result	Subject_auxiliary_inversion() ? Is the analysis of the data accurate and relevant to the research question ? How does it relate to your research question ?
3	System	In the study of +subject+, why +subject_auxiliary inversion() ? What are the strength and limitations of the system ? Does it relate to your research question ?
4	Application	Why+Subject_Verb_Inversion() ? Could the problem have been approached more effectively from another perspective ? Does it relate to your research question ?
5	Method	In the study of +subject+, why +subject_auxiliary _inversion() ? Which dataset does +subject+ use for this experiment ? What are the strengths and limitations of this approach ?
6	Aim	Why does +subject+ conduct this study to +predicate+ ? What is the research question formulated by +subject+ ? What is +subject+s contribution to our understanding of the problem ?

Tab. 2.1.: Modèles de questions permettant au système G-ask de générerer des questions ouvertes (LIU et al., 2012)

réponse, les auteurs ont introduit un système de distracteurs original. En effet, le nombre de bonnes réponses proposées peut varier de zéro à trois, mais dans les cas où il n'y a aucune bonne réponse, ou qu'il y en a plusieurs, une proposition supplémentaire est ajoutée. Dans ces cas-là, la personne interrogée se verra proposer l'option "*Aucun de ces choix*" (*None of the above*) s'il n'y a aucune bonne réponse, "*Toutes ces propositions*" (*All of the above*) si tous les autres choix sont des bonnes réponses, ou "*La première et la deuxième propositions*" (*First and second choice*) si les deux premières propositions sont des bonnes réponses.

(BECKER et al., 2012) ont défini une approche permettant de choisir automatiquement au sein d'une phrase quelle entité constitue le meilleur candidat pour construire un énoncé de question à trous. Les différentes possibilités d'énoncés qu'offre une phrase sont illustrées par la figure 2.1. On peut noter qu'une simple phrase de 11 mots offre dans cet exemple 7 possibilités différentes pour générer un énoncé de question à trous. Ainsi, afin de déterminer la meilleure solution, les auteurs ont mis en place une méthode statistique basée sur des contributions humaines.

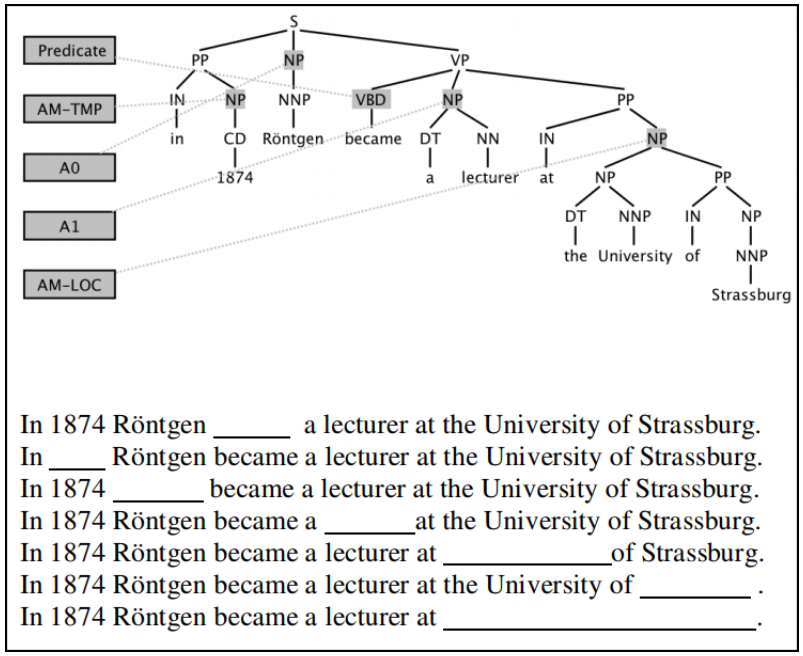


Fig. 2.1.: Les différentes possibilités d'énoncés de questions à trou offertes par une même phrase identifiées par (BECKER et al., 2012)

2.1.2 La génération de questions à partir de bases de connaissances

Dans le domaine de la génération automatique de questions, une seconde approche consiste à extraire des données à partir de bases de connaissances. Les bases de connaissances peuvent être définies comme étant des ontologies peuplées d'entités. En effet, une base de connaissances est décrite par un schéma qui conditionne et explicite les différentes classes. Cette description est généralement réalisée par une taxonomie, i.e. une organisation hiérarchique des classe (exemple : Personne $\rightarrow$ Artiste $\rightarrow$ Chanteur). Le schéma décrit également toutes les relations possibles entre ces différentes classes. Il s'agit de la description de l'univers lié à la base de connaissances, qui définit ainsi explicitement le champs des possibles. Ce schéma descriptif peut donc être assimilé à une ontologie. Parallèlement, les bases de connaissances sont peuplées d'entités qui correspondent aux instances des classes définies dans le schéma. Ces entités utilisent les relations préalablement définies dans le schéma pour décrire leur environnement, en faisant notamment référence à d'autres entités de la base. Les schémas qui décrivent les bases de connaissances varient fortement d'une base de connaissances à l'autre, notamment concernant la précision de la description de l'univers. Dans certains cas, il peut même arriver qu'il n'y ait pas de schéma. Dans d'autres cas, comme Freebase, ce schéma est extrêmement précis, et chaque relation est décrite par un *domaine* et une *portée*. Ces 2 éléments permettent de définir de façon précise les types (i.e. les classes) affectés par les relations.

En utilisant un sous-graphe issu d'une base de connaissances donnée, constitué d'un ensemble d'entités et de relations, il devient ainsi possible de décrire des faits complexes, et de créer des questions sur ces derniers. La figure 2.2 présente un sous-graphe issu d'une base de connaissances. On peut y voir un ensemble d'entités et de relations, et comment ces derniers s'assemblent pour former des faits. Ainsi, *person*, *date* ou *nationality* sont des classes qui décrivent des entités. Elles sont reliées entre elles par des relations. Ici, *hasNationality*, *birthDate* et *birthPlace* sont des exemples de relations. Les classes sont instanciées par des entités. Dans cet exemple, *Winston Churchill*, *British* ou *Oxfordshire, England* sont des exemples d'entités. Combinées, ces entités et ces relations forment des faits. Ainsi, cet exemple nous permettent de dire que "*Winston Churchill est né le 30 novembre 1874 à Oxfordshire, au Royaume-uni*".

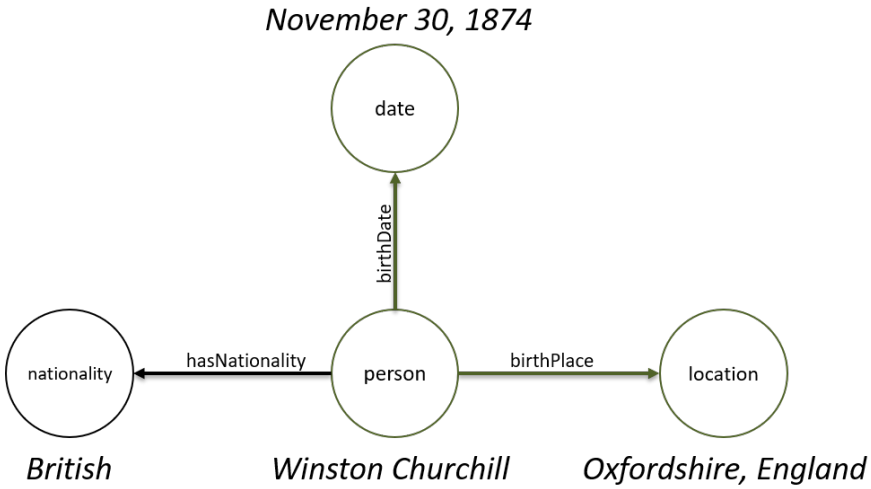


Fig. 2.2.: Exemple de sous-graphe de base de connaissances représentant diverses relations d'une personne et exemple d'instance

Différentes approches exploitent les avantages associés aux bases de connaissances pour générer des questions. On trouve notamment les travaux de (PAPASALOUROS et al., 2008) qui exploitent la structure de la base de connaissances pour générer des questions sur les entités qui la composent. Cette approche permet de générer des questions sur les liens entre les classes et leurs sous-classes au sein de la base, ou sur les propriétés de ces classes. Au total, les auteurs proposent 11 *stratégies* permettant de varier les types de questions générées. Ainsi, en utilisant la structure de la base, ce système est capable de formuler des affirmation vraies, et des affirmations fausses, ce qui est idéal pour générer des questions à choix multiples. Cependant, cette approche ne génère pas d'énoncé à la question, et se contente de demander à l'utilisateur de choisir la bonne option. Un énoncé unique est ainsi utilisé quelle que soit la question : "*Choose the correct answer*".

(ALSUBAIT et al., 2014) ont également mis en place une approche qui repose sur la structure de la base de connaissances pour générer des questions. En se basant

sur les liens entre les classes au sein de la base de connaissances, ils sont en mesure de générer des WH-questions, et des questions à trous. Pour ces deux types de questions, les auteurs génèrent des questions sous la forme de questions à choix multiples. La bonne réponse est accompagnée de distracteurs issus de la base de connaissances. Pour déterminer les distracteurs les plus appropriés, les auteurs ont mesuré la distance sémantique entre la bonne réponse et les autres entités de la base de connaissances. Ils ont ainsi pu déterminer au sein de cette dernière lesquelles étaient les plus crédibles. La limite de cette approche réside dans le fait que seules des relations simples (un seul triple) de la base de connaissances sont utilisées.

(AL-YAHYA, 2014) proposent *OntoQue*, un générateur de questions à choix multiples se basant sur des bases de connaissances. Ce système permet de générer deux types de questions : des questions vrai/faux, et des questions à choix multiples. Pour générer ces questions, divers aspects de la base de connaissances sont utilisés : les relations entre les différentes entités d'une part, et d'autre part les données issues de l'ontologie qui décrit les données. La limite de cette approche vient de deux points. (i) Les énoncés ne sont générées qu'à partir de triples simples de la base de connaissances, et sont basés sur des prédicats devant être manuellement interprétés (l'énoncé ou la partie de l'énoncé correspondant au prédicat est écrit en dur). (ii) Les distracteurs sont choisis aléatoirement parmi des entités de même type. Le type seul n'est pas suffisant pour que ces derniers soient crédibles aux yeux de la personne interrogée.

(DUAN et al., 2017) proposent une approche originale de génération automatique de questions, en se basant sur des données issues du domaine de la réponse automatique aux questions. En effet, en récupérant des questions fréquentes posées par des utilisateurs de *YahooAnswers*, les auteurs ont réutilisé des éléments d'énoncés pour produire de nouvelles questions. Ainsi, les entités présentes au sein des énoncés originaux sont substituées par de nouvelles entités, relatives à un thème préalablement déterminé automatiquement pour chaque question. Le thème des questions est déterminé en s'appuyant sur Freebase (BOLLACKER et al., 2008) et sur le parser de Stanford (KLEIN et MANNING, 2003). A l'heure actuelle, ce système est conçu pour approvisionner les données d'apprentissage de réseaux de neurones chargés de répondre automatiquement à des questions. Des questions sont ainsi massivement produites sans pouvoir préalablement sélectionner leurs thèmes, et les questions générées sont simples : elles ne concernent pas plus d'une entité.

(SEYLER et al., 2017) proposent une méthode de bout en bout pour générer automatiquement des questions à choix multiples à partir de bases de connaissances. A partir d'un domaine passé en paramètre, ce système est capable de générer des questions à choix multiples. Cette approche intègre la notion de difficulté des questions générées pendant la phase de génération, permettant de générer des questions à choix

multiples relatifs à un niveau scolaire prédéterminé. Les énoncés des questions sont générés à partir des différentes relations (prédicats) qui relient entre elles les entités de la question. Ceci constitue le point faible de cette approche : pour fonctionner correctement, la totalité des relations de la base de connaissances doit faire l'objet d'une verbalisation explicite.

(ROCHA et al., 2018) proposent de générer des questions à trous spécifiques à un domaine donné en se basant sur le contenu de DBpedia. Les éléments du domaine sont identifiés automatiquement dans la base de connaissances à partir d'un ensemble de mots-clés défini manuellement. Cette méthode est utilisée pour identifier des entités pertinentes autour desquelles seront centrées les questions. Les énoncés sont générés à partir d'une méthode des mêmes auteurs (ROCHA et FARON-ZUCKER, 2018).

2.1.3 Limites de la génération automatique de questions

Nous avons présenté ci-dessus les approches existantes dans le domaine de la génération automatique de questions. Cependant, certaines limites ne permettent pas de générer des questions de façon précise.

En ce qui concerne les techniques se basant sur l'extraction de connaissances depuis des documents, la limite récurrente provient du fait que de façon générale, les méthodes se limitent à un unique document. Dans le cadre de la génération de questions à choix multiples notamment, se limiter à un unique document ne permet pas de garantir que les données disponibles permettront de trouver un ensemble de distracteurs suffisamment pertinents pour être crédibles aux yeux de la personne interrogée.

Du point de vue de la génération de questions à partir de bases de connaissances, la principale limite vient du fait que les générateurs détectent difficilement les liens pertinents centrés autour d'une même entité. En effet, dans une base de connaissances, une entité peut être membre de plusieurs centaines de milliers de relations. Cependant, ces relations ne peuvent/doivent pas nécessairement être utilisées simultanément lors de la génération. Il convient donc d'identifier une sous-partie des éléments, qui doivent être utilisés simultanéments, afin de donner à la question générée un sens cohérent. Considérons l'exemple de sous-graphe de base de connaissances donné par la figure 2.2. Ici, le générateur de question pourrait générer une question telle que "*Which person with the British nationality is born in Oxforshire, England ?*". Cependant, cette dernière n'aurait aucun sens. En revanche, si on prend en compte le lien sémantique entre la date de naissance, et le lieu de naissance, il devient possible de générer une question cohérente comme "*Which person is born*

on November 30th 1874 in Oxforshire, England ?". C'est pourquoi des approches comme celle de (DUAN et al., 2017) se basent sur une ensemble de schémas qui ne génèrent des questions que sur les entités qui possèdent préalablement l'ensemble des relations requises dans la base de connaissances. La faiblesse de cette approche réside cependant dans la nécessité d'écrire manuellement et individuellement les éléments de l'énoncé correspondant à chacune des relations de la base de connaissances.

Ainsi, pour contourner le problème de déterminer au sein d'un graphe de connaissances quels sont les nœuds et les liens qui interragissent entre eux pour former un fait, la plupart des approches décrites ci-dessus se contentent de générer des questions à partir d'un triple unique. Pour illustrer cette assertion, reprenons l'exemple de la figure 2.2. Les relations *birthPlace* et *birthDate* sont sémantiquement liées. A partir de ces éléments, on attend ainsi une question type "*Which person is born on <date> in <location> ?*". Cependant, nombre d'approches préfèrent décomposer cette question en deux sous-questions distinctes : "*When is born <person> ?*" et "*Where is born <person> ?*". Ces questions sont instinctivement plus simples à générer, mais moins complètes dans leurs contenus.

Il n'existe ainsi à l'heure actuelle pas de solution permettant de générer automatiquement des questions à choix multiples thématiques. Cependant, la génération de questions à choix multiples thématiques est un processus complexe qui peut être décomposé en un ensemble problématiques distinctes. Nous présentons ainsi dans la suite de cet état de l'art les approches existantes vis-à-vis de chacune de ces problématiques.

2.2 Génération de distracteurs

Dans le cadre de questions à choix multiples, on nomme *distracteurs* (ou *distractors* en anglais) les mauvaises réponses qui accompagnent la ou les bonnes réponses à la question. Ainsi, la qualité d'une question est directement liée à la qualité de ses distracteurs, dans la mesure où des éléments peu crédibles seraient immédiatement écartés par les utilisateurs, qui pourraient ainsi déduire aisément la bonne réponse. (GRAESSER et WISHER, 2001) indiquent que le nombre idéal de réponses proposées au sein de questionnaires à choix multiples est de 4, soit 3 distracteurs. Deux processus de génération de distracteurs peuvent être distingués : (i) les distracteurs construits à partir de textes, et (ii) les distracteurs construits à partir de bases de connaissances.

2.2.1 Génération de distracteurs à partir de textes

Lors de la génération de questions à choix multiples à partir de textes, il convient de trouver des distracteurs pertinents, en s'aidant du contenu de ces derniers. De façon générale, on trouve peu d'approches qui permettent de générer des distracteurs à partir d'un texte unique. Il est en effet nécessaire de pouvoir comparer plusieurs éléments pour trouver des distracteurs pertinents. Certaines approches ont recours à des ressources externes (dictionnaires, WordNet, etc.) pour proposer des distracteurs adaptés.

(ALDABE et MARITXALAR, 2010) cherchent à générer des distracteurs proches de la bonne réponse, mais incorrectes dans le contexte pour éviter de proposer des distracteurs qui s'avéreraient être des bonnes réponses à la question. Pour cela, ils utilisent le logiciel Infomap, mis en place par (DOROW et WIDDOWS, 2003), qui indexe la totalité d'un corpus de textes, et calcule la distance sémantique entre chaque paire de mots. Ils sélectionnent ainsi comme distracteurs les mots ayant les scores de similarité les plus élevés avec la bonne réponse. Cependant, pour éviter aux personnes interrogées d'éliminer facilement certains distracteurs, les auteurs ont combiné cette mesure de distance sémantique avec une sélection basée sur la morphologie des candidats ainsi que sur leur sémantique. Les candidats sont analysés avec des dictionnaires pour déterminer si leurs types, ou leurs descriptions sont proches de la bonne réponse. Dans le cas contraire, ils ne sont pas retenus en tant que distracteurs.

(CORREIA et al., 2010) ont proposé une étude permettant de comparer les résultats obtenus par différentes méthodes de génération de distracteurs. Au total, 5 méthodes de génération de distracteurs ont été comparées :

1. **Génération manuelle :** Réalisés par des humains, les distracteurs générés manuellement devaient valider un ou plusieurs des critères suivants : (i) être un quasi synonyme/antonyme de la bonne réponse (ii) avoir une prononciation similaire à la bonne réponse, (iii) être un faux-ami similaire à la bonne réponse, pouvant induire en erreur un non-natif de la langue, et (iv) jouer sur la modification des préfixes/suffixes de la bonne réponse.
2. **Génération aléatoire :** Cette méthode, malgré son nom, n'est pas entièrement aléatoire. Elle sélectionne aléatoirement des mots parmi une liste ayant des caractéristiques communes avec la bonne réponse. Cette liste est construite à partir des mots ayant des POS tags (MARCUS et al., 1994) similaires à celui de la bonne réponse.
3. **Génération basée sur la distance :** Cette méthode se base sur la distance de Levenstein (LEVENSHTEIN, 1966), calculée à partir des scores obtenus entre la bonne réponse et les autres mots du corpus. Les distracteurs sont sélectionnés parmi les mots ayant une distance inférieure à 5 dans le graphe ainsi créé.
4. **Génération phonétique :** En se basant sur une table des erreurs d'écriture fréquentes, cette méthode génère volontairement des mots mal orthographiés pour induire en erreur la personne interrogée.
5. **Génération à partir de ressources lexicales :** Cette méthode permet de sélectionner automatiquement des synonymes ou antonymes de la bonne réponse en se basant sur des ressources lexicales contenant ce genre d'informations.

Contrairement à ce que l'on pourrait prévoir, l'évaluation des résultats indique que les distracteurs sélectionnés aléatoirement obtiennent le meilleur pourcentage de sélection, suivis de près par les distracteurs basés sur la distance sémantique.

(MITKOV et al., 2009) génèrent des distracteurs pour des questions préalablement générées à partir de textes éducatifs (MITKOV et HA, 2003). Ils comparent différentes approches de génération pour déterminer au travers d'une étude comparative laquelle donne les meilleurs résultats. Ainsi, les approches suivantes sont comparées :

1. **Similarité à partir de Wordnet :** Cette approche mesure la similarité entre les candidats et la bonne réponse en se basant sur Wordnet. Pour augmenter la précision de la mesure, les candidats ne sont pas comparés uniquement à la bonne réponse, mais également aux concepts proches de cette dernière.
2. **Similarité distributive :** Cette approche consiste à mesurer la co-occurrence linguistique des mots dans les phrases. On considère que les mots sont co-occurrents s'ils sont utilisés dans le même contexte dans deux phrases différentes. Les distracteurs sont sélectionnés parmi les mots disposant d'une co-occurrence élevée avec la bonne réponse.

3. **Similarité phonétique :** En se basant sur Soundex, qui permet de trier les mots selon leur phonétique, cette approche identifie des mots aux consonnances proches de la bonne réponse.
4. **Mélanges des approches précédentes :** Cette méthode ne génère pas de distracteurs à proprement parler, mais sélectionne des éléments parmi les approches précédentes de façon à proposer un choix varié de distracteurs.

Cette étude conclut qu'aucune des méthodes présentées n'offre de résultats particulièrement supérieurs aux autres, même si la dernière approche, regroupant un mélange des autres stratégies, se distingue légèrement.

(Brown et al., 2005) ont utilisé le guide pour créer des questions à choix multiples mis en place par (Graesser et Wisher, 2001). Ils ont ainsi mis en place un système basé sur plusieurs filtres successifs. Les distracteurs sont d'abord sélectionnés parmi les mots ayant les mêmes POS tags (Marcus et al., 1994) que la bonne réponse. L'ensemble de ces candidats est ensuite réduit en se basant sur Wordnet, utilisé ici pour trouver des mots (synonymes, antonymes, etc.) qui correspondent à la question posée. Pour restreindre les distracteurs proposés uniquement à des mots connus, l'analyse de fréquence des mots proposée par (Kilgarriff, 1995) est utilisée afin d'identifier les mots les plus fréquents parmi les candidats identifiés dans les étapes précédentes.

(Pino et Eskénazi, 2009) proposent une approche pour assister les créateurs de questions à choix multiples dans le choix des distracteurs. La génération n'est ainsi pas entièrement automatique, mais propose au créateur des questions une liste de distracteurs pouvant convenir, le choix final restant à la charge de ce dernier. Cette approche propose de générer des distracteurs de 5 types différents :

1. **Morphologique :** La racine de la bonne réponse est conservée, mais le suffixe ou le préfixe sont modifiés.
2. **Orthographique :** Propose un mot existant dont l'orthographe est similaire à la bonne réponse. Les candidats idéaux étant des mots composés des mêmes lettres que la bonne réponse, mais dont l'ordre des lettres est différent.
3. **Phonétique :** Les distracteurs sont choisis en fonctions des phonèmes qui les composent. En se basant sur une base de phonèmes, seuls les mots possédant les mêmes phonèmes que la bonne réponse, mais dans un ordre différent sont retenus.
4. **Orthographique et morphologique :** Une combinaison des deux premiers types de distracteurs.
5. **Phonétique et morphologique :** une combinaison des types 2 et 3.

Les types 4 et 5 sont nécessairement plus éloignés de la bonne réponse que les autres, et servent de témoins pour valider que le choix des réponses par les personnes

interrogées n'est pas fait au hasard, ces éléments devant ainsi être moins choisis que les autres propositions. La figure 2.3 donne des exemples de distracteurs obtenus à l'aide de chacune des méthodes présentées ci-dessus. Les résultats obtenus en utilisant cette approche dépendent de si les personnes interrogées sont natives de la langue dans laquelle sont formulées les questions.

Distractor Type	Target Word	Distractor
Morph	bored	boring
Orth	bread	beard
Phon	file	fly
OrthMorph	organ	groaning
PhonMorph	shared	shredded

Fig. 2.3.: Exemples de distracteurs générés en fonction de la bonne réponse et de la méthode utilisée dans l'approche de (PINO et ESKÉNAZI, 2009)

2.2.2 Génération de distracteurs à partir de bases de connaissances

Les bases de connaissances constituent un outil populaire dans la cadre de la génération de questions à choix multiples. Elles facilitent en effet la formation de questions en permettant d'identifier des faits au sein des triplets qui composent la base. Elles permettent également de trouver efficacement des distracteurs grâce au typage récurrent dans ces bases, ainsi qu'aux différentes relations qui composent le graphe de la base de connaissances, permettant de mesurer la distance qui sépare les entités deux à deux.

Plusieurs approches proposent de se baser à la fois sur le contenu et la structure de l'ontologie pour générer des questions à choix multiples (PAPASALOUROS et al., 2008)(AL-YAHYA, 2014)(CUBRIC et TOSIC, 2011). Pour sélectionner des distracteurs, ces approches utilisent la définition des classes de l'ontologie pour identifier des éléments similaires à la bonne réponse, qui ne vérifient toutefois pas les mêmes relations entre les sujets et objets du graphe de connaissances. Cependant, si ces approches donnent de bons résultats sur de petites ontologies ((PAPASALOUROS et al., 2008) utilisent par exemple l'ontologie Eupalinos Tunnel présentée en figure 2.4 qui n'est composée que d'une trentaine de classes), elles sont difficilement compatibles avec des bases de connaissances importantes, comme DBpedia, composées de centaines de milliers de concepts, qui conduiraient à la génération de distracteurs n'ayant aucun lien avec la bonne réponse.

(BHATIA et al., 2013) ont généré des questions à choix multiples se rapportant au domaine du baseball. Pour générer les distracteurs, ils ont identifié manuellement

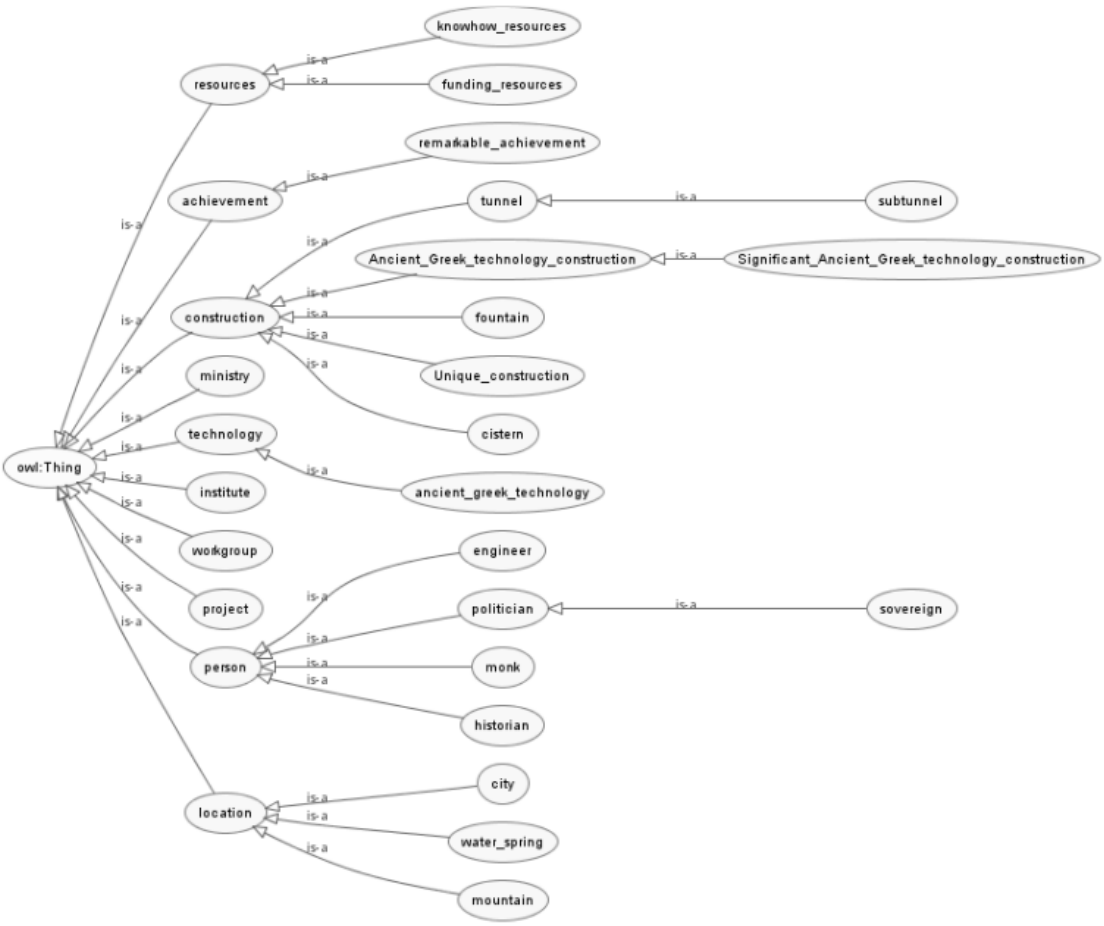


Fig. 2.4.: Ontologie Eupalinos Tunnel utilisée pour générer des questions à choix multiples dans l'approche de (PAPASALOUROS et al., 2008)

les différentes catégories de réponses possibles dans ce domaine (joueur, équipe, lieu, etc.), et déterminé les différents attributs pour chacune d'entre elles (nom, rôle, etc.). Ces différentes catégories sont ensuite approvisionnées automatiquement avec des données issues de Wikipedia. Les données contenant les mêmes attributs que les bonne réponses définissent ainsi des candidats aux caractéristiques extrêmement proches de l'entité constituant la bonne réponse. Cette approche permet de générer des distracteurs satisfaisants, mais on peut cependant noter que compte tenu de la nécessité d'identifier manuellement les classes et les attributs des bonnes réponses, cette approche peut difficilement être mise en place à grande échelle.

(SEYLER et al., 2017) génèrent des questions à choix multiples dans un niveau de difficulté donné. Pour sélectionner des distracteurs crédibles aux yeux de la personne interrogée, les auteurs ont identifié deux contraintes :

1. **Une contrainte de type :** Le type des distracteurs doit être identique ou proche du type de la bonne réponse.

2. **Une contrainte de distance sémantique :** Les distracteurs doivent être liés à la bonne réponse, donc avoir une distance sémantique faible avec elle.

Les auteurs ont ainsi mis en place une mesure qui prend en compte ces deux contraintes, en se basant sur l'hypothèse que plus le résultat de cette mesure sera élevé, plus les distracteurs seront facile à écarter. Ainsi, cette approche leur permet de sélectionner des distracteurs en prenant en compte un niveau de difficulté prédéterminé.

(Guo et al., 2016) génèrent des questions à choix multiples pour n'importe quel thème contenu dans Wikipedia. Les questions sont formées sous forme de textes à trous, et la personne interrogée doit choisir parmi un ensemble de réponses pouvant convenir. Pour identifier les distracteurs, cette approche se sert des catégories de Wikipedia pour identifier des thèmes similaires au thème préalablement passé en paramètre. Au sein de ces thèmes, des phrases ayant les mêmes arbres syntaxiques que la bonne réponse sont extraites. La recherche d'arbres syntaxiques similaires permet de s'assurer que les propositions s'intègreront dans l'énoncé. Une mesure de similarité (Kiros et al., 2015) est finalement effectuée dans cet ensemble de candidats pour identifier ceux qui sont les plus proches de la bonne réponse, donc les plus crédibles. Cette approche combine ainsi la génération de distracteurs à partir de bases de connaissances, et à partir de textes, dans la mesure où la structure, mais aussi le contenu de Wikipedia sont utilisés pour obtenir des distracteurs de qualité.

(Foulonneau, 2011) proposent de générer des questions à choix multiples spécifiques aux milieux éducatifs en se basant sur le contenu de DBpedia. Dans cette approche, les distracteurs sont sélectionnés aléatoirement parmi des entités de DBpedia possédant un type similaire à la bonne réponse. Ainsi, bien que les entités sélectionnées soient du même type que la bonne réponse, elles peuvent être facilement éliminées par les personnes interrogées dans la mesure où elles n'ont pas nécessairement de lien avec la question.

(Stasaski et Hearst, 2017) se servent d'ontologies pour générer des questions à choix multiples. Ils qualifient les distracteurs comme étant des éléments qui ne doivent pas être synonymes de la bonne réponse, tout en devant être suffisamment proches de cette dernière pour être crédibles. Ainsi, pour sélectionner les distracteurs, ils ciblent des entités semblables à la bonne réponse en cherchant celles avec qui elle a des propriétés en commun. Deux cas de figures sont envisagés : (i) des propriétés composées de la même relation et la même entité, (ii) des propriétés composées de la même entité mais d'une relation différente. Ainsi, les stratégies suivantes ont été testées :

1. **Deux propriétés identiques :** Les distracteurs sont sélectionnés parmi les entités ayant exactement deux propriétés en commun avec la bonne réponse.

2. **Une propriété identique, une différente :** les distracteurs sont sélectionnés parmi les entités ayant une propriété en commun avec la bonne réponse, et une ciblant la même entité mais avec une relation différente.
3. **Deux propriétés différentes :** les distracteurs sont sélectionnés parmi les entités ayant deux propriétés ciblant des entités communes avec la bonne réponse, mais avec des relations différentes.
4. **Une propriété identique :** Les distracteurs sont sélectionnés parmi les entités ayant exactement une propriété en commun avec la bonne réponse.

La figure 2.5 illustre le fonctionnement des différentes stratégies testées pour identifier des distracteurs. Les résultats obtenus en comparant les distracteurs générés à l'aide de ces différentes stratégies semblent indiquer que la stratégie 2 (*une propriété identique, une différente*) donne les meilleurs résultats.

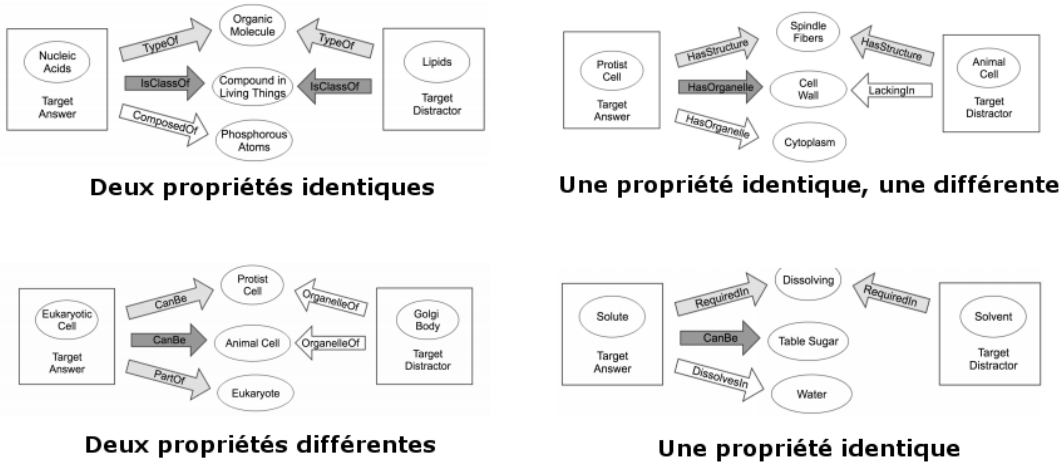


Fig. 2.5.: Illustration des stratégies utilisées pour sélectionner des distracteurs dans l'approche de (STASASKI et HEARST, 2017)

2.2.3 Limites de la génération de distracteurs

Les distracteurs constituent au même titre que l'énoncé un gage de qualité d'une question à choix multiples. En effet, si une question est bien formulée, mais possède des distracteurs faciles à éliminer par la personne interrogée, la qualité de la question est amoindrie. Que ce soit lors de la création manuelle d'une question à choix multiples, ou lors de sa génération automatique, trouver des distracteurs pertinents constitue donc un problème de premier plan.

En ce qui concerne les distracteurs générés à partir de textes, la difficulté récurrente consiste à trouver des candidats qui soient à la fois pertinents et compatibles. En effet, la mesure de distance entre deux mots au sein de textes ou de corpus de textes ne garantit pas nécessairement de forte corrélation entre ces derniers. Par ailleurs,

les mots fortement corrélés ne sont pas nécessairement compatibles avec le type de la bonne réponse, et peuvent souvent être éliminés facilement. Pour pallier ce problème, certaines approches ont ainsi recours à des modifications phonétiques ou orthographiques du mot, mais si ces propositions sont efficaces avec des personnes non-natives de la langue, un natif peut les écarter facilement.

En ce qui concerne les distracteurs générés à partir de bases de connaissances, le problème récurrent est généralement lié au contexte des questions générées. En effet, même en restreignant les candidats aux entités de même type et possédant une distance sémantique faible avec la bonne réponse, on remarque que les distracteurs sont souvent écartés facilement, car la notion de contexte est étrangère aux bases de connaissances.

Il n'existe ainsi pas à l'heure actuelle d'approches permettant de générer des distracteurs qui prennent en compte le contexte (i.e. le thème) de la question, et ce tout particulièrement dans les bases de connaissances, au sein desquelles c'est la notion de thème qui est absente. Ce problème de notion de thème dans les bases de connaissances doit donc être résolu pour pouvoir les utiliser comme sources de données à la génération de distracteurs thématiques. Dans cette optique, nous présentons dans le chapitre 3 une solution permettant d'introduire la notion de thème au sein des bases de connaissances.

2.3 Difficulté des questions

Les approches de génération de questions posent parfois le problème de la difficulté des questions générées. En effet, que ce soit dans un cadre scolaire ou non, certains utilisateurs peuvent ne pas avoir les connaissances requises pour répondre à une question, quand d'autres la trouveront triviale. Estimer la difficulté des questions générées est donc une tâche importante du processus. Certaines approches vont plus loin en proposant de générer des questions directement dans un niveau donné. Nous détaillons dans cette section ces deux types d'approches.

(Moreno et al., 2006) ont mis en place un guide pour aider les créateurs de questions à choix multiples, en détaillant point par point les attentes et les prérequis qui permettent d'obtenir une question à choix multiples complète et cohérente. Dans cet article, ils précisent que les facteurs qui déterminent la difficulté d'une question ne sont pas nécessairement liés uniquement au contenu de son énoncé, mais qu'ils peuvent également être induits par des ambiguités involontaires dans la formulation ou dans le choix des distracteurs. Par ailleurs, ils ajoutent que les questions contenant des choix comme "*aucun des éléments précédents*" ou "*tous les éléments précédents*" sont nécessairement plus compliquées que celles qui n'en ont pas, dans la mesure où ce type de proposition requiert de la part de l'utilisateur une connaissance de chaque réponse proposée, limitant de ce fait les raisonnements par élimination.

2.3.1 Estimation de la difficulté

Dans un grand nombre d'approches de génération de questions à choix multiples, le problème de la difficulté se pose pendant la phase d'évaluation, donc après que les questions aient été générées. On cherche alors à corréler les questions obtenues avec un niveau scolaire de référence, ou une valeur absolue de difficulté plus générale.

(Gronlund, 1982) a défini dans son ouvrage un ensemble de critères permettant de décrire les éléments nécessaires à l'évaluation de l'acquisition de la connaissance. Ces critères comprennent notamment les niveaux de connaissance, de compréhension et d'application autour des concepts évalués. Appliqués aux questions à choix multiples, ces critères permettent de juger de la difficulté et de l'utilité des questions générées. Ainsi, plusieurs approches utilisent cet ouvrage comme référence pour estimer la difficulté et la portée des questions générées (Al-Yahya, 2014)(Mitkov et al., 2009)(Bhatia et al., 2013).

(Cubric et Tosic, 2011) ont utilisé la taxonomie de Bloom (Bloom Benjamin et Krathwohl, 1956) pour catégoriser l'utilité et la difficulté des questions générées.

La taxonomie de Bloom peut être décrite comme étant un modèle descriptif des niveaux d'acquisition et de maîtrise de la connaissance, et ce particulièrement dans le milieu pédagogique. Elle se décompose en 6 niveaux différents : (1) la connaissance, (2) la compréhension, (3) l'application, (4) l'analyse, (5) la synthèse et (6) l'évaluation. Cette taxonomie est définie dans le but d'aider les enseignants à formuler des questions permettant de tester l'un ou l'autre de ces niveaux en fonction des attentes. Elle est donc parfaitement adaptée dans un contexte d'évaluation de la difficulté des questions générées.

(AL-YAHYA, 2014) utilisent une mesure binaire pour juger de la difficulté des questions générées. Ainsi, les évaluateurs doivent choisir si "*la question est utile pour l'enseignement*", ou si ce n'est pas le cas.

2.3.2 Génération dans un niveau de difficulté donné

De façon pragmatique, les questions à choix multiples sont souvent créés pour évaluer les connaissances et les acquis d'un groupe de personnes connu au moment de leur création. Le niveau des questions dépend donc directement du niveau des personnes à interroger. Le défi réside ainsi dans la possibilité de réussir à générer automatiquement des questions adaptées à un niveau prédéterminé.

(ALSUBAIT et al., 2013) introduisent une méthode basée sur la similarité pour être en mesure de contrôler la difficulté des questions générées. Selon les auteurs de ce papier, la difficulté d'une question ne s'évalue pas uniquement sur la difficulté des éléments de la question, mais sur la connaissance que la personne interrogée a du contexte de la question. C'est pourquoi ces auteurs mettent en avant la corrélation entre la difficulté et les niveaux scolaires, qui ne correspondent pas uniquement à l'acquisition de connaissances sur des domaines très précis, mais plutôt à une perception plus vaste d'un environnement. Les deux critères qui permettent à une personne de répondre à une question sont présentés comme étant : (1) la connaissance nécessaire à sa résolution et (2) la capacité cognitive requise pour résoudre la question. Ainsi, une mesure de similarité est utilisée pour identifier des éléments qui, compte tenu du niveau de connaissance, permettent de générer des questions auxquelles l'étudiant est censé pouvoir répondre.

(ALSUBAIT et al., 2014) ont mis en pratique le principe défini ci-dessus pour générer automatiquement des questions pour un niveau de difficulté donné. Dans cette approche, la notion de difficulté est estimée par rapport à un niveau scolaire donné : les auteurs estiment que des questions techniques ou complexes peuvent être résolues facilement si elles sont adaptées au niveau scolaire des étudiants à qui elles sont posées. Pour un niveau scolaire donné, un étudiant doit pouvoir répondre facilement

aux questions inférieures ou égales à son niveau. Pour valider cette hypothèse, les auteurs du papier ont mis en place une évaluation de la difficulté relative des questions : les mêmes questions sont évaluées par des personnes de différents niveaux. Chaque question a été évaluée selon une échelle de 4 niveaux : (1) trop facile, (2) raisonnablement facile, (3) raisonnablement difficile et (4) trop difficile.

(SEYLER et al., 2017) ont défini une approche pour générer des questions pour un niveau de difficulté donné. Ils se basent pour cela sur des méthodes d'entraînement supervisées, en annotant manuellement des questions selon leur difficulté. Cette approche cherche à juger de la difficulté des questions générées principalement sur le contenu, et non sur la formulation de ces dernières. A ce stade, les questions ne sont caractérisées que par 2 niveaux de difficulté : facile (*easy*) et difficile (*hard*). Les évaluations menées par les auteurs de ce papier ont montré qu'il est difficile pour des personnes qui ne sont pas expertes dans un domaine de juger de la difficulté des questions, et ont ainsi recueilli une grande disparité entre les estimations au sein des questions générées.

(PANDEY et RAJESWARI, 2013) ont proposé une approche pour générer des questions correspondant à un niveau de difficulté prédéterminé en reposant sur la taxonomie de Bloom (BLOOM BENJAMIN et KRATHWOHL, 1956). Ils mettent en avant que la difficulté de chaque question est définie par les mots-clés qui composent son énoncé. La génération de question dans un niveau donné se traduit donc dans ce papier par un choix de mots-clés spécifiques à ce niveau.

2.3.3 Limites de l'estimation de la difficulté des questions à choix multiples

Les approches de génération automatique de questions à choix multiples s'accompagnent généralement d'une estimation de la difficulté des résultats obtenus. Cependant, nous avons vu dans les sections précédentes que cette estimation est subjective et dépend fortement des personnes interrogées. (SEYLER et al., 2017) ont montré que les estimations de difficulté réalisées par des humains donnent des résultats extrêmement différents en fonction du domaine ciblé, même si ces derniers ont un niveau d'éducation similaire.

(PANDEY et RAJESWARI, 2013) ont par ailleurs montré que la notion de difficulté n'est pas uniquement liée au contenu de l'énoncé et des distracteurs d'une question, mais également à leur formulation. Une même question peut ainsi être considérée par une même personne comme facile ou difficile suivant sa formulation.

Les travaux de (ALSUBAIT et al., 2013) ont de plus défini la notion de difficulté d'une question pour un niveau scolaire comme n'étant pas uniquement le résultat de connaissances brutes, mais de raisonnement autour du contexte de la question.

Ainsi, et pour toutes ces raisons, on peut affirmer que la principale limite de la gestion de la difficulté dans les mécanismes de génération automatique de questions provient du fait qu'il est extrêmement difficile de corréler des questions, que ce soit par leur contenu ou leur formulation, avec un niveau scolaire de référence.

On note également un biais récurrent de l'estimation de la difficulté induit par le fait que les questions générées sont souvent évaluées par des enseignants ou des chercheurs de niveaux similaires, et que par conséquent, l'estimation de la difficulté n'est valable que vis à vis des profils de ces personnes, mais ne peut relever d'une valeur absolue.

Bien que la gestion de difficulté soit centrale dans un processus de génération de questions à choix multiples, nous avons mis de côté cet aspect dans les travaux réalisés au cours de cette thèse. En effet, les approches existantes présentées ci-dessus reflètent efficacement la complexité liée à cette tâche, qui aurait nécessité des travaux spécifiques pour être intégrée au processus de génération de questions thématiques présenté dans ce document.

2.4 Évaluation des questions à choix multiples

L'analyse des résultats obtenus permet d'obtenir un retour sur la qualité du travail effectué. Cependant, dans le domaine de la génération de questions à choix multiples, l'analyse des résultats n'est pas une tâche triviale, et ce pour plusieurs raisons :

1. Une question à choix multiples se décompose en plusieurs éléments : l'énoncé, la bonne réponse, et les distracteurs. Chacun de ces composants doit faire l'objet d'une évaluation individuelle.
2. Une question à choix multiples est principalement évaluée sur sa qualité. La qualité étant une notion d'appréciation humaine, la mise en place d'un processus d'évaluation entièrement automatique est très difficile, voire impossible.
3. Certains éléments d'une question, comme la validité grammaticale de l'énoncé ou la véracité de la bonne réponse, relèvent de mesures objectives qui sont théoriquement identiques d'un évaluateur à l'autre. Cependant, d'autres éléments comme la connotation de l'énoncé avec un domaine, ou la cohérence des distracteurs avec la bonne réponse relèvent de mesures plus subjectives. Pour valider l'évaluation d'une même question, il est donc important que plusieurs personnes distinctes l'évaluent.

Cette section détaille les différentes solutions utilisées par les créateurs de questions à choix multiples pour évaluer la qualité des questions générées automatiquement. Les différents aspects de l'évaluation de ces questions, listés ci-dessus, sont ainsi analysés séparément : (i) l'évaluation individuelle des composants d'une question, (ii) les critères qui déterminent la qualité d'une question, et (iii) les conditions d'évaluation des questions.

2.4.1 Évaluation des composants des questions à choix multiples

Nombre d'articles de génération de questions à choix multiples évaluent globalement le résultat obtenu pour chaque question générée (Bailey et al., 1998)(Al-Yahya, 2014)(Alsubait et al., 2014). Cependant, dans le cas d'une appréciation mitigée, il est difficile d'identifier le ou les composants qui dégradent la qualité de la question. Cette évaluation est ainsi régulièrement complétée par des évaluations individuelles des composants de la question.

(Bailey et al., 1998) ont évalué indépendamment les énoncés et les réponses. Une évaluation globale de la question est ensuite réalisée. Chacun de ces 3 éléments est évalué selon plusieurs critères. La liste complète des critères est présentée par la figure 2.6. L'ensemble de ces critères permet d'évaluer précisément chaque compo-

sant de la question, en prenant en compte pour chacun sa validité grammaticale, linguistique, contextuelle, etc.

The stem	
1	deals with one central problem. [a, b]
2	has the to-be-completed phrase placed at the end. [a]
3	uses simple numbers to test a concept. [b]
The responses	
4	are grammatically consistent with the stem. [a, c, e, f]
5	do not unnecessarily repeat language from the stem. [a, f, g, h]
6	are all of approximately the same length. [a, c, e, f, h]
7	consist only of numbers that are plausible based on the data in the problem. [a, b, c, e, f, h]
8	avoid the use of "all of the above." [a, f, g]
9	avoid the use of "none of the above." [a, f, g]
10	contain one allowable answer rather than the use of "A and B," "A, B, and C," etc. [f, g]
The question overall	
11	avoids excessive verbiage. [a, e, f, g, h]
12	is grammatically correct throughout. [a, e, f, g, h,]
13	is clearly written. [a, e, f, g, h]
14	contains no negative or other counter-intuitive wording without underlining or other special emphasis. [c, d, e, g, h]
15	contains no other offensive or undesirable characteristics not listed above. [i]

Fig. 2.6.: Liste des règles utilisées pour évaluer les composants d'une question dans l'approche de (BAILEY et al., 1998)

.

(AL-YAHYA, 2014) ont réutilisé les critères définis par l'approche précédente, en les améliorant de façon à être plus adéquats avec la méthode utilisée, et moins contraignants pour l'évaluateur. Le nombre de critères a ainsi été réduit de 15 à 10. Cependant, les énoncés, les réponses et les questions en général restent évalués indépendamment.

(NARENDRA et al., 2013) ont également évalué indépendamment les énoncés, la bonne réponse et les distracteurs. Cependant, cette distinction s'est limitée à donner une appréciation globale à chacun de ces composants, sans rentrer dans un niveau de détails complexe comme les approches précédentes.

(PHO et al., 2015) ont proposé une méthode permettant d'évaluer automatiquement la crédibilité des distracteurs de questions à choix multiples. Cette approche compare la composition syntaxique de la bonne réponse avec celles des distracteurs, partant du principe qu'en cas de différences trop importantes entre les deux, les distracteurs ne seraient pas crédibles pour la personne interrogée. Cette méthode compare également la cohérence entre les distracteurs et la bonne réponse sur la base de la distance sémantique en utilisant plusieurs supports, dont DBpedia et WordNet.

Nous avons détaillé dans cette section les différents éléments qui composent les questions à choix multiples, et de quelle manière les différentes approches de la

littérature distinguent ces composants pour obtenir une évaluation plus précise des questions. Nous verrons dans la section suivante quels sont précisément les critères utilisés pour juger de la qualité d'une question, ou de ses composants.

2.4.2 Qualité des questions à choix multiples

L'évaluation de la qualité des questions à choix multiples est une tâche qui peut paraître objective à première vue, mais qui varie de façon pratique en fonction des éléments que les auteurs souhaitent valoriser. Il n'existe ainsi pas de critères d'évaluation communs à toutes les évaluations de questions à choix multiples qui pourraient être utilisés comme référence. Cette section présente les différents critères permettant de mesurer la qualité des questions générées par les différentes approches de génération de questions à choix multiples de l'état de l'art.

De nombreuses évaluations (Mitkov et al., 2009)(Afzal et Mitkov, 2014)(Liu et al., 2018)(Al-Yahya, 2014) se sont basées sur les critères initialement introduits par (Gronlund, 1982). L'auteur y établit que la qualité des questions à choix multiples se mesure selon trois critères principaux :

1. **La difficulté de la question :** Il s'agit de donner une valeur aussi objective que possible pour définir la difficulté de la question.
2. **Le pouvoir discriminant de la question :** Il s'agit d'identifier sa valeur ajoutée par rapport aux autres questions générées.
3. **L'utilité de chaque distracteur :** Il s'agit de déterminer de façon globale dans quelle mesure chacune des mauvaises réponses proposées est pertinente vis-à-vis de la question.

(Bhatia et al., 2013) ont réutilisé les critères présentés ci-dessus, tout en affinant les résultats. Ils ont ainsi proposé une amélioration de ces critères en ajoutant des mesures de pertinence plus spécifiques. Ces critères additionnels comprennent les éléments suivants :

— **La question est pertinente dans son domaine :** Ce critère ne s'applique que pour les cas où on cherche à obtenir des questions spécifiques à un domaine prédéterminé.

— **La qualité de la bonne réponse :** Regroupant les notions de validité et de pertinence par rapport à l'énoncé.

— **La qualité grammaticale de l'énoncé :** Ce critère est utilisé pour évaluer la qualité grammaticale de l'énoncé, donc les erreurs de langage et de syntaxe qu'il pourrait contenir.

— **La qualité informative de l'énoncé :** Contrairement au critère précédent, ce critère est utilisé pour juger de la complétude de l'énoncé. Ainsi, si ce dernier

contient trop d'informations (la bonne réponse ou des éléments trop explicites), ou trop peu (impossibilité technique de répondre à la question par manque d'informations), ce critère aura une évaluation négative.

— **Le lien des distracteurs avec la bonne réponse :** Critère qui consiste à valider la crédibilité individuelle des distracteurs.

(BHATIA et al., 2013) ont ensuite introduit plusieurs critères destinés à juger individuellement de la qualité des distracteurs. Ainsi, pour chacun d'entre eux, les critères suivants sont évalués : sa difficulté, son utilité, et sa capacité à convaincre en tant que bonne réponse.

(NARENDRA et al., 2013) distinguent deux critères pour évaluer la qualité d'un énoncé : la pertinence de l'énoncé, et sa valeur informative. Ils argumentent en effet que les énoncés générés peuvent posséder un de ces deux critères, sans posséder l'autre, mais que les deux sont requis pour obtenir un énoncé de qualité. Cependant, la qualité des distracteurs dans ce papier n'est mesurée que sur leur *utilité*, qui est considérée comme bonne si les distracteurs ne peuvent pas être éliminés avec des techniques simples.

(PINO et al., 2008) ont mis en place un arbre de décision, dont les options sont automatiquement proposées aux évaluateurs au travers d'une page web, afin d'évaluer la qualité des distracteurs. Cet arbre est présenté par la figure 2.7. On peut voir que les choix qui sont proposés aux évaluateurs dépendent des problèmes rencontrés avec les distracteurs, permettant de cibler au mieux les éléments qui doivent être améliorés.

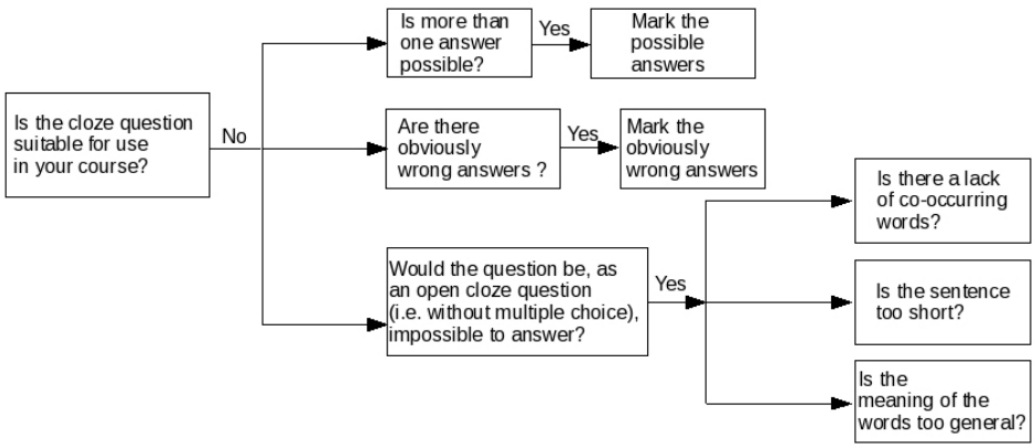


Fig. 2.7.: Arbre de décision permettant d'évaluer les distracteurs dans l'approche de (PINO et al., 2008)

.

(ALSUBAIT et al., 2014) ont proposé une approche originale en demandant aux utilisateurs de répondre à la question avant de l'évaluer. L'évaluation est ensuite réa-

lisée sur deux critères demandés séquentiellement aux évaluateurs : une évaluation globale, et une estimation de la difficulté.

(Seyler et al., 2017) ont principalement axé leur générateur de questions à choix multiples autour de la difficulté des questions générées. L'évaluation des questions a ainsi été principalement centrée autour de l'estimation de la difficulté de ces questions. Cela a permis de juger de l'efficacité du système à générer des questions pour un niveau scolaire donné.

(Papasalouros et al., 2008) ont mis en place une évaluation multidimensionnelle des questions générées. En effet, pour juger de la qualité des questions générées, elles sont parallèlement évaluées selon leur qualité pédagogique, et selon leur qualité linguistique.

Nous avons vu dans cette section les différents critères qui permettent aux évaluateurs de juger la qualité des questions à choix multiples. Bien que la formulation de ces critères tente de proposer une évaluation aussi objective que possible, l'appréciation dépend néanmoins fortement de l'évaluateur. Nous verrons ainsi dans la section suivante quelles sont les conditions d'évaluation mises en place pour refléter avec un maximum d'objectivité la qualité des questions à choix multiples générés, notamment en agrégeant et comparant les résultats de plusieurs évaluateurs.

2.4.3 Conditions d'évaluation des questions à choix multiples

La plupart des critères d'évaluation des questions à choix multiples sont subjectifs. Nous avons en effet vu dans la section précédente que la notion de qualité dépend en général de l'approche qui génère des questions à choix multiples, et que les critères qui déterminent cette qualité dépendent eux-mêmes du ressenti de l'utilisateur. Ainsi, il est nécessaire de mobiliser plusieurs personnes et d'évaluer plusieurs questions pour finalement comparer les évaluations obtenues. Cette section détaille les différentes conditions d'évaluation qui ont été mises en place pour juger de la qualité des questions générées.

(Mitkov et al., 2009) ont réalisé un système qui propose automatiquement des distracteurs pour des textes à trous. Ils ont ainsi testé différentes stratégies pour générer ces distracteurs, dont des stratégies basées sur la distance sémantique, les arbres syntaxiques, ou la phonétique. Pour valider la qualité des distracteurs obtenus avec ces stratégies, une évaluation à grande échelle a été réalisée. En effet, des distracteurs en accord avec chaque stratégie ont été parallèlement générés pour un ensemble de questions prédéterminées, et les ensembles ainsi générés ont été

répartis pour être évalués séparément. Au final, c'est 243 étudiants, provenant de 5 universités de 5 pays différents qui ont participé à l'évaluation de ces questions à choix multiples. La totalité des étudiants de l'évaluation suivaient un parcours littéraire, et maîtrisaient l'anglais. Chaque étudiant a évalué 20 questions en 30 minutes. En variant ainsi la provenance des évaluations et les pays des personnes évaluées (et donc les langues natives), tout en garantissant des conditions d'évaluation similaires, cette évaluation offre une comparaison détaillée des différentes stratégies proposées, et a permis ainsi de minimiser le caractère subjectif de l'évaluation humaine.

Certaines approches distinguent les évaluations réalisées par des natifs et des non-natifs de la langue évaluée (PINO et al., 2008) (CORREIA et al., 2010). Ce type d'évaluation est notamment utilisé lorsque les personnes interrogées doivent évaluer des distracteurs générés sur une base phonétique, ou par distance sémantique avec des mots similaires. Cependant, ici, seul (CORREIA et al., 2010) a mis en place un test à grande échelle, en faisant évaluer la qualité des distracteurs générés par plus de 200 natifs, et 35 non-natifs de la langue des questions. Cette comparaison entre natifs et non-natifs permet de comparer l'intérêt de distracteurs générés notamment sur des bases grammaticales ou phonétiques auprès de personnes qui ne sont pas forcément à l'aise avec toutes les subtilités d'un langage.

(BAILEY et al., 1998) ont réalisé une évaluation en deux étapes. Un test pilote a été réalisé sur une vingtaine de questions, avec 3 juges différents. L'objectif de cette première évaluation était d'ajuster les paramètres du générateur pour augmenter la qualité des questions générées. Un test plus important a ensuite été réalisé sur 200 questions par 2 juges différents ne possédant pas de lien avec le projet, et rémunérés pour l'opération, de façon à obtenir des résultats aussi objectifs que possible.

Cependant, compte tenu de la difficulté à mettre en place une évaluation à grande échelle, un grand nombre d'approches n'évaluent qu'un petit nombre de questions (une vingtaine), et ne sollicitent généralement qu'un petit nombre d'évaluateurs (entre 3 et 5) : (NARENDRA et al., 2013)(PAPASALOUROS et al., 2008)(AL-YAHYA, 2014)(ALSUBAIT et al., 2014)(PINO et al., 2008). Toutefois, on remarque que dans la totalité de la littérature, l'évaluation manuelle des questions à choix multiples est systématiquement réalisée par plusieurs juges dans l'optique de minimiser le caractère subjectif de l'évaluation.

2.4.4 Limites de l'évaluation des questions à choix multiples

Nous avons vu dans les sections précédentes qu'une évaluation entièrement automatisée des questions à choix multiples est extrêmement compliquée, voire impossible. Ces évaluations doivent ainsi être réalisées manuellement. Nous avons également

vu qu'une évaluation globale ne suffit pas pour déterminer si la question générée est de qualité. Les composants doivent être évalués indépendamment. Enfin, nous avons vu qu'une évaluation unique n'était pas suffisante, dans la mesure où certains critères d'évaluation sont subjectifs.

Ainsi, la principale limite des évaluations de questions à choix multiples réside dans le fait qu'il s'agit d'un processus extrêmement compliqué à mettre en place. Il faut en effet évaluer un nombre suffisamment représentatif de questions pour juger de l'efficacité du système de génération, chacune des questions doit être évaluée par plusieurs juges, et ce sur un grand nombre de critères. Nombre d'approches font effectivement état de plus de 10 critères différents qui doivent être évalués pour chaque question.

Une autre limite de l'évaluation de questions à choix multiples réside également dans le fait qu'il n'existe pas réellement de standard pour évaluer les questions. En effet, nous avons vu dans les approches précédentes que les auteurs définissent souvent eux-mêmes les critères qui serviront à évaluer leurs systèmes. Cette absence de standard complique ainsi la comparaison des résultats obtenus avec les travaux existants de l'état de l'art.

Pour être en adéquation avec ce qui a été présenté dans cette section, et ainsi réduire le caractère objectif de l'appréciation humaine, tout en offrant une visibilité sur la qualité des différents composants des questions générées, l'évaluation de la solution que nous proposons doit nécessairement respecter un ensemble de critères :

— Évaluer un nombre de question significatif, permettant de refléter les résultats de la méthode de façon générale,

— Évaluer indépendamment les composants de chaque question, plutôt que la question de façon générale,

— Comparer les avis de plusieurs évaluateurs.

L'évaluation que nous avons mise en place respecte l'ensemble de ces critères. Les modalités d'évaluation, ainsi que les résultats obtenus sont présentés dans le chapitre 6.

2.5 Génération en langage naturel

On regroupe sous le terme de *génération en langage naturel* (ou *Natural Language Génération (NLG)* en anglais) l'ensemble des opérations visant à créer automatiquement une phrase intelligible par l'homme à partir de données numériques. Différents aspects sont ainsi regroupés sous cette appellation, dont notamment la gestion du temps de la phrase, de sa ponctuation, de sa forme (active ou passive), etc. En somme, doivent être prises en compte toutes les règles élémentaires de grammaire et de conjugaison qui permettent de former des phrases correctes.

(GATT et KRAHMER, 2018) proposent un résumé très détaillé des différentes approches qui composent ce domaine. De nombreux aspects de la génération en langage naturel y sont abordés, notamment la traduction, la simplification ou la correction orthographique automatique, mais nous nous contenterons de reprendre ici les approches centrées autour de la génération de phrases et de la génération de questions.

Nous détaillons ainsi dans la première partie de cette section les approches et les procédés de NLG centrés autour de la génération de phrases. Nous détaillons dans une seconde partie les approches spécifiques à la génération d'énoncés de questions, et dans une troisième partie nous présentons les approches se basant sur des modèles prédéfinis pour la génération de texte.

2.5.1 Génération automatique de phrases

La génération en langage naturel consistant à former une phrase intelligible par un humain à partir d'une source de données n'est pas considérée comme une tâche unique, mais comme un ensemble de sous-tâches qui doivent être résolues séquentiellement pour obtenir un résultat satisfaisant. (REITER et DALE, 2000) identifient ainsi 6 problèmes qui seront repris dans de nombreuses approches de NLG :

1. **Déterminer le contenu :** Déterminer quelles informations devront apparaître dans le texte qui sera généré.
2. **Structurer le texte :** Déterminer dans quel ordre les informations devront apparaître.
3. **Décomposer et organiser le texte en phrases :** Déterminer la répartition des informations sous forme de phrase dans l'optique d'organiser le texte généré.
4. **Choix du vocabulaire :** Trouver le vocabulaire adéquat pour exprimer les idées souhaitées au sein du texte.

5. **Intégration dans le contexte :** Respecter le vocabulaire contextuel lié au domaine du texte.
6. **Réalisation :** Combiner les mots pour former des phrases correctes.

Ces tâches font l'objet de nombreux travaux, séparément ou non, que nous ne détaillerons pas ici.

(GATT et REITER, 2009) ont mis en place en 2009, et font évoluer progressivement depuis une API de génération automatique de phrases nommée *simpleNLG* [1]. Cette API permet notamment de modifier les composants linguistiques d'une phrase, comme le temps, la forme passive/active, la forme affirmative/interrogative, etc. Cette API, développée en Java est extrêmement complète dans les possibilités qu'elle offre, mais demande une connaissance très approfondie des règles de grammaire pour être utilisée correctement.

De plus en plus d'approches utilisent des réseaux de neurones pour générer du texte en langage naturel (WISEMAN et al., 2018)(HU et al., 2017)(SHEN et al., 2017). Certaines approches se spécialisent particulièrement sur la tâche de génération de questions à partir de ces réseaux de neurones (DU et al., 2017)(ZHOU et al., 2017). Cependant les réseaux de neurones, bien que relativement efficaces pour cette tâche, ne permettent pas de cibler un thème particulier lors de la génération en langage naturelle. C'est pourquoi nous avons écarté leur utilisation dans le cadre de cette thèse.

2.5.2 Génération automatique d'énoncés de questions

La génération automatique de questions a fait l'objet d'un résumé présenté par (PIWEK et BOYER, 2012). Les différentes méthodes utilisées pour générer des questions, et entre autres leurs énoncés, y sont ainsi abordées. On trouve notamment les approches basées sur du texte, dont l'énoncé peut être créé en inversant certains mots à partir d'une phrase déclarative (YAO et al., 2012). On y trouve également l'approche de (BERNHARD et al., 2012) basée sur l'analyse de la syntaxe d'un texte et la reconnaissance d'entités, afin de proposer un énoncé en reformulant des phrases et en y substituant des éléments afin de former des questions.

Les approches de génération automatique basées sur du texte ont la possibilité d'utiliser les éléments syntaxiques de ce dernier pour formuler des énoncés de question. Dans certains cas, on peut même générer des énoncés en retirant des mots-clés de phrases afin de proposer des textes à trous (BECKER et al., 2012)(NARENDRA et al., 2013)(GUO et al., 2016). Ces approches sont cependant difficilement applicables

1. `https://github.com/simplenlg/simplenlg`

à la génération de questions à partir de bases de connaissances, qui ne contiennent pas d'informations linguistiques réutilisables ou reformulables. Afin de contourner ce problème, ces approches utilisent un système de modèles que nous détaillons dans la prochaine section.

2.5.3 Approches basées sur des modèles

Dans le cadre de la génération en langage naturel, on désigne sous l'appellation de *modèle* le fait de contraindre tout ou partie de la génération à l'aide d'éléments prédéfinis. Ce procédé offre ainsi l'assurance que la forme ou l'orthographe des morceaux prédéfinis sera valide. Ces modèles sont également un moyen d'assurer une quantité d'information minimale, particulièrement dans le contexte de la génération automatique d'énoncés. Nous verrons ainsi dans cette section les différents usages qui sont faits des modèles dans le domaine de la génération automatique de phrases et de questions en langage naturel.

(Liu et al., 2012) utilisent un ensemble prédéfini de 6 modèles de phrases qui leur permettent de concevoir à l'avance la question qui sera posée en fonction de la situation, en substituant le sujet au moment de la génération.

(Olney et al., 2012) et (Seyler et al., 2017) utilisent des modèles pour générer des questions à partir de bases de connaissances. Les modèles sont utilisés pour convertir les relations de la base en morceaux de phrases prédéterminés, auxquels seront ajoutés les sujets et objets lors de la génération. Chaque relation utilisée pour générer les questions doit ainsi faire l'objet d'un mapping manuel.

(Deemter et al., 2005) comparent les approches traditionnelles de génération en langage naturel avec les approches utilisant des modèles. La conclusion de ce papier met en contraste trois points majeurs qui distinguent les approches basées sur les modèles de celles qui ne le sont pas :

— **La couverture est moins élevée :** En effet, seuls les éléments qui sont compatibles avec les modèles prédéfinis peuvent être pris en compte. Le champ des possibilité est donc nécessairement restreint.

— **Les phrases sont plus précises :** Les modèles offrent l'avantage de limiter l'information contenue dans la phrase. On y présente uniquement les informations souhaitée, limitant le risque d'ambiguité.

— **les phrases sont plus correctes :** La partie que l'on peut considérer comme *difficile* de la phrase est déterminée à l'avance. La gestion de la conjugaison, de la forme de la phrase, des temps, etc. qui constituent les tâches complexes de la génération en langage naturel, est ainsi nécessairement simplifiée.

Il est important de noter que les auteurs de ce papier ne dissocient pas l'utilisation des modèles des techniques de génération en langage naturel traditionnelles. En effet, les modèles ne conditionnent pas nécessairement la totalité des phrases générées, et doivent parfois être complétés par des procédés de génération tels que la gestion du temps, des accords, etc.

(CURTO et al., 2012) ont généré des énoncés de questions en se basant sur des agent capables de générer automatiquement des modèles à partir de paires [question/réponses] issues du domaine de la réponse automatique aux questions. Les éléments ainsi extraits sont ensuite réutilisés pour créer des questions à choix multiples.

En l'absence d'autres possibilités, on remarque toutefois que les approches se basant exclusivement sur des bases de connaissances pour générer des questions utilisent généralement des modèles pour produire leurs énoncés. En effet, les bases de connaissances ne contiennent nativement pas suffisamment d'informations pour déduire la sémantique présente dans les relations et s'en servir pour former des phrases.

2.5.4 Limites des approches de génération en langage naturel

Dans le cadre de la génération de questions, la génération en langage naturel des énoncés est une phase centrale du processus. En effet, les énoncés constituent l'information centrale affichée à la personne interrogée. Il est donc indispensable que les énoncés soient clairs et grammaticalement corrects, de façon à lever toute ambiguité quant à leur signification.

Nous avons expliqué ci-dessus pourquoi nous écartons dans cette thèse l'utilisation des réseaux de neurones pour la génération d'énoncés thématiques. Les autres possibilités ne sont cependant pas nécessairement satisfaisantes pour autant. En effet, nous ne disposons pas de support de texte nous permettant de réutiliser les travaux de génération de questions basés sur des documents. Ces méthodes ne sont donc pas non plus applicables dans notre cas.

Par ailleurs, on constate que les méthodes de génération de questions qui utilisent des bases de connaissances comme sources de données font systématiquement usage de modèles pour générer l'énoncé. Cette notion de modèle, bien que nécessaire, n'est à l'heure actuelle pas très développée. En effet, l'ensemble des relations de la base doivent généralement être traduites en portions de phrases, qui seront assemblées pour former les énoncés des questions. L'alternative réside dans la mise en place de

quelques templates très génériques, alternativement utilisés en fonction du type des données utilisées pour former l'énoncé.

Ainsi, pour toutes les raisons énumérées ci-dessus, nous proposons une solution basée sur des modèles de questions liés à la verbalisation d'une question plutôt qu'à celle d'une base de connaissances. Ces modèles sont définis dans le chapitre 4 de ce manuscrit.

2.6 Identification automatique de thèmes et classement d'entités

On considère dans cette section les approches existantes dans les domaines de l'identification de thèmes et de classement d'entités.

Nous définirons ainsi dans un premier temps **l'identification automatique de thèmes** comme étant "*la tâche capable de déterminer automatiquement, au sein d'une source de données déterminée (texte, portion de base de connaissances, etc.) le domaine/thème des entités qui la compose*". La tâche qui nous importe plus précisément ici est l'identification et la délimitation de thèmes au sein des bases de connaissances.

Nous compléterons cette définition avec celle du **classement d'entités** comme étant "*la tâche consistant à déterminer l'importance relative de chaque entité au sein d'un ensemble*". Cette tâche résulte donc en la transformation d'un ensemble désordonné d'entités en une liste contenant les mêmes entités, et où toute entité à un rang i est plus importante que les entités aux rangs supérieur à i.

Nous verrons ainsi dans cette section les approches de l'état de l'art permettant l'identification de thèmes ou le classement d'entités, et finalement de quelle façon leur combinaison peut mener au *classement d'entités par thèmes*.

2.6.1 Identification automatique de thèmes

La tâche d'identification automatique de thèmes est particulièrement présente dans l'analyse de documents textuels. L'objectif est de déterminer le ou les sujets traités par un document. L'approche principale dans l'état de l'art est celle des modèles de sujet ou *topic model* en anglais. Ces méthodes utilisent des approches probabilistes pour déterminer des thèmes présents dans des collections de documents. Cette approche *topic model* est donc une approche *bottom-up* qui part d'une collection de documents pour en déterminer les thèmes "cachés". Cependant ces approches ne peuvent pas fonctionner de manière inverse, c'est-à-dire qu'elles ne peuvent pas déterminer pour un ensemble de thèmes donnés et définis, l'appartenance de chaque document de la collection à ces thèmes.

(Blei et al., 2003) ont introduit LDA (*Latent Dirichlet Allocation*), une mesure permettant de classer chacun des éléments d'un ensemble dans K groupes d'éléments, K étant fixé a priori. L'hypothèse fondatrice de LDA est que la distribution des thèmes possède une distribution de Dirichlet. La phase d'apprentissage est de type bayésien,

sachant que différents types d'inférence peuvent être utilisés (*Gibbs Sampling*, *Expectation Propagation*). LDA est donc particulièrement utile pour la tâche consistant à faire émerger des thèmes d'une collection de documents.

La *Latent Semantic Analysis* ou LSA est une approche visant à analyser la relation entre une collection de documents et un ensemble de concepts. Cette approche se base sur la décomposition en valeurs singulières de la matrice mot/document. De la même manière que pour LDA, le nombre de thèmes à identifier est passé en entrée de l'algorithme.

D'autres approches utilisent des bases de connaissances comme apport externe pour identifier le thème d'un document, en le reliant à des entités de cette base.

(MEDELYAN et al., 2008) utilisent Wikipedia pour identifier automatiquement le contexte d'un document. Ils procèdent pour cela en deux étapes : (i) identifier au sein d'un document les mots et les phrases importants, et (ii) identifier au sein de Wikipedia les articles qui utilisent ces mots et ces phrases. Cette approche permet ainsi d'identifier les articles liés aux mots présents dans le document, ce qui permet de donner un sens aux mots ambigus de l'article.

(HULPUS et al., 2013) se basent sur DBpedia pour déterminer automatiquement les thèmes d'un document. Ils procèdent pour cela à une phase d'analyse de DBpedia, au cours de laquelle ils identifient des ensembles d'entités aux thématiques communes, qui constitueront les thèmes. Ces ensembles sont ensuite comparés avec le contenu du document afin de déterminer les différents thèmes qui le composent.

Ces approches sont cependant axées sur l'identification du thème au sein d'un document. Elles nécessitent donc d'avoir une source de données, dans l'optique de l'analyser et de déterminer son sens. Ces solutions ne sont ainsi pas réutilisables dans notre problématique, dont l'objectif est opposé : trouver des entités liées à un thème connu à priori.

2.6.2 Classement d'entités

Classer les entités d'un ensemble prédéterminé est une tâche qui offre de nombreuses applications, comme notamment identifier les concepts essentiels d'un document, d'une phrase, ou d'une source de données en général. On trouve ainsi un certain nombre d'approches visant à utiliser les bases de connaissances, et tout particulièrement Wikipedia, afin de déterminer le classement de ces entités.

L'algorithme du *Pagerank*, introduit par (BRIN et PAGE, 1998), a été initialement mis en place pour classer des pages Web en fonction de leur popularité. Le fonctionnement de l'algorithme du Pagerank s'illustre par l'image d'un utilisateur cliquant aléatoirement sur un des liens présents dans une page Web. Il se déplace ainsi sur une autre page Web, et recommence l'opération. En recommençant cette navigation un grand nombre de fois, on peut ainsi identifier les pages Web les plus référencées. Le principe de cet algorithme, bien que conçu pour le référencement de pages Web, est applicable à tout graphe orienté, et permet de déterminer au sein de ces graphes l'importance relative des différents noeuds qui les composent.

(ZHIROV et al., 2010) comparent l'algorithme du CheiRank (CHEPELIANSKII, 2010) avec celui du Pagerank (LANGVILLE et MEYER, 2011) pour classer les entités de Wikipedia. Ils appliquent ces algorithmes à des sous-domaines précis afin de déterminer manuellement leur efficacité. Ils introduisent également *2DRank*, une mesure de classement qui combine les deux approches précédentes.

(KAPTEIN et al., 2010) déportent le problème de classer les entités du Web à celui de classer les entités de Wikipedia. En effet, la plupart des éléments ayant un article correspondant dans l'encyclopédie, le classement des entités internes à Wikipedia est étendu à des applications externes à cette dernière.

2.6.3 Classement d'entités au sein des thèmes

Le classement d'entités, de façon absolue, peut se révéler utile pour déterminer les points importants d'un document. On travaille alors avec un ensemble restreint d'entités, et on souhaite déterminer parmi elles les plus importantes. Cependant, quel que soit le document ou la source de données, les entités jugées comme importantes sont les mêmes à chaque fois. Afin de prendre en compte la notion de contexte, certaines approches ne calculent pas une valeur absolue de classement, mais un ensemble de valeurs qui dépendent également du thème considéré.

(COURSEY et al., 2009) ont combiné classement d'entité et extraction automatique de thèmes pour identifier automatiquement le thème d'un document passé en paramètre en utilisant Wikipedia. Pour cela ils mettent en place une étape de prétraitement au cours de laquelle ils construisent un graphe composé des articles et catégories de Wikipedia, les nœuds représentant les différents articles et catégories, et les arêtes sont les liens entre ces derniers. Ensuite, pour chaque document testé, un score est attribué à chacun des nœuds du graphe en fonction des mots-clés présents dans le document (identifiés à l'aide de Wikify (MIHALCEA et CSOMAI, 2007)). Comme les mots-clés correspondent à des articles de Wikipedia, et donc à des nœuds du graphe, les auteurs ont mis en place un algorithme de Pagerank pondéré, prenant en compte

le score du noeud calculé précédemment. Ce Pagerank permet ainsi d'identifier un ensemble d'articles et de ressources qui, bien qu'ils ne soient pas explicitement mentionnés dans le texte du document, ont un rapport élevé avec ce dernier.

(BOUGOUIN et al., 2013) proposent *TopicRank*, une approche permettant d'identifier automatiquement le thème d'un document en créant un graphe basé sur les mots-clés qu'il contient. Les auteurs utilisent ensuite TextRank (MIHALCEA et TARAU, 2004) pour déterminer le thème du document en fonction des différents mots-clés identifiés.

2.6.4 Limites des approches d'identification automatique de thèmes

A l'heure actuelle, les approches liées à l'identification automatique de thèmes sont majoritairement utilisées pour la tâche consistant à déterminer le ou les thèmes d'un document. Cela signifie que la notion de thèmes est difficilement reliable à des référentiels fixes d'éléments connus a priori, qui pourraient être utilisés de façon transversale dans un contexte plus large. L'identification automatique de thèmes au sein d'une base de connaissances, ainsi que le classement de ses entités parmi les thèmes reste donc une question de recherche ouverte pour laquelle il n'existe pas aujourd'hui, à notre connaissance, de solutions.

2.7 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art sur les différentes questions relatives à la génération automatique de questions thématiques à partir de base de connaissances.

Tout d'abord, concernant l'identification de thématiques au sein des bases de connaissances, l'analyse de l'état de l'art, présentée en section 2.6, a montré que ce problème en tant que tel n'a pas été traité par la communauté. Les techniques classiques de *topic detection*, type LDA ou LSA, ne permettent pas d'obtenir de résultat satisfaisant sur une base de connaissances, puisque le contenu textuel d'une base de connaissances est très limité. De plus le nombre de thèmes devant être fixé comme paramètre d'entrée de l'algorithme, ceci constitue une limitation importante des approches *bottom-up*.

Ce problème d'identification de thèmes au sein d'une base de connaissances reste donc une question de recherche ouverte. Cette question est particulièrement intéressante puisque sa portée va au-delà de notre cas d'application. La capacité à organiser une base de connaissances par thèmes ouvre la porte à de nombreuses applications, notamment autour de l'exploration des bases de connaissances, ainsi que sur leur utilisation comme apport externe dans des solutions d'identification de thèmes.

L'étape suivante dans la génération de questions consiste à identifier des faits qui serviront à la génération de questions en langage naturel. Sur cette question, notre analyse de l'état de l'art (Section 2.1) a montré que les approches actuelles sont limitées quant à la détection efficace des liens autour d'une identité et nécessitent des étapes manuelles qui sont lourdes et coûteuses en temps. Cette étape reste donc une question ouverte et centrale dans notre approche.

La génération des distracteurs pour les questions issues de bases de connaissances ne peut pas non plus être considéré comme un problème résolu. En effet, notre étude de l'état de l'art (section 2.2) a montré une limite récurrente liée au contexte des questions générées, qui rend les distracteurs faciles à écarter, même s'ils sont sémantiquement proches. Nous pensons que l'introduction de la notion de thèmes pourrait permettre d'adresser ce problème en proposant des distracteurs plus en rapport avec la réponse correcte.

La génération de question en langage naturel, en dépit des avancées récentes en apprentissage profond, reste un problème traité avec une qualité très élevée en utilisant des approches par modèles. La problématique principale de l'approche par modèles est la création de ces derniers. En effet les approches manuelles sont

particulièrement fastidieuses. Un question intéressante serait d'étudier la génération semi-automatique ou automatique de modèles pour combiner une approche à forte précision avec un large volume de questions différentes.

Ainsi, nous pouvons déduire de cet état de l'art et des limites des différents travaux présentés que la mise en place d'un générateur de questions thématiques soulève à l'heure actuelle un ensemble de problématiques qui doivent être résolues individuellement pour obtenir un résultat fonctionnel de bout-en-bout :

1. Introduire la notion de thème dans les bases de connaissances
2. Identifier au sein d'une base de connaissances des faits composés d'entités et de relations respectant la contrainte de complétude pour générer des questions.
3. Générer des énoncés à partir des faits identifiés précédemment.
4. Associer à la question générée un ensemble de distracteurs crédibles.

La section suivante de ce manuscrit présente les méthodes et contributions que nous avons mises en place pour résoudre ces différentes problématiques. Nous y présentons également la solution utilisée pour transformer ces contributions isolées en un générateur intégré et fonctionnel de questions à choix multiples thématiques.

3

La notion de thème dans les bases de connaissances

Introduction

Dans le cadre de cette thèse, nous avons pour objectif de construire automatiquement des questions à choix multiples en rapport avec un thème donné à partir de bases de connaissances. Ainsi, l'énoncé de la question et ses distracteurs doivent tous deux être cohérents avec ce thème. Se pose ainsi en premier lieu le problème de l'identification de données thématiques au sein des sources de données, et tout particulièrement des bases de connaissances. Nous avons ainsi mis en place une solution qui soit capable, de façon autonome, de délimiter le périmètre d'un thème. Cette opération inclut le fait d'identifier un ensemble d'éléments en rapport avec un thème, et de déterminer pour chacun d'entre eux dans quelle mesure il est pertinent vis-à-vis du thème.

Nous avons vu dans notre état de l'art que les bases de connaissances sont particulièrement avantageuses pour générer des questions à choix multiples, gràce la richesse des informations qu'elles contiennent. Cependant, nous avons également vu que les bases de connaissances ne contiennent pas d'informations permettant d'associer les entités qui y sont présentes avec des thèmes. Par ailleurs, les différentes méthodes existantes de recherche de thèmes que nous avons présentées dans la section 2.6 n'apportent pas de solutions répondant à ce besoin. En effet, les tâches de recherche de thèmes, comme LDA (Blei et al., 2003), y sont utilisées pour déterminer le thème d'un document, ou d'un ensemble de mots. Cette opération est réalisée en déterminant un ensemble de thèmes et de scores spécifiques à chaque document analysé.

Au contraire, la tâche de génération automatique de questions à choix multiples thématiques nécessite que les thèmes soient connus a priori, afin qu'ils soient utilisables comme référentiels pour la sélection des entités des questions. Nous expliquons ainsi dans cette section la méthode utilisée pour introduire la notion de thème dans des bases de connaissances, et tout particulièrement, dans les bases de connaissances dérivées de Wikipedia.

Ainsi, pour expliquer les travaux que nous avons réalisés pour introduire des thèmes dans les bases de connaissances, ce chapitre s'organise de la façon suivante. Nous présentons dans la section 3.1 les bases de connaissances dérivées de Wikipedia et dans la section 3.2 les caractéristiques de Wikipedia. Nous expliquons ensuite dans les sections 3.3 et 3.4 les différentes méthodes d'extraction utilisées pour parcourir Wikipedia et utiliser ses méta-données pour créer et approvisionner un ensemble de thèmes, et pour trier les éléments qui s'y rattachent. La section 3.5 présente ensuite quelques éléments techniques liés à la mise en place des thèmes, et nous terminons ce chapitre avec la section 3.6 en présentant Fouilla, un cas d'utilisation lié aux thèmes.

3.1 Bases de connaissances dérivées de Wikipedia

En se basant sur la littérature du domaine, nous définissons une base de connaissances de la façon suivante :

Définition 1: Base de Connaissances

Une *Base de Connaissances* est une base de données stockée sous la forme d'un graphe, où les nœuds représentent des entités ou des littéraux, et sont reliés entre eux à travers des arêtes, qui représentent les relations (KRISHNA, 1992).

A travers sa structure et le contenu de ses articles, Wikipedia contient un ensemble de données formatées ou non, extrêmement riches et variées. Fortes de ce constat, de nombreuses approches ont utilisé le contenu de cette encyclopédie pour créer et alimenter des bases de connaissances. Parmi les plus connues, nous listons les suivantes :

— **DBpedia :** Initialement lancé en tant que projet universitaire, DBpedia [1] a été ouvert au public en 2007 (AUER et al., 2007). Chaque page de Wikipedia contient ainsi un équivalent sur DBPedia. Les pages de DBpedia regroupent des données factuelles, stockées sous forme de triplets RDF, directement extraits depuis les pages Wikipedia équivalentes. On y retrouve ainsi par exemple les dates de naissance et de décès des personnes, la superficie et la population des pays, etc. Les entités de DBpedia sont couplées à une ontologie permettant de catégoriser et décrire sémantiquement les données contenues dans cette base de connaissances.

— **Wikidata :** Ouverte au public fin 2012, Wikidata [2] est une base de connaissances initialement construite à partir des données de Wikipedia (VRANDECIC et KRÖTZSCH, 2014). Elle est depuis ouverte aux améliorations, tant par des humains que par des machines. Additionnellement aux liens internes entre les entités de la base, qui sont propres aux bases de connaissances, Wikidata se distingue en proposant également un grand nombre de références à des identifiants de bases de données externes. Pour un acteur par exemple, on trouve notamment son identifiant IMDB ou Allociné.

— **Yago :** lancée en 2008 par le Max Planck Institute (MPI), YAGO [3] est une base de connaissances construite en combinant Wikipedia avec d'autres sources de données (SUCHANEK et al., 2007). Les données de Yago sont décrites à

1. https://dbpedia.org/
2. https://www.wikidata.org
3. https://www.mpi-inf.mpg.de/yago-naga/yago/

travers la combinaison de deux ontologies : celle de DBpedia, et celle de SUMO [4] (PEASE et al., 2002).

— **Freebase :** Ouverte au public en 2007, Freebase [5] est une base de connaissances collaborative initialement construite à partir de Wikipedia (BOLLACKER et al., 2008). Le but est de permettre à des personnes de tous horizons d'ajouter et modifier les données. Freebase possède ainsi sa propre ontologie pour décrire sémantiquement ses données. Cependant, Google a racheté la société qui gérait Freebase en 2010, et a mis fin à l'API de Freebase fin 2014. Certains dumps datant de cette année restent cependant disponibles pour une utilisation libre.

Étant dérivées de Wikipedia, toutes ces bases de connaissances offrent un ensemble d'entités correspondant aux articles de Wikipedia, avec une correspondance théorique de 1:1 (illustrée par la figure 3.1). On parle ici de correspondance théorique, dans la mesure où ces bases peuvent être enrichies à l'aide d'entités provenant de sources externes à Wikipedia. On retrouve quoi qu'il en soit la totalité des articles de Wikipedia dans chacune d'entre elles.

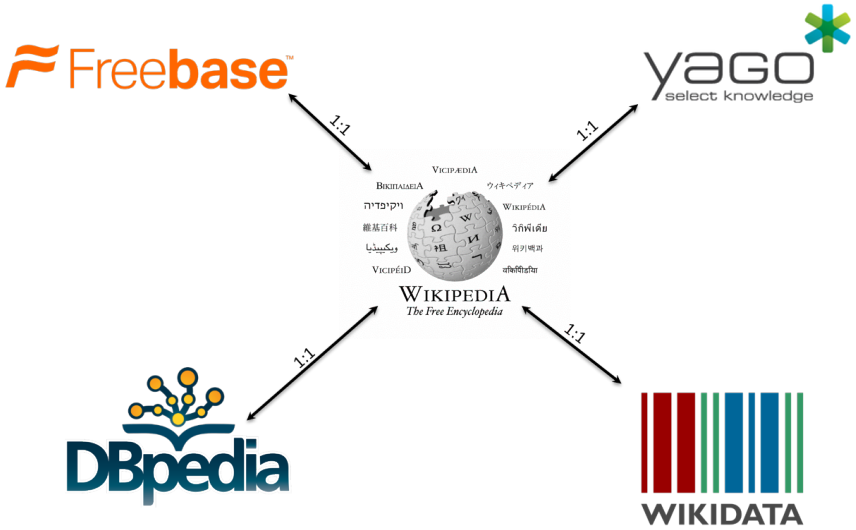


Fig. 3.1.: Wikipedia et les différentes bases de connaissances qui en sont dérivées.

Cette correspondance 1:1 signifie que pour tout article de Wikipedia donné, il existe une entité correspondante dans chacune de ces bases de connaissances. Nous avons donc émis l'hypothèse que le problème consistant à introduire la notion de thèmes au sein des bases de connaissances dérivées de Wikipedia peut être ramené à celui de définir un ensemble de thèmes à partir des données de Wikipedia.

En effet, Wikipedia est composée d'articles, qui constituent la partie visible par les utilisateurs, mais elle ne se limite pas à ces articles. La valeur ajoutée de Wikipedia

4. http://www.adampease.org/OP/
5. https://en.wikipedia.org/wiki/Freebase

repose plutôt sur les nombreuses méta-données qui forment sa structure. Ainsi, on y trouve entre autres un système de catégories hiérarchiques qui permettent de regrouper des articles ayant de sujets communs.

Dans ces travaux, nous utilisons les méta-données de Wikipedia afin de définir automatiquement un ensemble de thèmes pouvant être utilisé dans des bases de connaissances. Cette tâche consiste donc à identifier un ensemble de thèmes, leur associer des articles, et calculer l'importance relative des articles au sein de chaque thème.

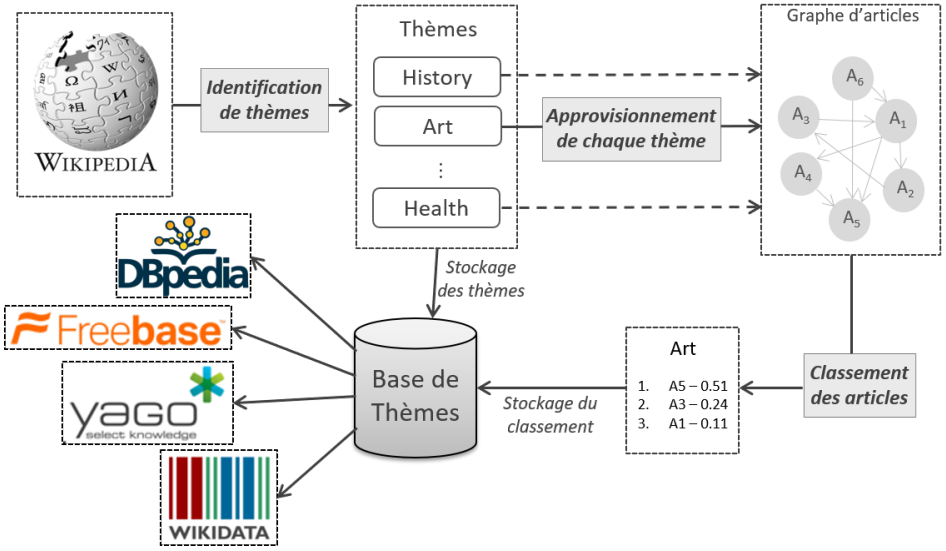


Fig. 3.2.: Vue d'ensemble de la solution d'identification et création des thèmes à partir des méta-données de Wikipedia

La figure 3.2 présente une vue d'ensemble de la solution proposée pour identifier et approvisionner automatiquement des thèmes en se basant sur les méta-données présentes dans Wikipedia. Cette solution est composée de trois étapes sur lesquels nous revenons en détail dans la suite de cette section : 1. Identification des thèmes ; 2. Approvisionnement des ensembles thématiques ; 3. Classement des articles. De façon plus approfondie, on peut résumer ces trois étapes de la façon suivante :

1. **Identification des thèmes** (sec. 3.3) : En se servant des méta-données navigationelles de Wikipedia, nous identifions automatiquement un ensemble de thèmes. Ces thèmes sont stockés dans une base de données.

2. **Approvisionnement des ensembles thématiques** (sec. 3.4 et 3.4.3) : A partir de notre ensemble de thèmes préalablement définis, nous utilisons la structure de Wikipedia, ainsi que ses méta-données pour associer à chaque thème un ensemble d'articles. Dans un second temps, une mesure de similarité est utilisée pour compléter cet ensemble avec des articles au contenu similaire qui n'auraient pas été identifiés uniquement à l'aide de la structure.

3. **Classement des articles** (sec. 3.4.2) : A l'aide de l'algorithme Pagerank, nous calculons l'importance relative des articles de chaque thème et obtenons ainsi un classement des articles dans chaque thème.

Cette approche repose principalement sur l'exploitation de la structure et des méta-données de Wikipedia. C'est pourquoi, avant d'expliquer en détail les différentes étapes de la solution que nous avons mise en place, nous détaillons dans la section suivante les différents composants de Wikipedia.

3.2 Structure et méta-données de Wikipédia

Wikipedia est une encyclopédie en ligne permettant à ses utilisateurs de rechercher et consulter des articles. A ce titre, Wikipedia contient un certain nombre de méta-données qui permettent aux utilisateurs de trouver rapidement et facilement l'information souhaitée.

Dans notre contexte de génération de questions relatives à des thèmes donnés, nous avons identifié au sein de Wikipedia un sous-ensemble de méta-données nous permettant de mettre en place un processus capable à la fois d'extraire automatiquement une liste de thèmes, et d'identifier et classer des articles de Wikipedia pertinents pour chacun d'entre eux.

Ces méta-données sont présentes avant tout dans la structure de l'encyclopédie. En effet, les articles (i.e. les pages) contiennent un ensemble de données non formatées (texte, images, schémas), ainsi que des liens menant à d'autres articles de l'encyclopédie. Ces liens constituent ce que l'on décrit comme étant le **graphe des articles** de Wikipedia. Ce graphe est orienté, dans la mesure où un article peut en référencer un autre sans être nécessairement lui même référencé par ce premier article.

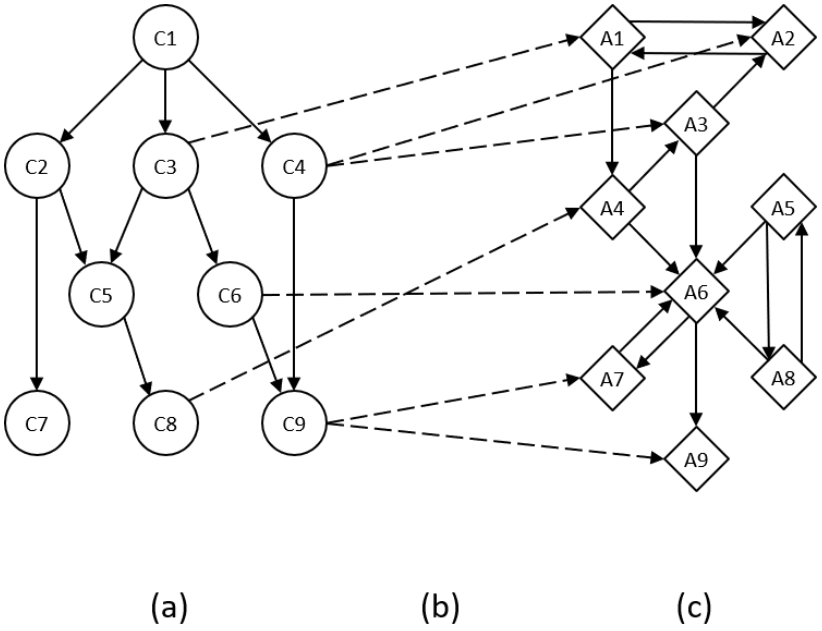


Fig. 3.3.: Schéma représentant la structure interne de Wikipedia, composée d'un graphe de catégories (a), et un graphe d'articles (c)

Pour simplifier la navigation, Wikipedia contient également un graphe de méta-données : **le graphe des catégories**. En effet, les catégories sont elles-mêmes organisées en un graphe hiérarchique, variant de domaines généraux (i.e. *Category*:*Geography*)

à des domaines extrêmement précis (i.e. *Category:Ports_and_harbours_of_Hauts-de-France*). L'aspect hiérarchique de ce graphe permet de considérer que si un article appartient à une catégorie donnée, il appartient également à toutes ses super-catégories. Par exemple, les *Ports_and_harbours_of_Hauts-de-France* sont liés au domaine de la géographie.

La figure 3.3 illustre la double structure en graphes de Wikipedia, et permet de visualiser l'interaction entre ces deux graphes. On y voit le graphe des catégories **(a)** avec sa structure hiérarchique, et le graphe des articles **(c)** avec sa structure sous forme de graphe orienté. Des liens bijectifs unissent les catégories et les articles **(b)**. Ils permettent aux utilisateurs de Wikipedia de naviguer jusqu'aux articles en se servant du système de catégories, et inversement de retrouver la catégorie d'un article.

Additionnellement à ce double graphe structurel des articles et des catégories de Wikipedia, nous avons exploité un certain nombre de méta-données supplémentaires présentes dans l'encyclopédie pour créer notre ensemble de thèmes. Ces méta-données sont énumérées ci-dessous :

Portails : Les Portails [6] (*Portals* en anglais), sont des méta-pages qui regroupent un ensemble d'articles et de catégories pertinentes vis-à-vis d'un sujet particulier. Ces pages sont destinées aux utilisateurs de Wikipedia qui souhaiteraient approfondir leurs connaissances dans un domaine particulier, sans réellement savoir par où commencer. Ainsi, les portails sont souvent réservés à des sujets très généraux, qui regroupent un grand nombre d'articles, tels que des domaines (ex : *Portal:Sciences*) ou des pays (ex : *Portal:China*). Au total, Wikipedia comprend plus de 1500 portails, parmis lesquels on dénombre 173 portails dit *présentable* par Wikipedia (*Featured Portals* en Anglais). Pour comprendre ces *featured portals* [7], il faut revenir sur l'aspect communautaire et collaboratif de Wikipedia, qui permet à tout un chacun d'en modifier le contenu et la structure, et par conséquent de créer et modifier des portails à volonté. Ainsi, les *featured portals* sont des portails particuliers sur lesquels Wikipedia, en tant qu'entité, garantit que le contenu qui s'y trouve est effectivement pertinent vis à vis du sujet traité.

Listes : Les listes [8] sont des méta-pages de Wikipedia qui regroupent de façon alphabétique un ensemble d'articles qui sont, d'une façon ou d'une autre, pertinents vis à vis d'un sujet commun. Ces listes sont souvent référencées à partir des Portails, et portent généralement sur une partie spécifique d'un domaine plus vaste (par exemple, le portail sur la guerre contient la liste : *List_of_states_with_nuclear_weapons*).

6. `https://en.wikipedia.org/wiki/Portal:Contents/Portals`
7. `https://en.wikipedia.org/wiki/Wikipedia:Featured_portals`
8. `https://en.wikipedia.org/wiki/Portal:Contents/Lists`

Outlines : Les Outlines [9] sont également des listes d'articles en rapport avec un sujet commun, mais se distinguent des listes ci-dessus par la façon dont sont présentés les articles sur la page. En effet, les listes classiques se contentent d'énumérer des articles, alors que les Outlines intègrent une notion d'importance et de hiérarchie. Ainsi, dans les Outlines, la position des articles présentés détermine leur importance. Cependant, au même titre que les *Portails*, les Outlines portent sur des sujets généraux (eg. *history, geography*). Ils permettent aux utilisateurs de cibler plus facilement un article précis mais de façon moins visuelle que les Portails.

9. `https://en.wikipedia.org/wiki/Portal:Contents/Outlines`

3.3 Identification des thèmes

Nous avons vu dans les sections précédentes que les bases de connaissances dérivées de Wikipedia utilisent les articles de l'encyclopédie pour créer des entités. Il existe donc un lien bijectif entre articles et entités, où chaque entité correspond à un article de Wikipedia. Par ailleurs, Wikipedia contient des méta-données de catégorisation de ces articles : les *catégories*. Ces dernières sont organisées hiérarchiquement en fonction de leur niveau de précision, et sont identifiées par un libellé.

En se basant sur la description des bases de connaissances dérivées de Wikipedia, et des éléments issus de l'encyclopédie elle-même, on restreindra dans la suite du document l'utilisation du terme *thème* pour désigner un ensemble d'articles/entités traitant d'une thématique commune. Les thèmes ont un identifiant, qui correspond au libellé d'une catégorie de Wikipedia. On définit ainsi la notion de thème de la façon suivante :

Définition 2: thème

Un *thème* est un ensemble d'entités qui ont en commun un sujet donné. Les thèmes sont désignés par un libellé. L'appartenance d'une entité à un thème est non exclusive, une entité peut donc appartenir à plusieurs thèmes simultanément.

Dans cette thèse, les thèmes jouent un rôle central dans le processus de génération de questions, car ils permettent de regrouper des articles liés à une thématique commune.

Dans le contexte de la génération automatique de questions relatives à un thème donné, la définition de ces thèmes soulève deux problématiques. La première problématique consiste à identifier les thèmes qui seront utilisés dans le cadre de la génération automatique de questions. Il s'agit de définir les critères qui sont propres à un thème, de façon à rendre son extraction et son exploitation aussi générique que possible. Il faut notamment que les thèmes utilisés soient suffisamment généraux et variés pour qu'ils soient en mesure de couvrir un maximum d'articles de l'encyclopédie. La seconde problématique est de réussir à identifier de façon automatique des ensembles d'articles liés à un thème commun et de les trier par ordre d'importance pour ce thème. Ainsi, en appliquant cette méthode sur les thèmes identifiés précédemment, on est effectivement en mesure d'obtenir un ensemble d'entités triées relatives à un thème donné.

Dans un premier temps, nous expliquons de quelle façon nous avons identifié automatiquement les thèmes qui seront utilisés par la suite. L'objectif étant *in fine* de mettre en place un référentiel suffisamment varié pour qu'il puisse être utilisable dans n'importe quel contexte. En effet, comme nous le détaillons ci-dessous, le processus d'approvisionnement d'un thème est une opération relativement coûteuse en termes de performance et de temps. La mise en place d'un référentiel général permet de s'adapter à une majorité de situations sans nécessiter de calculs systématiques.

Par ailleurs, l'importance d'une couverture maximale de l'ensemble des articles disponibles au travers des thèmes est elle aussi centrale dans le cadre d'un générateur de questions, dans la mesure où cela permet d'apporter une diversité, tant dans les thématiques des questions, que dans le contenu des énoncés ou les propositions de réponses.

Pour définir cet ensemble de thèmes généraux, notre approche consiste à exploiter la structure et les méta-données de Wikipedia présentées ci-dessus. En effet, avec son aspect collaboratif, Wikipedia offre une couverture maximale des événements, lieux, personnages, etc. tout en garantissant une précision factuelle extrêmement élevée en raison des révisions successives (Kräenbring et al., 2014).

Dans notre cas, ces méta-données structurelles se sont révélées particulièrement utiles pour définir nos thèmes, en nous permettant de nous baser sur des regroupements existants d'articles. En effet, à l'aide de son contenu et de ses méta-données, Wikipedia nous a permis de d'identifier des thèmes transversaux, et donc de définir automatiquement notre ensemble de thèmes. Nous avons vu en section 3.2 les méta-données de navigation de Wikipedia utilisées pour définir notre ensemble de thèmes. Du point de vue opérationnel, cela s'est traduit par un ensemble d'opérations successives listées ci dessous :

— Identifier et extraire automatiquement l'ensemble des *featured portals*

— Identifier et extraire automatiquement l'ensemble des *Outlines*

— Mettre en place une exploration récursive des catégories

— Extraire l'intersection des ensembles décrits ci-dessus, en limitant l'exploration récursive de l'arbre des catégories à une profondeur prédéterminée.

Dans la mesure où Wikipedia est une encyclopédie majoritairement éditée par ses utilisateurs, et non par un organisme central, un certain nombre de problèmes de cohérence se retrouvent disséminés au sein des méta-données. Ainsi, des tâches qui peuvent sembler triviales comme l'extraction récursive des catégories, nécessitent en réalité une attention particulière pour être résolues. En effet, cette structure qui devrait être simplement hiérarchique contient en réalité un grand nombre de boucles et de liens inappropriés.

Dans notre cas, nous souhaitons construire nos thèmes en se basant sur des thématiques générales. Nous avons émis l'hypothèse que plus une catégorie est haute dans la hiérarchie des catégories de Wikipedia, plus elle représente un thème général. Afin d'extraire ces catégories de l'arbre des catégories, il était ainsi nécessaire de connaître leur profondeur absolue dans l'arbre. On désigne ici sous l'appellation de *profondeur absolue* la distance la plus courte entre un noeud du graphe, et la racine, la racine ayant une profondeur absolue de 0.

Pour calculer cette profondeur absolue, en prenant en compte les incohérences et les erreurs présentes dans l'arbre des catégories, nous avons utilisé la stratégie suivante :

— Identification de la racine du graphe de catégories

— Exploration récursive de l'ensemble des sous-catégories depuis la racine :

— Pour chaque sous-catégorie non explorée, exploration récursive en augmentant la profondeur de 1

— Pour chaque sous-catégorie déjà explorée depuis une autre branche et ayant une profondeur strictement supérieur à la profondeur actuelle de la catégorie + 1, nouvelle exploration récursive en mettant à jour les profondeurs.

— Pour chaque sous-catégorie déjà explorée depuis une autre branche et ayant une profondeur inférieure ou égale à la profondeur du noeud + 1, aucune exploration supplémentaire n'est effectuée.

L'algorithme 1 représente l'application algorithmique de cette stratégie d'exploration récursive pour calculer la profondeur absolue des catégories de Wikipedia.

Algorithme 1 Calcul récursif de la profondeur des catégories

Require: *cat*, l'identifiant d'une catégorie quelconque de Wikipedia
Require: *p*, la profondeur actuelle, sois 0 pour la racine
Require: *mapCategories*, une Map contenant les informations des catégories

```
1:  function PROFONDEURCATEGORIES(cat : entier, p : entier)
2:      mapCategories.GET(cat) → profondeur ← p.
3:      entier[ ] subcats ← EXTRACTSUBCATSFROMFILE(cat, Subcategories.wiki)
4:      for each subcat ∈ subcats do
5:          if mapCategories.GET(subcat) → profondeur = −1 then
                                                  ▷ Nouvelle exploration
6:              PROFONDEURCATEGORIES(subcat, p + 1)
                                                  ▷ Appel récursif
7:          else if mapCategories.GET(subcat) → profondeur > p + 1 then
                                                  ▷ Mise à jour
8:              PROFONDEURCATEGORIES(subcat, p + 1)
9:          end if
                                  ▷ Les autres nœuds déjà explorés sont ignorés
10:     end for
11: end function
```

Le fichier $Subcategories.wiki$ contient une table de hashage correspondant aux sous-catégories de Wikipedia. Ce fichier est présenté plus précisément en section 3.5.

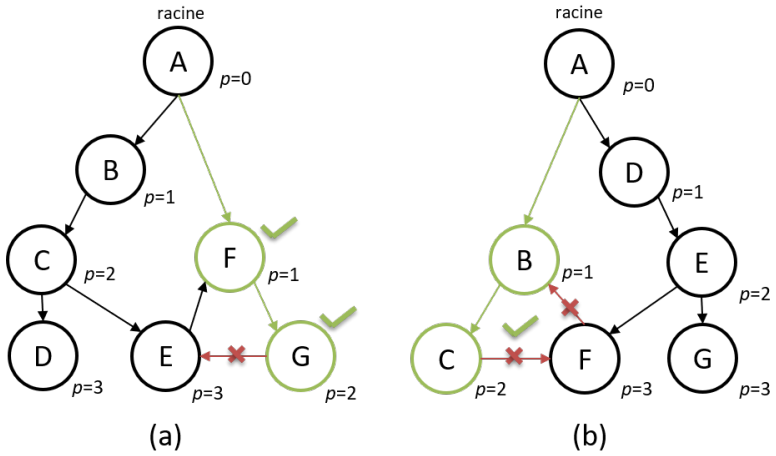


Fig. 3.4.: Exploration récursive du graphe de catégories de Wikipedia dans différents scénarios

Cette exploration est illustrée par la figure 3.4. Pour cet exemple, on part du principe que les branches sont récursivement explorées de gauche à droite. Notre solution, présentée avec le scénario *(b)* va ainsi explorer le sous-arbre de gauche en mettant à jour les profondeurs de façon classique. En revanche, lors de l'exploration de la seconde branche, le fils de droite ayant une profondeur supérieure à la profondeur du père + 1, cette branche sera elle aussi explorée et mise à jour. Dans le dernier scénario *(c)*, l'arbre est inversé. Dans ce cas-là, la branche de gauche est explorée normalement. Lors de l'exploration de la branche de droite, on ignorera l'exploration de la boucle dans la mesure où le noeud à déjà été exploré, et que sa profondeur est inférieure à la profondeur de son parent. La stratégie d'exploration que nous avons mise en place permet ainsi de valider l'objectif initial : calculer la profondeur absolue de chacun des nœuds du graphe (i.e. des catégories) indépendamment de l'ordre dans lequel ils sont explorés.

La valeur absolue de la profondeur des catégories calculée avec la méthode présentée ci-dessus nous permet ainsi de définir le niveau de précision de nos thèmes. Comme mentionné ci-dessus, plus la profondeur de la catégorie est importante, plus elle se référera à du contenu spécifique. Dans notre cas, nous avons opté expérimentalement pour une profondeur maximale de 7, afin de trouver un compromis entre la généralité des thèmes sélectionnés, leur nombre, voulu volontairement bas, et la couverture cumulée de ces thèmes en terme d'articles au sein de l'encyclopédie. Il est important de préciser ici que cette profondeur élevée se justifie par le fait que les premiers niveaux des catégories de Wikipedia regroupent principalement des méta-données transversales qui ne sont destinées qu'au fonctionnement interne de l'encyclopédie. On y trouve par exemple les catégories "*Category*:*Wikipedia_information_pages*", ou

"*Category*:*Wikipedia_administration*", qui ne sont pas pertinentes pour la définition de nos thèmes.

A l'aide des éléments présentés ci-dessus, nous avons pu définir automatiquement un ensemble de thèmes, tout en gardant la possibilité de l'affiner ou de l'étendre en faisant varier la profondeur de recherche de notre exploration récursive. Au final, nous avons identifié 52 thèmes transversaux à l'aide de cette méthode. La liste exhaustive de ces thèmes est présentée dans l'annexe A.

3.4 Approvisionnement du contenu des thèmes

Une fois notre ensemble de thèmes déterminé, l'objectif est d'associer à chacun d'entre eux un ensemble d'éléments qui lui correspondent. Nous utilisons pour cela les éléments de la structure de Wikipedia afin d'identifier et d'associer automatiquement un ensemble d'articles à chaque thème. L'objectif de cet approvisionnement est de mettre en place une méthode générique capable d'identifier automatiquement un ensemble d'éléments liés à un thème, et ce quel que soit le thème choisi.

3.4.1 Utilisation des méta-données de Wikipedia

En se basant uniquement sur la structure de Wikipedia, illustrée précédemment par la figure 3.3, on pourrait penser qu'une approche à la fois triviale et efficace consisterait à extraire récursivement les articles associés à toutes les sous-catégories liées à un thème donné. Cependant, en raison de son aspect collaboratif, de nombreux éléments de la structure de Wikipedia sont incohérents, et compliquent ce genre d'opération. Par exemple, de nombreuses boucles de l'arbre des catégories ne devraient pas s'y trouver. Pire encore, en raison de certaines méta-données internes au fonctionnement de Wikipedia, une partie de ces boucles fait référence à des catégories se trouvant en amont de la catégorie correspondant au thème, ce qui se traduit presque systématiquement par l'ajout de la quasi totalité des articles de Wikipedia dans chaque thème.

Ainsi, pour accomplir notre objectif consistant à n'associer aux thèmes que des articles qui sont réellement pertinents pour ces derniers, nous avons combiné plusieurs approches :

Premièrement, il a fallu **nettoyer l'arbre des catégories** de ses incohérences. Pour cela nous avons réutilisé l'approche présentée en section 3.3, dans laquelle nous avions calculé la profondeur absolue de chaque catégorie, en commençant l'exploration à la racine (voir figure 3.4). En se basant sur cette profondeur, nous avons pu comparer les profondeurs des sous-catégories liées au thème. Cette opération a permis à la fois d'éviter les boucles référençant des articles externes au thème, mais également d'éviter l'exploration de sous-branches dont les profondeurs absolues auraient été supérieures à celle de la branche du thème. Cependant, malgré ces mécanismes, les catégories internes au fonctionnement de Wikipedia, ainsi que les erreurs humaines de référencement continuaient systématiquement à faire diverger une exploration récursive complète. Pour contrer ce problème, nous avons limité l'exploration récursive aux n niveaux suivant la racine du thème. Nous avons mesuré expérimentalement que les meilleurs résultats sont obtenus quand $3 \leq n \leq 5$.

Algorithme 2 Extraction récursive des articles des catégories

Require: *cat*, l'identifiant d'une catégorie quelconque de Wikipedia
Require: *p*, la profondeur actuelle, sois 0 à la première itération
Require: *pmax*, la profondeur d'exploration recherchée
Require: *mapCategories*, une Map contenant les informations des catégories

```
1:  function EXPLORECATEGORIE(cat : entier, p : entier, pmax : entier) : Article[ ]
2:      Article[ ] articles ← ∅
3:      if p > pmax then
                                                          ▷ condition d'arrêt
4:          return articles
5:      end if
6:      entier p_cat ← mapCategories.GET(cat) → profondeur
                                  ▷ on récupère la profondeur de la catégorie
7:      entier[ ] subcats ← EXTRACTSUBCATSFROMFILE(cat, Subcategories.wiki)
8:      for each subcat ∈ subcats do
9:          entier p_subcat ← mapCategories.GET(subcat) → profondeur
                                  ▷ on récupère la profondeur de la sous-catégorie
10:         if p_subcat < p_cat then
                                  ▷ on ignore les branches qui remontent dans l'arbre
11:             articles+ = EXPLORECATEGORIE(subcat, p + 1, pmax)
                                                          ▷ Appel récursif
12:         end if
13:     end for
14:     Article[ ] cat_arts ← EXTRACTCATARTICLESFROMFILE(cat, Page_category.wiki)
15:     for each art ∈ cat_arts do
16:         articles+ = art
17:     end for
18:     return articles
19: end function
```

Cet algorithme se base sur une fonction récursive, présentée dans l'algorithme 2, qui extrait récursivement les articles liés aux sous-catégories d'une catégorie passée en paramètre. Ainsi, pour la profondeur de l'arbre choisie, l'ensemble des articles de la catégorie passée en paramètre et de ses sous-catégories sont récursivement extraits. On obtient ainsi un ensemble composé de tous les articles de la catégorie et de ses sous-catégories, pour une profondeur d'exploration donnée. On peut noter à la ligne 10 de l'algorithme 2, que la valeur de profondeur absolue des catégories, calculée par l'algorithme 1 nous permet d'éviter d'explorer les branches qui remontent dans l'arbre. Les fichiers $Subcategories.wiki$ et $Page\_category.wiki$ contiennent des tables de hashage correspondant respectivement aux sous-catégories et aux liens pages/catégories de Wikipedia. Ces fichiers sont présentés plus précisément en section 3.5.

Deuxièmement, nous avons complété cette approche en intégrant aux ensembles d'éléments qui constituent nos thèmes des articles issus directement de sources considérées comme pertinentes. Les pages d'Outlines, Listes et Portails contiennent en effet un certain nombre d'articles directement liés à un thème donné. Ces pages constituent une synthèse du thème présenté, en rassemblant des liens vers tous ses articles importants. On considère donc comme pertinents tous les éléments qui y sont présentés. Ces éléments peuvent être extraits à partir de deux types de pages : les *Outlines*, et les *Portails*. En plus des articles, ces pages peuvent contenir des listes et des catégories, dont les articles sont également pertinents. Pour extraire et ajouter les éléments contenu dans ces pages nous avons procédé comme suit :

1. Vérifier s'il existe une page Outline et/ou une page Portail correspondant au thème,
2. Extraire la totalité des articles de ces pages, et les ajouter à la liste des articles du thème,
3. Extraire les listes présentes dans les pages, et ajouter tous les articles de chaque liste aux articles du thème,
4. Extraire les catégories référencées dans la page, et pour chacune d'entre elles, ajouter les articles directement référencés aux articles du thème, sans exploration récursive.

Finalement, en combinant les deux approches présentées ci-dessus (extraction supervisée des catégories de Wikipedia, et extraction thématique des métas-pages de navigation), nous avons été en mesure d'associer un ensemble d'articles à chacun de nos thèmes. De façon pragmatique, ces articles marquent le périmètre de chaque thème, distinguant ainsi les éléments qui lui sont rattachés de ceux qui ne le sont pas. A ce stade, ces éléments ne constituent qu'un ensemble non ordonné d'éléments. Nous expliquons dans la section suivante la méthode utilisée pour trier cet ensemble, afin de déterminer les éléments les plus importants d'un thème donné.

3.4.2 Classement des articles

En se basant sur la structure de Wikipedia, nous avons déterminé à l'étape précédente l'ensemble des articles qui délimitent nos thèmes. Ce périmètre est construit à partir du lien bijectif qui unit les catégories et les articles de Wikipedia, ainsi que sur les éléments directement identifiés au sein des *Portails* et des *Outlines*. Pour utiliser au mieux cet ensemble d'articles qui compose chaque thème, il est nécessaire de déterminer l'importance relative de chacun de ses éléments. Nous avons mis en place une méthode permettant de trier les éléments *par ordre d'importance* au sein de chaque thème. En se basant sur la structure de Wikipedia, et plus précisément sur le graphe des articles présenté par la figure 3.3, nous avons utilisé les liens entre les articles pour établir ces classements.

L'algorithme du *Pagerank* (BRIN et PAGE, 1998) considère au sein d'un graphe orienté que plus un noeud est référencé par d'autres nœuds, plus il est important. Pour obtenir ce résultat, l'algorithme fonctionne sur le principe d'un déplacement aléatoire d'un noeud à l'autre. Chaque fois qu'un noeud est visité, son importance augmente. Un autre noeud est ensuite choisi aléatoirement parmi les noeuds accessibles. L'opération se répète jusqu'à ce qu'il n'y ai plus de noeud accessible (impasse), ou aléatoirement, en se basant sur une probabilité d'arrêt à chaque noeud. Cette probabilité est généralement fixée à 0.15, et c'est également la valeur que nous avons utilisé lors de l'application de l'algorithme du *Pagerank* à notre problématique. Cette opération de parcours est ainsi répétée un grand nombre de fois, jusqu'à obtenir une convergence des valeurs d'importance des noeuds du graphe. Il s'agit de la phase où ces probabilités ne seront plus amenées à évoluer, ou alors très faiblement.

Dans notre cas, nous n'appliquons pas cet algorithme à la totalité de Wikipedia, mais à chaque sous-graphe constitué par les articles de chaque thème. Cela nous permet de calculer un score de pertinence individuel pour chaque article lié à un thème. De cette façon, chacun des articles peut avoir un score de *Pagerank* différent pour chacun des thèmes, voire un score nul dans certains d'entre eux si cet article n'a aucun rapport avec ces thèmes.

La figure 3.5 illustre l'application de l'algorithme de *Pagerank* au thème 'Art'. Dans un premier temps, on peut y voir le graphe orienté construit à partir des articles de Wikipedia. Ce graphe ne contient que les articles reliés à l'art de près ou de loin. Dans un second temps, en utilisant l'algorithme du *Pagerank*, l'importance de chacun de ces articles est pondérée. Cela permet par la suite d'obtenir un score pour chacun d'entre eux, et de les classer au sein du thème.

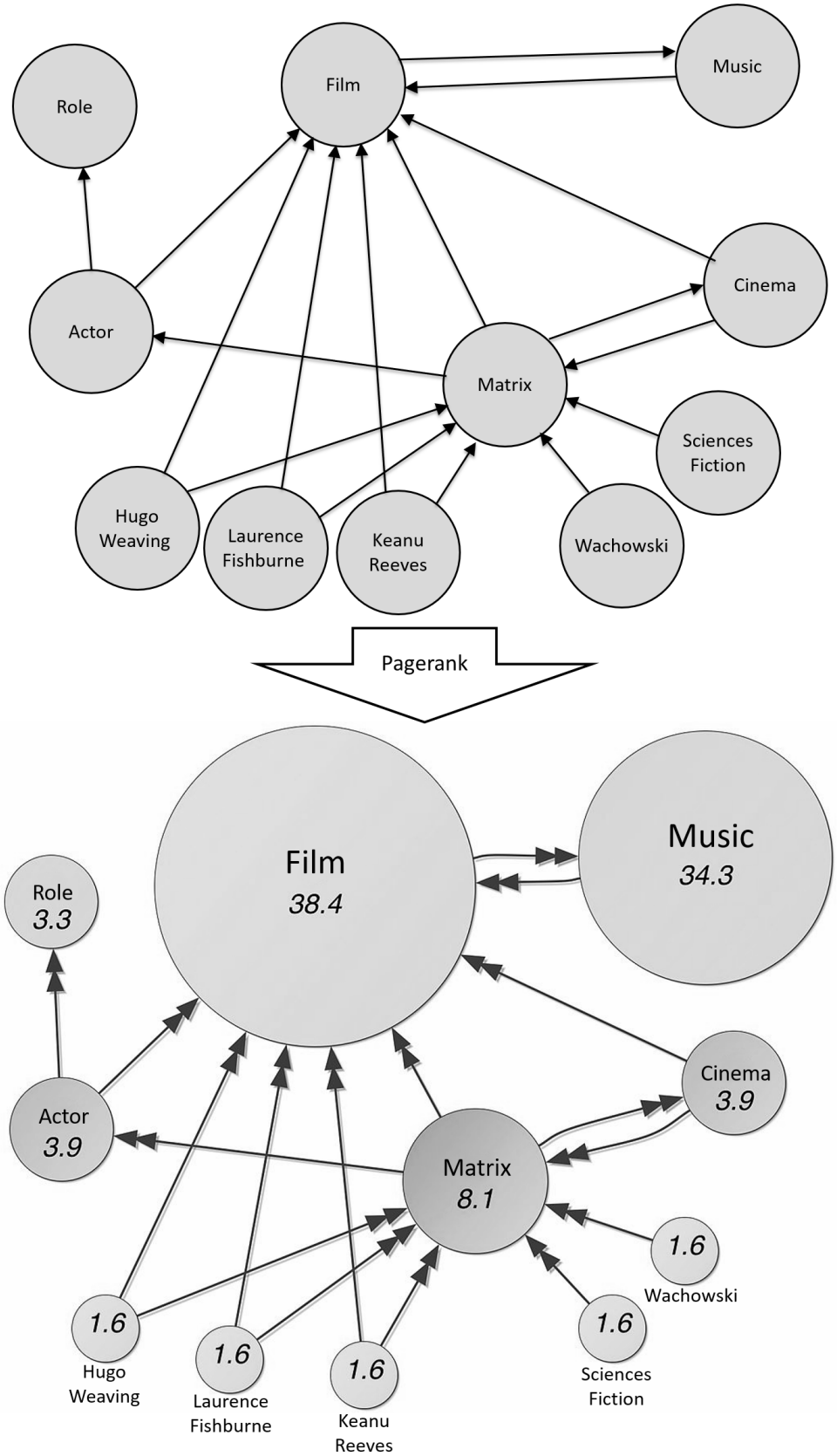


Fig. 3.5.: Illustration de l'application de l'algorithme du *Pagerank* sur le thème 'Art'

Dans la suite du document, nous utiliseront le symbole $sa(i,t)$, pour décrire le score d'un article i au sein d'un thème t donné. Ce score est la résultante du score obtenu après l'application de l'algorithme du *Pagerank* sur le graphe orienté issu du thème t.

3.4.3 Extension du contenu des thèmes avec LSA

Wikipedia est une encyclopédie dont le contenu et la structure sont majoritairement réalisés par des humains, et ce, sans concertation préalable entre les différents contributeurs. Cela explique que certains défauts de catégorisation sont présents dans la structure. On trouve par exemple des articles référencés par des catégories inappropriés, ou inversement une absence de référencement pertinente. Pour minimiser l'impact de ces incohérences, nous avons rajouté une étape destinée à améliorer le contour des thèmes, en y insérant des articles pertinents qui n'auraient pas été identifiés lors de l'extraction présentée précédemment. Cette extension se base sur la similarité sémantique avec le *contenu* des articles déjà présents dans le thème, et non sur la structure de Wikipedia déjà utilisée par les opérations précédentes.

Pour identifier la totalité des candidats à inclure dans un thème donné, il aurait été nécessaire de mesurer la similarité sémantique de chaque article du thème avec tout article de Wikipedia n'en faisant pas partie. Compte tenu du temps de traitement démesuré requis pour cette tâche, nous avons restreint le problème en ne considérant que les articles les plus importants du thème, en se référant aux résultats du *Pagerank* obtenus à l'étape précédente. En effet, la répartition des scores du *Pagerank*, visible sur la figure 3.6 indique que la pertinence d'un article donné au sein du thème décroit rapidement quand son rang augmente. Il n'était par conséquent pas nécessaire d'effectuer la comparaison avec la totalité des articles du thème, mais seulement avec les mieux classés.

Ainsi, pour chaque thème, nous avons extrait le contenu textuel des pages correspondant au top-k des articles les mieux classés au sein du thème, où k correspond au rang à partir duquel la courbe du *Pagerank* se stabilise. k est ainsi variable d'un thème à l'autre. Nous avons ensuite mesuré la similarité sémantique à l'aide de LSA (WIEMER-HASTINGS et al., 2004) entre chacun des articles du top-k, et la totalité des autres articles de Wikipedia. Les nouveaux articles obtenus de cette façon ont ensuite été ajoutés à l'ensemble des articles qui constituent le thème.

Une fois les nouveaux articles ajoutés, il est nécessaire de re-calculer le *Pagerank* du thème en utilisant le même procédé que celui expliqué en section 3.4.2. Afin d'améliorer encore l'ensemble des articles du thème, il est possible de recommencer ces étapes d'extension LSA et de *Pagerank* en ajoutant au thème les articles sémantiquement proches des nouveaux articles du top-k, et ce jusqu'à ce qu'il n'y ait plus de nouvel article dans le top-k.

3.4.4 Filtrage des articles faiblement classés

L'exploration récursive des catégories de Wikipedia, et les extensions LSA successives permettent de créer des thèmes qui contiennent pour la plupart des dizaines de milliers d'articles. Ces étapes sont nécessaires pour identifier la totalité des articles pertinents pour chaque thème, mais cela ne signifie pas que la totalité des articles identifiés soient pertinents. Pour illustrer cette affirmation, on peut citer l'exemple du *Onrust* [10], un bateau marchand allemand du XVII^{ème} siècle qui n'a rien d'exceptionnel. Cet article est pourtant rattaché au thème *geography*, car il figure dans l'une des sous-catégories de ce dernier. Ces articles, qui sont en réalité extrêmement nombreux, se révèlent coûteux en termes de temps de traitement alors qu'ils n'apportent pas de plus-value concrète aux thèmes. Ainsi, nous détaillons dans cette section la méthode que nous utilisons pour filtrer ce genre d'articles au sein de nos thèmes, tout en conservant les éléments pertinents.

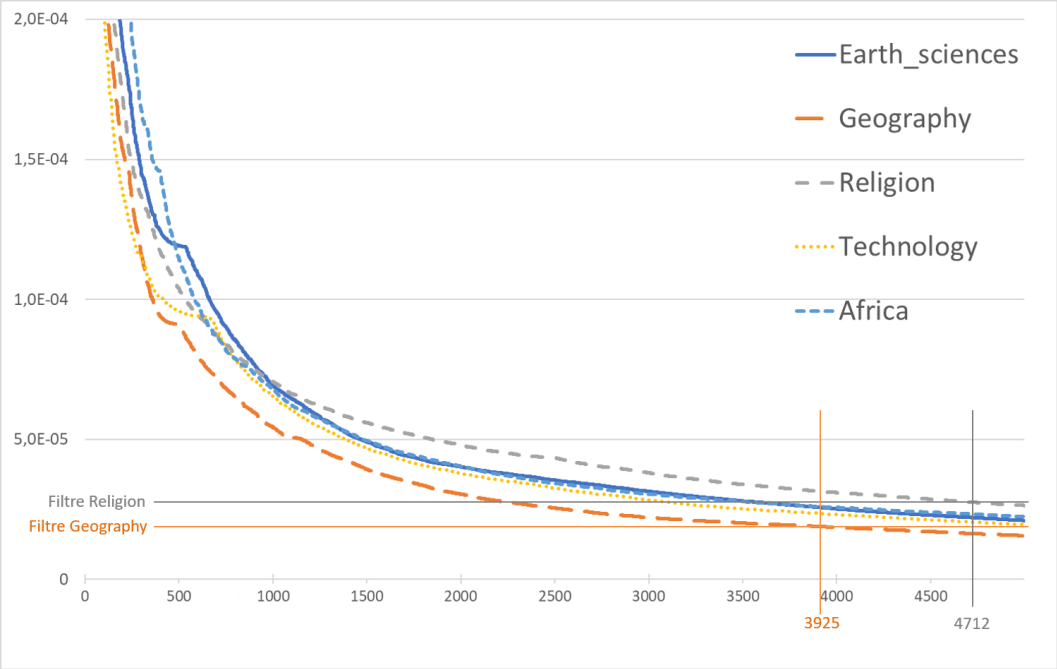


Fig. 3.6.: Exemples de distribution des scores de *Pagerank* sur plusieurs thèmes, et application du flitrage

Le score de *Pagerank* des articles au sein des thèmes, calculé lors de la section 3.4.2, affiche une distribution proche d'une fonction de variation inverse. Le score des articles décroit rapidement quand leur rang augmente, ce qui indique que seule la première partie des articles de chaque thème est réellement pertinente. Le graphique de la figure 3.6 montre le tracé des scores de *Pagerank* de plusieurs thèmes. Précisons que même s'il ne sont pas tous représentés, les thèmes suivent tous une distribution similaire de *Pagerank*.

10. `https://en.wikipedia.org/wiki/Onrust`

En se basant sur ce constat, nous avons émis l'hypothèse qu'il n'était pas nécessaire de conserver la totalité des articles de chaque thème, mais seulement les mieux classés. Dans cette optique, nous avons mis en place un filtre, destiné à limiter le nombre d'articles de chaque thème, tout en conservant les éléments pertinents. Pour vérifier cette hypothèse, nous avons opté pour la mise en place d'un filtre à seuil variable, qui ignore tous les articles dont le score est inférieur à un seuil défini automatiquement pour chaque thème. Ce seuil est déterminé automatiquement en se basant sur la distribution du *Pagerank*. Lorsque la courbe se rapproche suffisamment de zéro, c'est-à-dire que le score des articles commence à stagner, nous filtrons les articles suivants.

Nous avons ensuite comparé la pertinence des résultats obtenus en faisant varier la précision du seuil. Cette mesure de pertinence a été réalisée à travers le NDCG (Normalized Discounted Cumulative Gain) (JÄRVELIN et KEKÄLÄINEN, 2002), permettant d'établir la pertinence du classement obtenu après le filtre, en le comparant aux résultats obtenus avant.

Filtre	$PR < 5.10^{-6}$ NDCG	$PR < 10^{-5}$ NDCG	$PR < 5.10^{-5}$ NDCG
Arts (1.6)	.99	.96	.58
Politics (1.2)	1	1	.76
⋮	⋮	⋮	⋮
WW1 (0.1)	1	1	.99
Average	.99	.96	.68

Tab. 3.1.: Résultats expérimentaux pour divers filtres de *Pagerank*. Les résultats sont présentés avec des exemples de thèmes, et pour la moyenne de ces derniers.

Les résultats, visibles dans la table 3.1, indiquent que le meilleur compromis est atteint quand le filtre a une précision de 10^{-6}. En effet, pour cette valeur on maximise le NCDG obtenu (99%), i.e. la liste des articles les mieux classés n'est presque pas modifiée, ce qui n'affecte pas le classement défini dans ce chapitre.

3.5 Implémentation

Wikipedia est avant tout une encyclopédie en ligne destinée à des utilisateurs humains. Cela implique que l'encyclopédie se présente sous la forme de pages Web, hébergées sur les serveurs de Wikipedia, ce qui complique et ralentit fortement les traitements automatiques. Sous ce format, il est difficile, voire impossible de l'exploiter efficacement. Il existe cependant des fichiers dump, qui permettent aux administrateurs de Wikipedia de sauvegarder l'ensemble des données du site, pour permettre une restauration en cas de besoin. Ces fichiers sont à la base de nos traitements.

Les fichiers dump de Wikipedia se présentent sous différents formats, la plupart étant stockés en SQL et en XML. Bien que ces fichiers soient plus propices à un traitement automatique que les pages HTML du site, ils n'en contiennent pas moins de nombreuses informations se révélant superflues pour notre générateur. Nous avons ainsi mis en place un format de stockage optimisé, ne contenant que les éléments dont nous avons besoin. Voici quelques détails sur les fichiers dump que nous avons utilisés :

— **La liste des pages**, donnée par le fichier *pages.sql*. Nous avons extrait de ce fichier l'identifiant et l'intitulé de chaque page.
— **La liste des catégories**, donnée par le fichier *category.sql*.
— **La liste des liens entre les pages**, donnée par le fichier *pagelinks.sql*. Une occurrence est présente dans ce fichier chaque fois qu'une page est référencée dans le corps d'une autre.
— **La liste des liens entre pages et catégories**, donnée par le fichier *categorylinks.sql*. Les catégories étant elles-mêmes stockées sous forme de pages, ce fichier a permis d'extraire à la fois les liens entre les catégories et les pages, et les liens hiérarchiques entre les catégories.

Ces fichiers dump sont stockés sous forme SQL pour les besoins de Wikipedia. L'excédant d'éléments que contiennent ces fichiers, ainsi que leur syntaxe lourde due à SQL, et donc coûteuse en mémoire, nous ont amenés à ne conserver que les éléments dont nous avions besoin. Nous utilisons pour cela des fichiers de stockage optimisés, que nous avons appelés *Wikis*, et qui ne contiennent qu'une version strictement épurée des fichiers présentés ci-dessus. Nous avons défini 5 fichiers :

— **Pages.wiki** contient la liste des pages de Wikipedia. Ce fichier contient uniquement l'identifiant et l'intitulé de l'article sous forme de Map, où l'identifiant de la page représente la clé, et l'intitulé représente la valeur.

— **Categories.wiki** contient la liste des catégories de Wikipedia. Wikipedia distingue les identifiants de pages des identifiants de catégories dans ses métadonnées : pour Wikipedia, les catégories sont à la fois l'un et l'autre. Nous stockons donc dans ce fichier ces deux éléments, ainsi que l'intitulé de la catégorie. Le stockage est réalisé sous forme d'une double Map, où les clés sont à la fois l'identifiant de page et l'identifiant de catégorie.

— **Pagelinks.wiki** contient les liens entre les pages de Wikipedia. Pour plus de simplicité, le fichier est stocké sous la forme d'une *Multimap*. Pour chaque entrée, le premier élément contient l'identifiant de l'article source du lien, tous les suivants sont les destinations correspondantes.

— **Page_category.wiki** relie une page avec ses catégories directes. Ce fichier est stocké sous forme de Multimap, la clé étant l'identifiant de l'article, et l'ensemble de valeurs représentant toutes ses catégories directes. On utilise ici le terme de *catégorie directe* pour désigner une catégorie explicitement référencée dans l'article. Les catégories *indirectes* identifiées automatiquement à partir d'un lien hiérarchique ne sont pas stockées dans ce fichier.

— **Sub_categories.wiki** contient les liens hiérarchiques entre les catégories. Pour augmenter la rapidité des opérations, nous avons séparé la gestion des sous-catégories, de celle de l'appartenance des pages aux catégories. Ce fichier est stocké sous forme de Multimap, la clé étant l'identifiant d'une catégorie, et les valeurs correspondant à l'ensemble de ses super-catégories directes au sein de la structure de Wikipedia.

Cette extraction de la structure de Wikipedia offre de nombreux avantages pour la manipulation de son contenu. Il devient par exemple possible de charger tout ou partie de la structure en mémoire vive, évitant ainsi des accès systématiques aux disques durs, ce qui augmente considérablement la vitesse de traitement. Par ailleurs, la répartition des fichiers permet de n'utiliser que les parties de la structure nécessaires à une tâche donnée. On évite ainsi de charger des parties inutiles, ce qui accélère le traitement.

Cette structure ayant été ramenée à la forme de tables de hachage, l'espace mémoire requis pour sa manipulation est fortement réduit. La figure 3.7 permet de visualiser les différentes tables de hachage utilisées pour stocker la structure de Wikipedia, et leurs interactions. On peut y voir que l'ensemble des chaînes de caractères, qui constituent les éléments qui consomment le plus de mémoire, ne sont stockées qu'une seule fois. Toutes les autres tables utilisent des identifiants uniques permettant de faire le lien avec ces chaînes. Ce format sous forme d'identifiants permet d'accélérer considérablement la vitesse d'accès aux différentes ressources, et réduit donc drastiquement le temps d'exécution nécessaire à la construction et a la manipulation des thèmes.

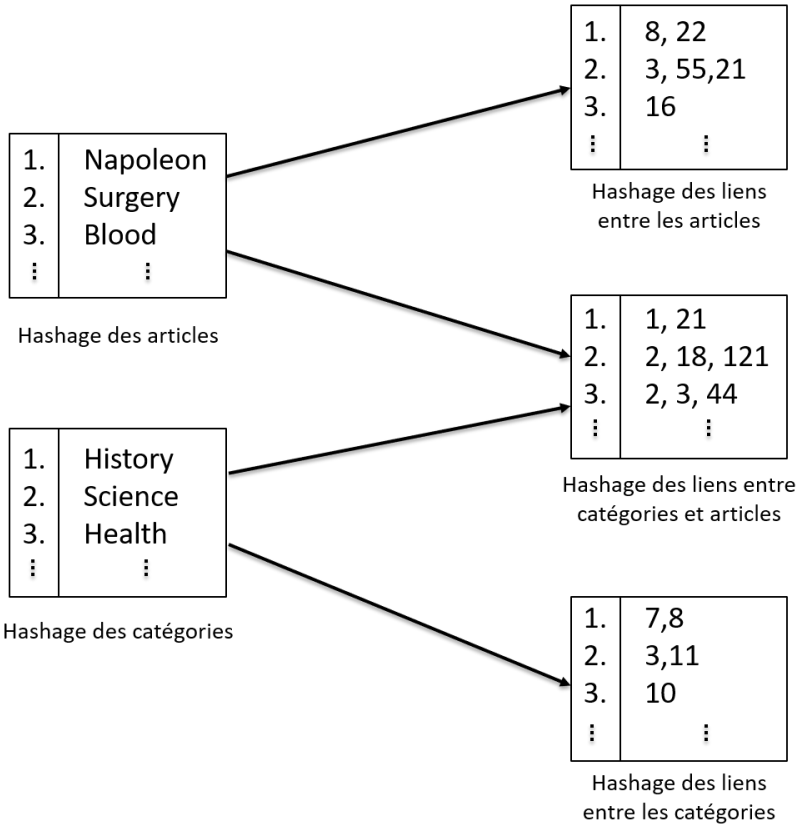


Fig. 3.7.: Illustration du système de hashage utilisé pour stocker la structure de Wikipedia

Dans la mesure où les étapes de création et d'extraction des thèmes sont des opérations coûteuses en termes de ressources et de temps de traitement, il n'est pas envisageable de refaire l'ensemble de ces opérations à chaque fois que l'on souhaite utiliser les thèmes. Ainsi, leur extraction et leur approvisionnement intervient comme une étape de pré-traitement. Nous avons ainsi mis en place une base de données pour stocker ces résultats.

Compte tenu de la nature des données à stocker, le langage SQL offrait un bon compromis en terme de stockage. En effet, les données des thèmes peuvent se représenter facilement sous forme de tables SQL, et la vitesse d'exécution des requêtes est sans équivoque pour identifier les données recherchées au sein des tables. Par ailleurs, SQL peut s'interconnecter avec la quasi totalité des langages de programmation, ce qui est idéal pour du traitement automatique.

Le schéma relationnel de la figure 3.8 décrit les tables et les attributs utilisés pour représenter notre base de données de thèmes. Ce schéma permet de stocker le score de classement correspondant à l'association d'un thème et d'un article. Les thèmes et les articles sont tous les deux composés d'un identifiant et d'un label, des tables spécifiques leur sont attribuées, au sein desquelles l'identifiant est utilisé en tant que clé primaire. La table *Ranking* utilise pour sa part l'association d'un identifiant

de thème et d'un identifiant d'article pour constituer sa clé primaire, afin de leur associer le score de l'article au sein du thème. Une table additionnelle de ressources externes est constituée à partir de l'identifiant de l'article. Cette table permet de stocker des identifiants désignant des entités spécifiques correspondant à l'article au sein des bases de connaissances dérivées de Wikipedia. Cette table opère comme un dictionnaire permettant de faire le lien de l'un à l'autre.

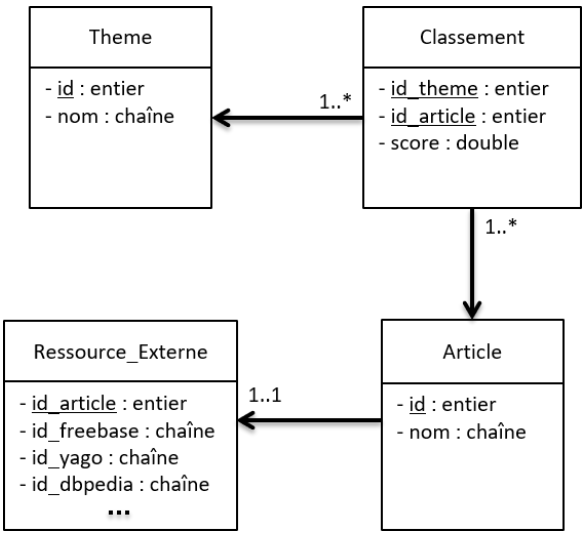


Fig. 3.8.: Schéma relationnel de la base de données de thèmes

3.6 Application : Navigation par thèmes

Afin de valoriser l'utilité des thèmes, et de démontrer leur efficacité, nous avons mis en place une utilisation dérivée, avec Fouilla (RAYNAUD et al., 2018a). Nous avons ainsi introduit le concept de *Navigation par thèmes* (*topical browsing* en anglais). La navigation par thèmes consiste à explorer des bases de connaissances, en filtrant et triant les entités qui la composent en fonction de thèmes choisis par l'utilisateur. Cette méthode, qui s'applique aux bases de connaissances dérivées de Wikipedia, permet d'identifier et trier des relations entre plusieurs entités là où il est normalement impossible de le faire. En effet, les bases de connaissances doivent théoriquement contenir des méta-données explicites, qui sont ensuite utilisées comme filtre par le langage SPARQL lors de la recherche de données. Dans notre cas, ces données ne sont pas présentes dans la base de connaissances, mais se basent sur des méta-données externes.

L'approche de navigation par thèmes que nous introduisons ici se distingue des approches existantes de *navigation par facettes* permettant de filtrer le contenu des bases de connaissances à partir de méta-données internes à la base. Pour mieux expliquer l'intérêt de Fouilla et de la navigation par thèmes, nous détaillons dans un premier temps les travaux existants sur la navigation par facettes.

3.6.1 Approches existantes de la navigation par facettes

Les bases de connaissances sont stockées sous forme de triplets. Ces triplets sont composés d'un sujet, d'un prédicat, et d'un objet. Le sujet d'un triple est toujours une entité de la base de connaissances, mais ce n'est pas toujours le cas des objets. En effet, ces derniers peuvent être des objets ou des littéraux, auxquels sont notamment associées des données numériques (dates, superficies, distances, etc.), ou des données textuelles (labels, noms, descriptions, etc.).

On appelle *navigation par facettes* (ou *faceted browsing* en anglais), le fait de filtrer les triples d'une base de connaissances en utilisant les attributs des entités. Il s'agit donc d'un filtre qui utilise les méta-données de la base de connaissances pour obtenir un sous-ensemble de résultats spécifiques. De façon pratique, on pourra ainsi obtenir l'ensemble des chefs d'état des pays ayant une superficie supérieure à $500\,000\ km^2$, ou encore la liste des acteurs de 25 à 30 ans ayant tourné dans des films entre 2000 et 2015, car ces méta-données sont présentes dans la base de connaissances, et ainsi exploitables directement a l'aide de requêtes SPARQL.

Plusieurs approches utilisent ainsi les méta-données et la sémantique liée aux relations des bases de connaissances pour appliquer un filtre partiel sur les données des bases de connaissances en fonction de modèles prédéfinis. (Ross et al., 2005) ont mis en place un système qui utilise les méta-données de bases de connaissances afin d'identifier automatiquement des facettes pertinentes dans cette dernière. (Ding et al., 2005) ont introduit *Swoogle*, un moteur d'indexation se servant des facettes de bases de connaissances pour indexer le contenu de documents disponibles sur le Web. (Oren et al., 2006) ont mis en place une interface permettant de naviguer dans les bases de connaissances en sélectionnant des critères, issus des relations des bases explorées. Cette navigation consiste à filtrer les résultats qui ne correspondent pas aux critères définis par l'utilisateur. (Tvarozek et Bieliková, 2007) ont mis en place un navigateur par facette qui s'adapte au profil de l'utilisateur, et filtre ainsi automatiquement le contenu de la base de connaissances sur un ensemble de relations liées au profil.

Les approches de navigation par facettes, ou de filtrage par facettes sont donc particulièrement utiles pour faciliter l'exploration d'une base de connaissances, pouvant contenir plusieurs millions d'entités, aux quelques-unes qui correspondent à des critères de recherche définis par l'utilisateur. Cependant, les filtres proposés par la navigation par facettes sont limités par les méta-données directement présentes dans la base de connaissances. Nous avons donc introduit la notion de navigation par thèmes, détaillée ci-dessous, afin d'introduire une dimension de filtre supplémentaire aux bases de connaissances. En effet, la navigation par thèmes permet de filtrer le

contenu de ces bases en utilisant des notions thématiques qui ne sont initialement pas présentes au sein de ces dernières.

Il est important de noter que la navigation par facettes et la navigation par thèmes ne constituent pas deux approches incompatibles l'une avec l'autre, et pourraient être combinées pour obtenir des résultats précis à la fois vis-à-vis d'une thématique externe à la base de connaissances, et d'un ensemble de relations internes à cette dernière.

3.6.2 Fouilla : démonstration de navigation par thèmes

Nous présentons notre interface de navigation par thèmes sous la forme d'une application web. Sur cette dernière, nous proposons à l'utilisateur d'explorer le contenu de DBPedia en rajoutant le filtre des thèmes, ce qui permet d'afficher et de trier uniquement les éléments en rapport avec le sujet qu'il a choisi. Nous avons envisagé trois scénarios qui permettent d'utiliser au mieux ce filtre par thèmes au sein de DBpedia :

1. **Trier les entités d'un thème :** Dans ce scénario, l'utilisateur choisit un thème, et obtient la liste triée des entités considérées comme les plus pertinentes pour ce thème. Il s'agit dans les faits du classement obtenu directement à l'aide de l'algorithme du *Pagerank*, présenté dans la section 3.4.2. Un lien vers DBpedia permet ensuite à l'utilisateur d'obtenir plus de méta-données, tandis qu'un lien vers Wikipedia permet de visualiser la page de l'entité.
2. **Trier les triples d'un thème :** Dans ce scénario, on ne présente plus seulement des entités à l'utilisateur, mais des triples composés de deux entités et d'un prédicat qui les relie. Comme dans le cas précédent, l'utilisateur choisit un thème, et nous lui présentons les triples qui sont les plus pertinents pour ce thème. En plus des liens vers DBpedia et Wikipedia pour les entités, un lien vers DBpedia permet d'obtenir plus de détails sur la nature du prédicat.
3. **Filtrer les triples d'une entité au sein d'un thème :** Ce scénario présente le cas d'utilisation le plus intéressant de la navigation par thème. Dans ce scénario, l'utilisateur choisit une entité et Fouilla filtre dans la page de cette entité les triples liés à un thème prédéfini. Cela permet de n'afficher qu'un aspect de cette entité, permettant à l'utilisateur de trouver des faits parfois occultés par un aspect plus important de cette entité. Par ailleurs, certaines entités pluridisciplinaires présentent des éléments complètement différents selon le thème par lequel elles sont analysées. C'est le cas de la France par exemple, où les triples présentés sont très différents selon si on cherche plutôt un aspect géographique, historique, ou artistique.

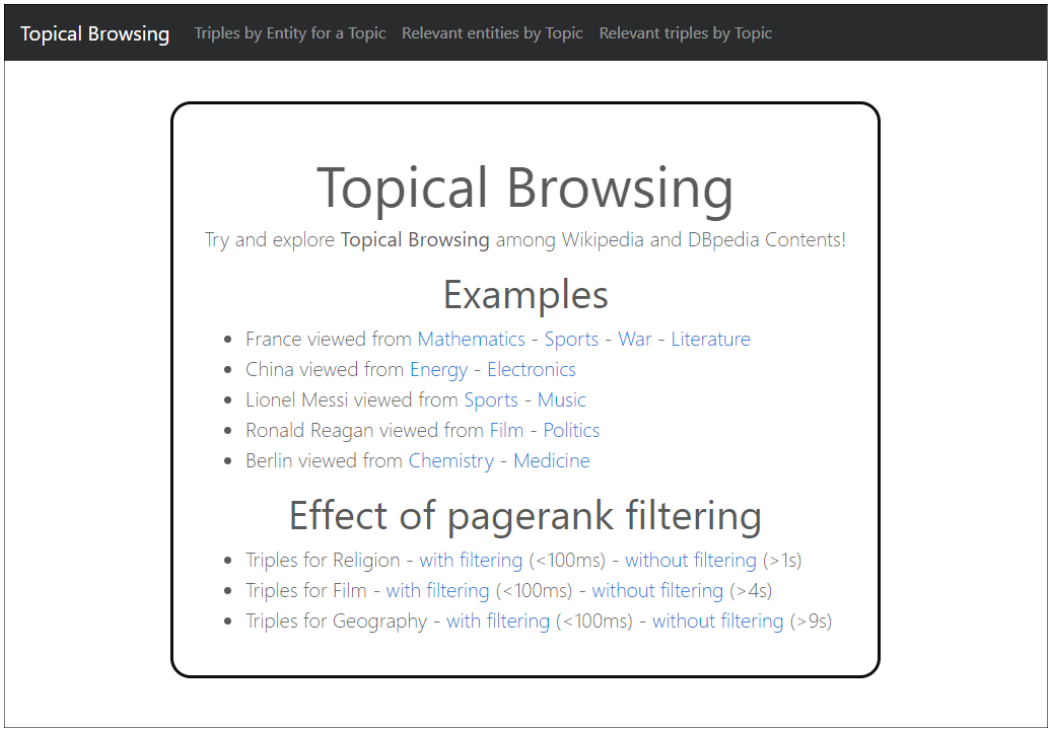


Fig. 3.9.: Page d'accueil de Fouilla, le site web de navigation par thème

Ces différents scénarios sont proposés sur le site web de l'application, à l'adresse http://demo-satin.telecom-st-etienne.fr/fouilla/. Un certain nombre de cas d'utilisation sont proposés à l'utilisateur pour lui permettre de juger par lui-même de l'efficacité de la méthode. Il peut par exemple choisir de filtrer les triples de la France avec le thème des mathématiques, ou celui de la littérature. L'application propose également par exemple de séparer la carrière politique de la carrière cinématographique de Ronald Reagan. Ces différents scénarios sont proposés dans la page d'accueil du site Web (voir figure 3.9). Il reste bien sûr possible à l'utilisateur d'effectuer une recherche libre, lui permettant de filtrer toute entité avec un thème de son choix.

	$PR < 5.10^{-6}$		$PR < 10^{-5}$		$PR < 5.10^{-5}$	
	NDCG	Speedup	NDCG	Speedup	NDCG	Speedup
Arts (1.6)	.99	26.89	.96	52.8	.58	122
Politics (1.2)	1	19.87	1	36.08	.76	84.35
⋮	⋮	⋮	⋮	⋮	⋮	⋮
WW1 (0.1)	1	.97	1	1.71	.99	3.24
Average	.99	6.19	.96	10.02	.68	22.75

Tab. 3.2.: Filtre sur le *Pagerank* : analyse du temps de chargement des pages

Chaque thème comprend un grand nombre d'entités. Retrouver toutes ces entités dans les contraintes de temps réel d'une page web est extrêmement difficile, même avec une base de données optimisée. Nous avons ainsi appliqué le filtre des articles défini dans la section 3.4.4. Nous avons comparé le temps de chargement des pages avec et sans filtre pour différents thèmes, et reporté les résultats dans la table 3.2. Grâce au filtre des entités faiblement classées, on constate que le temps de chargement est considérablement réduit, passant parfois de plusieurs dizaines de secondes à quelques centaines de millisecondes. Par ailleurs, la pertinence des résultats affichés aux utilisateurs n'est quasiment pas affectée, dans la mesure où le NDCG@100 indique que le top 100 des résultats présentés n'est quasiment pas modifié.

3.7 Conclusion

Dans cette section, nous avons utilisé les méta-données de Wikipedia pour créer et approvisionner automatiquement un ensemble de thèmes. Nous avons conçu cette solution de façon paramétrable pour permettre aux utilisateurs d'influencer sur le niveau de précision des thèmes (de génériques à spécifiques), ou pour laisser à ces derniers la possibilité de définir manuellement des thèmes en se contentant des phases d'approvisionnement et de classement des articles correspondants.

Dans cette thèse, ces thèmes sont utilisés pour générer automatiquement des questions en rapport avec un sujet donné. Ils servent ainsi, comme nous le verrons dans les sections suivantes, à identifier au sein de bases de connaissances dérivées de Wikipedia des entités ayant une relation significative avec le thème prédéfini.

Nous avons également montré que les différents thèmes obtenus à l'aide de cette approche peuvent être utilisés dans des contextes bien différents, en l'occurence dans tout scénario nécessitant des thèmes connus a priori. Ainsi, pour illustrer l'utilité de ces thèmes dans un cadre externe à la génération de questions, nous avons mis en place un navigateur par thèmes, Fouilla (RAYNAUD et al., 2018a). L'objectif de cette navigation par thème est de filtrer et réorganiser le contenu de bases de connaissances, en ne présentant aux utilisateurs que les éléments qui se rapportent à un thème de leur choix.

D'autres applications pouvant mettre à profit ces thèmes sont également envisageables. On peut imaginer par exemple des systèmes de *recommandation thématique*, qui se serviraient des thèmes et d'un historique de navigation pour recommander automatiquement des articles abordant des thématiques similaires. Dans le contexte de Wikipedia, un système de recommandation pourrait proposer automatiquement

des articles importants pour mettre à jour les pages de méta-données thématiques, comme les Portails ou les Outlines. Dans cette optique, nous avons créé un jeu de données, stocké au format RDF Turtle [11], permettant à toute personne qui le souhaiterait d'utiliser les thèmes que nous avons créés. Ce jeu de données contient l'ensemble des thèmes identifiés, ainsi que l'ensemble des articles de Wikipedia de chaque thème. Pour chaque article, un score est donné pour indiquer sa pertinence vis à vis de chaque thème.

11. datasets-satin.telecom-st-etienne.fr/traynaud/fouilla

4 Modèles de questions : définition et approvisionnement

Introduction

Dans l'approche présentée dans cette thèse, nous définissons et utilisons des modèles de questions pour générer automatiquement des questions à choix multiples. Ces modèles de questions ont été définis en se basant sur l'analyse de la structure et des composants de questions à choix multiples. L'objectif lié à la mise en place de ces modèles est de simplifier le processus de génération de questions, en offrant des modèles à la fois fonctionnels et thématiques.

Ainsi, d'un point de vue fonctionnel, les modèles de questions doivent permettre de générer des questions à choix multiples, incluant énoncé, bonne réponse et mauvaises réponses.

Parallèlement, dans notre contexte de génération de question thématiques, les modèles de questions doivent pouvoir être associés à des thèmes. Cette association a pour objectif d'identifier les modèles les plus pertinents pour un thème donné, et ainsi de générer des questions pertinentes pour ce dernier.

Cette section présente les modèles de questions que nous avons mis en place pour répondre à ces deux problèmes. La section est organisée de la façon suivante. Dans la section 4.1, nous revenons sur la définition et la structure des composants d'une question à choix multiples, afin d'introduire nos modèles de questions. Nous expliquons dans la section 4.2 les différents procédés envisagés pour alimenter un jeu de modèles de questions. Nous présentons ensuite dans la section 4.3 les différentes stratégies envisagées pour établir une corrélation entre modèles et thèmes. Nous terminons ce chapitre par une discussion sur les limites des modèles de questions, dans la section 4.4.

4.1 Modèles de questions : définitions

Notre approche de génération de questions thématiques se fonde sur l'utilisation de *modèles de questions* que nous avons définis. Ces modèles nous permettent à la fois de générer l'énoncé des questions, et d'identifier un ensemble d'entités pertinentes pour construire ces questions. Nous expliquons comment sont construits ces modèles dans ce chapitre 4 et comment ils sont utilisés pour générer des questions dans le chapitre 5. Cette section regroupe et définit l'ensemble des notions nécessaires à la génération de questions et aux modèles de questions.

4.1.1 Questionnaires et Questions à choix multiples

Les questionnaires à choix multiples, et les questions à choix multiples sont deux éléments distincts. On définit ces deux éléments de la façon suivante :

Définition 3: Questionnaire à choix multiples

Un *Questionnaire à choix multiples* est un ensemble de questions à choix multiples relatives à un thème commun.

Le but d'un questionnaire est de fournir un ensemble de questions liées entre elles, permettant de maximiser la couverture des sujets couverts au sein du thème, tout en minimisant l'intersection entre les différentes questions. A l'heure actuelle, les questions générées à l'aide des travaux présentés dans ce manuscrit sont indépendantes, et ne constituent donc pas de questionnaires. C'est pourquoi dans la suite du manuscrit nous traiterons du problème de la génération de *questions à choix multiples* et non de la génération de *questionnaires*.

Nous définissons une question à choix multiples de la façon suivante :

Définition 4: Question à choix multiples

Une question à choix multiples est un 3-tuples(Q, K, D) où :

- Q est un énoncé écrit en langage naturel,
- K est un ensemble de bonnes réponses, constitué au minimum d'un élément,
- D est un ensemble de mauvaises réponses appelées distracteurs.

K et D regroupent l'ensemble des choix proposés à la personne interrogée, K étant l'ensemble des bonnes réponses, et D celui de mauvaises. Le nombre total de ces propositions étant constant et connu à l'avance (e.g. 4 choix), le nombre d'éléments présent dans chaque ensemble varie donc en fonction du nombre présent dans l'autre. Ainsi dans le cas où la question admet deux bonnes réponses $|K| = 2$, le nombre de distracteurs $|D|$ est déterminé par $|D| = 4 - |K|$ soit 2 également.

La solution proposée dans ce manuscrit permet de gérer des questions à choix multiples admettant plusieurs bonnes réponses ($|K| \geq 1$). Cependant, pour des raisons de lisibilité et de difficulté, nous avons volontairement restreint ce nombre à une seule bonne réponse ($|K| = 1$). Par ailleurs, les questions à choix multiples générés comportent systématiquement 4 propositions ($|K| + |D| = 4$), ce nombre étant le plus courant et le plus pertinent pour ce type de question (Graesser et Wisher, 2001). Les propositions sont donc systématiquement composées d'une bonne réponse ($|K| = 1$) et de trois distracteurs ($|D| = 3$). Ces travaux peuvent cependant être facilement généralisés à la génération de questions à choix multiples comportant n propositions, dont plusieurs bonnes réponses.

On définira spécifiquement les distracteurs, i.e. les mauvaises réponses de la question de la façon suivante :

Définition 5: Distracteur

On définit comme *distracteur* (*distractor* en Anglais) une réponse proposée alternativement à la bonne réponse au sein d'une question à choix multiples. Pour être valide, un distracteur ne doit pas être une réponse correcte à la question.

4.1.2 Modèles de questions

Cette thèse a pour vocation la production de questions à choix multiples thématiques. Nous avons identifié dans le chapitre précédent les bases de connaissances dérivées de Wikipedia comme sources de données thématiques permettant la génération de questions. Il est donc nécessaire de déterminer un processus qui permettra de transformer les données issues de ces bases de connaissances en questions à choix multiples. Nous avons vu dans le chapitre 2 que les approches cherchant à générer des questions à partir de bases de connaissances utilisent généralement des modèles qui consistent à contraindre tout ou partie de l'énoncé.

Cependant, nous avons vu que ces approches se limitent pour la plupart à un triplet unique, ou mettent dans une même phrase des triplets relatifs au même sujet, mais

potentiellement n'ayant pas ou que peu de liens les uns avec les autres. Ainsi, les questions obtenues sont généralement très basiques dans leur structure grammaticale (*sujet verbe objet*), ou peuvent ne pas contenir suffisamment d'informations pour permettre aux personnes interrogées d'y répondre correctement. Elles peuvent également être composées d'éléments sans réel rapport avec le coeur de la question.

Deux problématiques scientifiques font ainsi obstacle à la génération de question à choix multiples à partir de bases de connaissances :

1. **La complétude de l'énoncé :** Obtenir un énoncé cohérent qui ne contient ni trop, ni insuffisamment d'information pour répondre à la question.

2. **La verbalisation de l'énoncé :** Générer un énoncé en langage naturel qui prenne en compte les caractéristiques des entités qu'il contient.

Nous apportons une solution à ces problématiques en introduisant le concept de *modèle de questions*. Il s'agit de modèles qui regroupent l'ensemble des informations nécessaires à la génération de questions à choix multiples en utilisant comme source de données une base de connaissances prédéterminée. Chaque modèle de questions a ainsi pour vocation de générer une multitude de questions ayant un sens identique, et ce avec plusieurs énoncés équivalents et prédéfinis avec des variables, au sein desquels seront substituées des entités. Les modèles s'articulent ainsi autour d'une représentation sémantique de leurs énoncés dans une base de connaissances. Cette représentation y apparaît sous la forme d'un sous-graphe connexe de la base de connaissances.

La structure des modèles de questions se justifie par leur capacité à former des questions à choix multiples. En partant d'un modèle de questions, on doit être en mesure d'obtenir un énoncé, une bonne réponse, et des distracteurs. L'existence des modèles de questions se justifie par l'hypothèse qu'il est possible de séparer la sémantique véhiculée par un énoncé de l'énoncé lui même. Ainsi, des questions comme "*Who plays Pâris in Troy ?*", "*Who plays Neo in Matrix*", "*Who plays Lois Lane in Superman*", "*Who plays Raoul Volfoni in Les Tontons Flingueurs*" sont toutes formées à partir d'un énoncé véhiculant la même sémantique : Quel acteur joue le rôle X dans le film Y. On peut donc se servir de ces phrases pour créer un modèle de questions dont l'énoncé serait : "*Who plays* $?pl_1$ *in* $?pl_2$".

On définit ainsi formellement un modèle de questions de la façon suivante :

Définition 6: Modèle de questions

Un modèle de questions est un 3-tuple (E, P, R) où :

— E est un ensemble d'énoncés, chaque énoncé étant individuellement représenté par un ensemble de mots organisés sous la forme d'un arbre syntaxique.

— P est un ensemble de variables. Chaque variable a un identifiant unique que l'on retrouve dans chaque énoncé de E. Lors de la génération d'une question à partir de ce modèle, les variables sont substituées par des entités.

— R est un sous-graphe issu d'une base de connaissances donnée. R est utilisé pour traduire sémantiquement l'énoncé au sein de la base de connaissances. R permet ainsi d'identifier les entités de la base de connaissances qui pourront se substituer aux variables de P lors de la génération des questions. On utilisera le terme de *candidat* dans la suite du document pour désigner ces entités substituantes. Au sein de R, les variables sont explicitement identifiées par leur identifiant unique. R indique également quel élément du sous-graphe constitue la bonne réponse à la question. On utilisera le mot-clé *answer* pour l'identifier.

Cette définition de modèle permet d'inclure plusieurs entités au sein d'un énoncé. On peut ainsi qualifier ces modèles de *modèles n-aire* : chaque modèle est composé de n entités, avec l'entier $n \geq 2$.

Un modèle de questions permet de regrouper plusieurs énoncés jugés équivalents autour d'un sous-graphe de relations unique. Ce dernier représente leur signification au sein d'une base de connaissances.

4.1.3 Variables

Les variables sont les éléments substituables de nos modèles. On les définit ainsi de la façon suivante :

Définition 7: Variable

Une *variable* marque un emplacement substituable au sein des énoncés d'un modèle. Elle est représentée par un identifiant unique qui reste constant au sein de chaque énoncé. Les variables sont remplacées par des entités lors de la génération de la question.

Dans le cadre de la génération de questions, les variables constituent la partie de la phrase déterminante pour identifier la réponse attendue. C'est en effet à partir des entités qui substitueront ces variables dans l'énoncé que la personne interrogée pourra comprendre le sujet de la question, et y répondre.

On distingue ainsi la notion de *variable* de la notion de *candidat*. Les variables marquent des emplacements substituables, alors que les candidats sont les entités qui substitueront les variables correspondantes lors de la génération de la phrase. On définit les *candidats* de la manière suivante :

Définition 8: Candidat

Un *candidat* est une entité de la base de connaissances utilisée pour substituer une variable prédéfinie lors de la phase de génération.

Dans notre exemple situé section 4.1.6, les variables sont représentées par les identifiants $?pl_1$ et $?pl_2$, alors que les candidats sont les entités extraites des n-uplets. Ainsi, *Neo* et *Pâris* sont des candidats pour $?pl_1$, alors que *Matrix* et *Troy* sont des candidats pour $?pl_2$.

4.1.4 Sous-graphes de relations

Dans la définition de modèle utilisée ci-dessus (Définition 6), nous introduisons la notion de sous-graphe de relations. Ces sous-graphes de relations sont utilisés pour lier les modèles avec des bases de connaissances.

On définit les sous-graphes de relations de la façon suivante :

Définition 9: Sous-graphes de relations

Un *sous-graphe de relations* est un sous-graphe connexe issu du schéma d'une base de connaissances donnée. Il permet l'exécution d'une requête SPARQL générée automatiquement, afin d'extraire de la base de connaissances un ensemble de candidats correspondant aux différents composants d'un modèle. Un sous-graphe de relations est valide si et seulement s'il contient de façon exhaustive l'ensemble des relations du sous-graphe correspondant à la question.

Tels qu'ils sont définis ci-dessus, nos modèles de questions peuvent associer plusieurs énoncés à un sous-graphe de relations, ce qui se traduit par la possibilité de paraphraser.

4.1.5 Arbres syntaxiques

Dans la définition de modèle de questions présentée ci-dessus (Définition 6), il est précisé que chaque énoncé de l'ensemble E est stocké sous la forme d'un arbre syntaxique. Nous avons choisi ce format d'énoncés sous la forme d'arbres syntaxiques afin de proposer une manière originale de lexicaliser les relations de la base de connaissances présentes dans le modèle. L'objectif est proposer une solution capable de répondre aux problématiques récurrentes en matière de génération d'énoncés en langage naturel, à savoir le manque de souplesse des modèles existants. L'utilisation des arbres syntaxiques offre en effet certains avantages intéressants, dont les principaux sont énumérés ci-dessous :

— Les informations sur la nature des mots, notamment des précisions sur leur genre et nombre, peuvent influencer le sens de la phrase,

— La modularité de la phrase, notamment pour positionner et substituer les variables, ou modifier la conjugaison ou le temps d'un verbe, le genre et nombre d'un élément, etc.

En se basant sur les travaux existants sur les arbres syntaxiques, on utilise la définition formelle suivante :

Définition 10: Arbre syntaxique

Un *arbre syntaxique* (*dependency parse tree* en anglais) est la représentation d'une phrase donnée sous la forme d'un arbre hiérarchique de mots. Les mots y sont reliés entre eux à l'aide de relations grammaticales. La nature de chaque mot y est explicitée à l'aide d'attributs(AHO, 2003).

Les relations grammaticales entre les mots ont été exhaustivement définies par (DE MARNEFFE et MANNING, 2008). Les arbres syntaxiques que nous utilisons sont conformes à ce standard. On peut citer par exemple la dépendance *aux* qui permet de lier un verbe et son auxiliaire : *aux(has, died)*.

Les attributs qui définissent la nature des mots au sein de la phrase portent le nom de POS tags. Ces attributs sont décrits par (MARCUS et al., 1994). Contrairement aux dépendances définies ci-dessus qui s'appliquent à un groupe de mots, les POS tags ne décrivent la nature que d'un mot à la fois. Par exemple, *NNP(John)* signifie que John est un nom propre.

La figure 4.1 donne un exemple d'arbre syntaxique incluant ses relations et ses POS tags. Chaque mot est séparé des autres par des dépendances qui donnent la nature des liens qui les unissent. Ces dépendances portent plus précisément sur le lien entre un mot et un sous-arbre, ayant pour racine le second mot. Dans cet exemple,

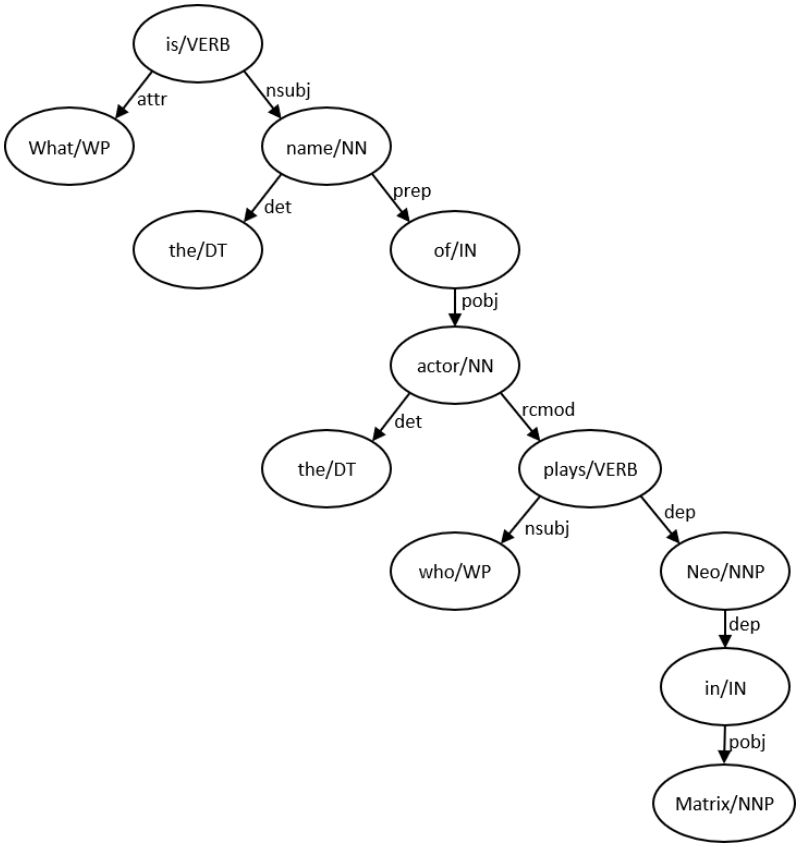


Fig. 4.1.: Exemple d'arbre syntaxique pour la phrase "*What is the name of the actor who plays Neo in Matrix*"

la dépendance "*nsubj(is,name)*" marque la relation entre un verbe au présent et son sujet. On constate que cette relation ne concerne pas que ces deux mots, mais l'ensemble du sous-arbre concerné : "*the name of the actor [...]*". Par ailleurs, on peut noter que dans cet arbre syntaxique, chaque mot est individuellement décrit par son POS-tag, qui indique sa nature. Ainsi, pour le sujet "*the actor*", on marque explicitement la distinction entre le déterminent (DT) : *the* et le nom (NOUN) : *actor*. De même, on aura également une distinction explicite entre les noms communs et les noms propres : on note que le nom commun "*name*" est décrit par le tag *NN*, alors que le nom propre "*Matrix*" est décrit par le tag *NNP*.

4.1.6 Exemple

On peut illustrer l'ensemble des concepts définis ci-dessus à travers l'exemple suivant :

$$T_1 = \begin{cases} E = \{\text{Who plays } ?pl_1 \text{ in } ?pl_2\,?, \text{ Which actor stars } ?pl_1 \text{ in } ?pl_2\,?\} \\ Pl = \{?pl_1, ?pl_2\} \\ R = \begin{cases} \{?pl_1, \text{ portrayed\_in\_films}, ?cvt\} \\ \{?cvt, \text{ actor}, ?answer\} \\ \{?cvt, \text{ film}, ?pl_2\} \end{cases} \end{cases}$$

Cet exemple définit le modèle $T1$, qui permet de générer des questions demandant le nom d'un acteur, en fournissant un nom de personnage et un titre du film. Le modèle $T1$ comporte deux énoncés équivalents, représentés par E : "*Who plays* $?pl_1$ *in* $?pl_2$" et "*Which actor stars* $?pl_1$ *in* $?pl_2$". Au sein de ces phrases, on identifie clairement les deux variables de Pl.

R définit le sous-graphe de relations entre acteur, rôle et film au sein de la base de connaissances (respectivement *actor*, *portrayed_in_films* et *film*). Cet ensemble est utilisé pour générer la requête SPARQL permettant d'identifier des entités correspondant à ces critères. Certains faits complexes ne peuvent pas être décrits par des relations liant directement entre elles toutes les entités invoquées dans la question. On utilise dans ce cas des entités intermédiaires, comme $?cvt$ dans cet exemple, qui sert de lien entre un acteur et un film.

Afin de trouver des entités appropriées, une requête SPARQL est automatiquement générée à partir de R pour interroger la base de connaissances. Dans cet exemple, la requête SPARQL automatiquement générée aura pour résultat un ensemble de triplets ($?answer$, $?pl_1$, $?pl_2$) correspondant respectivement aux acteurs, rôles et films. Ces éléments seront automatiquement remplacés au sein de E pour générer des énoncés de questions.

Dans cet exemple, un ensemble de résultat possibles pour la requête générée à partir de $T1$ pourrait être composé des triples (*Orlando Bloom, Pâris, Troy*), (*Keanu Reeves, Neo, Matrix*), (*Margot Kidder, Lois Lane, Superman*), (*Bertrand Blier, Raoul Volfoni, Les Tontons Flingueurs*). Chacun de ces triples permet de générer deux questions qui correspondent aux deux énoncés définis dans E. Pour le triple (*Orlando Bloom, Pâris, Troy*), les deux questions correspondantes sont : "*Who plays Pâris in Troy ?*" et "*Which actor stars Pâris in Troy ?*". De même, pour le triple (*Keanu Reeves, Neo, Matrix*), les deux questions correspondantes sont : "*Who plays Neo in Matrix ?*" et "*Which actor stars Neo in Matrix ?*". Ici, les entités *Pâris*, *Troy*, *Neo*, *Matrix*, *Lois Lane*, *Superman*, *Raoul Volfoni*, *Les Tontons Flingueurs* sont des candidats.

4.2 Alimentation du jeu de modèles de questions

Pour être utilisables dans un contexte de génération de questions, les modèles de questions doivent être disponibles a priori. Il est donc nécessaire d'alimenter notre ensemble de modèles de questions, et de lui permettre d'évoluer. Nous avons ainsi prévu deux cas d'utilisation permettant d'alimenter cet ensemble : premièrement à travers une édition manuelle assistée de modèles, et deuxièmement en convertissant et intégrant massivement des données existantes. Ces deux procédés ayant pour but d'alimenter une base de modèles, nous détaillons dans un premier temps les moyens utilisés pour stocker les modèles de questions au sein de notre système, puis nous présentons individuellement chacun de nos deux procédés d'alimentation.

4.2.1 Stockage des modèles de questions

Nous avons mis en place une base de données relationnelle, permettant de stocker l'ensemble des données qui forment nos modèles de questions. Le choix de ce format se justifie par un accès rapide aux données, tout en permettant de lier nos modèles à nos thèmes, définis dans le chapitre précédent. Le détail des opérations permettant de corréler modèles et thèmes est détaillé en section 4.3.

Le schéma de la figure 4.2 présente l'ensemble des tables utilisées pour stocker les modèles de questions. Ce schéma se connecte à notre base de données de thèmes, définie précédemment au travers de la table *Ranking_template*, qui stocke le score de pertinence des modèles au sein des thèmes. L'ensemble du schéma est construit autour de la table *Template* permettant de lier entres eux tous les composants d'un modèle de questions. En effet, les modèles de questions étant des éléments complexes, ils ne peuvent pas être stockés dans une table unique. On note ainsi la présence de la table *Sous-graphe*, utilisée pour regrouper l'ensemble des éléments qui composent le sous-graphe de relations du modèle, et la table *Enoncé*, à laquelle sont rattachés tous les éléments qui permettent de construire un énoncé. Les arbres syntaxiques sont des éléments complexes qui ne peuvent pas être stockés sous la forme d'une table unique. C'est pourquoi les tables *Mot* et *Dépendance* sont utilisées pour stocker et relier entre eux tous les éléments qui composent l'arbres syntaxiques d'un énoncé. On peut ainsi stocker les différents mots qui composent les phrases, ainsi que leurs POS tags, tout en les assemblant pour former une phrase à l'aide de la table *Dépendance*, qui respecte l'ordre d'assemblage obtenu par les outils de traitement du langage. De la même façon, le sous-graphe de relations, qui correspond de façon pratique à un ensemble de triples issus d'une base de connaissances est également stcké dans plusieurs tables : les tables *Triple* et *Relation* permettent de stocker et relier entre eux l'ensemble des éléments qui composent ces graphes. La

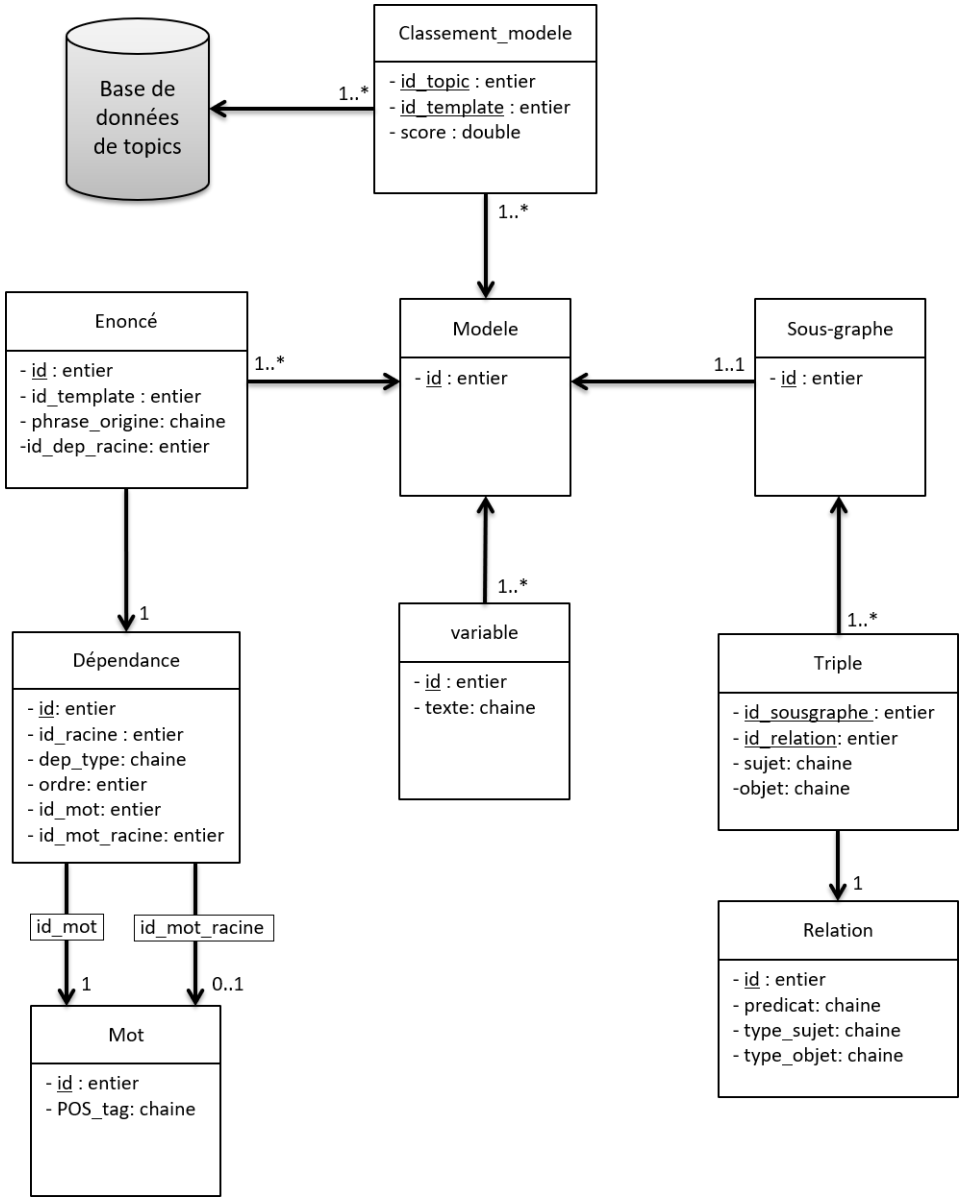


Fig. 4.2.: Schéma relationnel de la base de données de modèles de questions

table *Triple* stocke les informations qui constituent un triple spo, alors que la table *Relation* stocke les informations spécifiques aux prédicats de ces triples spo. Ensuite, le score de pertinence des modèles au sein des thèmes est stocké dans la table *Ranking_template*. Cette table utilise l'identifiant d'un modèle, et l'identifiant d'un thème pour former sa clé primaire, et l'attribut *score* est utilisé pour stocker la valeur du score lui-même.

4.2.2 Editeur assisté de modèles de questions

Nous avons développé un éditeur de modèles permettant à un utilisateur de définir des modèles de questions et de les enregistrer dans la base de modèles. Les

éléments E et Pl consistant à rédiger des énoncés et à y identifier des variables sont relativement simples et ne requièrent pas d'aide particulière. En revanche, concernant l'élément R, il s'agit d'identifier dans une base de connaissances les relations qui représentent sémantiquement l'énoncé saisi. Étant donné que peu de personnes sont familières avec les bases de connaissances et les requêtes SPARQL, cela nécessite la mise en place d'une solution adaptée, sous la forme d'un outil assistant semi-automatique. L'objectif de cet outil est de permettre à un utilisateur de définir entièrement des modèles, même lorsque l'utilisateur n'a pas de connaissances particulières en SPARQL. Nous avons conçu cet éditeur de modèles sous la forme d'une application Web (voir figure 4.3). Avec l'éditeur, la création d'un template requiert trois étapes :

Etape 1 : énoncés. L'utilisateur rédige une ou plusieurs versions paraphrasées de la même question. Il indique explicitement l'emplacement des variables au sein de ces énoncés. A ce stade, les variables se présentent sous la forme de noms d'entités, qui permettront d'identifier à l'étape suivante les entités correspondantes dans la base de connaissances. L'utilisateur définit également la bonne réponse, et si possible, son type, sous la forme d'un type d'entité disponible dans la base de connaissances.

Etape 2 : entités. L'outil recherche au sein de la base de connaissances toutes les entités pouvant correspondre aux variables saisies. Les différentes entités identifiées par l'outil à la fin de l'étape 1 sont affichées à l'utilisateur, dont le rôle est de choisir pour chaque variable l'entité la plus cohérente vis à vis de son énoncé. Notons que dans les cas où aucune ambiguité n'est détectée, cette étape est automatique.

Etape 3 : sélection du sous-graphe. L'outil identifie automatiquement les sous-graphes de la base de connaissances qui permettent de relier entre elles toutes les entités de la question : les variables et la bonne réponse. Dans le cas où il existe plus d'un sous-graphe possible, l'utilisateur choisit le sous-graphe le plus pertinent vis à vis de la question.

La figure 4.3 issue de notre éditeur de modèles, permet d'illustrer le fait que les énoncés sont saisis en utilisant des noms d'entités pour marquer l'emplacement des variables. Cette façon de faire est visuelle pour l'utilisateur, mais sa mise en place s'explique plutôt par le fait que c'est le moyen le plus simple (outre une exploration manuelle de la base) pour identifier le sous-graphe des relations correspondant à la question posée.

Ces 3 étapes de création assistée permettent ainsi à un utilisateur de mettre en place des modèles conformes à notre définition de *modèles de questions* sans aucun prérequis de SPARQL. Cette assistance à la création de modèles permet de vérifier

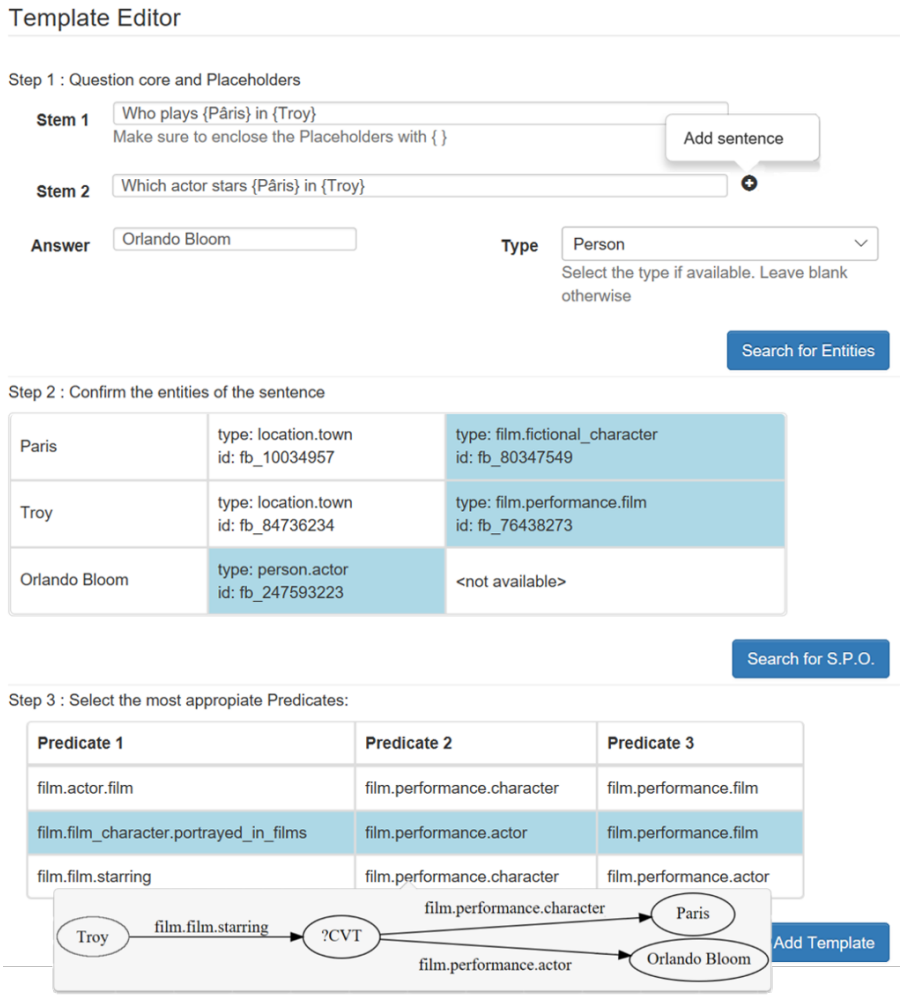


Fig. 4.3.: Outil de création assistée de modèles de questions

à chaque étape la validité des données automatiquement extraites depuis la base de connaissances. L'utilisateur obtient finalement des modèles conformes à ses attentes, tout en ayant la possibilité d'intervenir pour ajuster les actions automatiques. Notre éditeur est actuellement très basique et nous sommes conscients qu'il ne répond pas à tous les critères de qualité que l'on pourrait attendre d'une application professionnelle. Il assure cependant le B-A-BA des besoins pour la création semi-automatique de modèles de questions.

Dans le cadre de la génération automatique de questions se basant sur des modèles, offrir à l'utilisateur la possibilité de créer ou modifier manuellement des modèles est indispensable pour lui permettre de répondre à des besoins spécifiques. Cette méthode reste toutefois contraignante pour l'utilisateur, et rend fastidieuse la création d'un grand nombre de modèles. Dans la section suivante, nous présentons un procédé complémentaire permettant dans une certaine mesure de construire automatiquement un grand nombre de modèles de questions.

4.2.3 Extraction et transformation de données pour former des modèles de questions

Dans le domaine de la *réponse automatique aux questions*, Abujabal et al. ont proposé de réutiliser des questions posées par des utilisateurs de différents forums d'entraide afin de répondre automatiquement à des questions similaires (ABUJABAL et al., 2017a). Ces travaux se concentrent sur Freebase, une base de connaissances stockée en triplestore : chaque élément de la base est stocké sous la forme d'un triplet, composé d'un *sujet*, d'un *prédicat* et d'un *objet* (spo). De façon générale, le *sujet* est toujours une entité Freebase, désignée par un ID unique. Le *prédicat* définit le type de relation qui unit le *sujet* avec l'*objet*. L'objet peut être de différentes natures, à savoir une autre entité (e.g. *Paris, Ville*) ou un littéral (e.g. *2018*).

L'objectif des travaux de Abujabal et al. était de séparer les entités et le corps de la question, dans le but de répondre automatiquement à des questions semblables en s'appuyant sur Freebase. La première étape de leur procédé consiste à extraire les arbres syntaxiques des phrases à l'aide de l'outil Reverb OpenIE (FADER et al., 2011).

Les arbres syntaxiques permettent notamment d'identifier les noms propres et les verbes au sein d'une phrase. Cette décomposition a permis à Abujabal et al. d'identifier les noms propres puis de les associer à des entités Freebase à l'aide d'un dictionnaire spécialisé (ABUJABAL et al., 2017b). Une fois les entités identifiées, Abujabal et al. ont automatiquement identifié les relations entre ces entités au sein de Freebase, permettant d'associer à chaque question Web un template de réponses [1].

Ces questions extraites automatiquement ont ainsi permis à Abujabal et al. de créer un ensemble de *templates de réponses* composés des énoncés, des sous-graphes de Freebase (que Abujabal et al. nomment *Query Templates*), ainsi que de la bonne réponse à la question et des entités identifiées dans la phrase d'origine.

Afin de grouper les templates de réponses qui répondent à des questions similaires, Abujabal et al. ont distingué les concepts de *Template* et de *Utterance-Query Pair*. Les *templates de réponses* qu'ils utilisent sont constitués d'un ensemble de Utterance-Query Pairs, chacune d'entre elles étant le résultat de l'extraction d'une question d'origine. Ces questions d'origine étant des questions posées par des utilisateurs de forums, il est fréquent d'y trouver des reformulations de questions visant à obtenir la même réponse ou au moins le même type de réponse. Ainsi, les templates de

1. On utilisera les termes anglais pour les éléments définis par Abujabal et al. : leur vocabulaire étant proche du nôtre, le lecteur les distinguera plus facilement.

réponses définis par Abujabal et al. peuvent être décrits comme étant un ensemble de Utterance-Query Pairs relatives à un sujet commun.

Les *templates de réponses* présentés par Abujabal et al. ont une structure relativement proche de ceux que nous avons définis pour notre problème de génération automatique de questions. Ils peuvent par conséquent, après transformation, constituer une source intéressante de données pour alimenter automatiquement notre base de modèles de questions. Nous obtenons ainsi la possibilité de générer des questions que l'on peut qualifier de pertinentes dans la mesure où elles ont été posées par des utilisateurs. La figure 4.4 donne un exemple de *template de réponses* mis en place par Abujabal et al.. Un template de réponses se compose de deux éléments : un corps générique utile pour leur problématique de réponse automatique aux questions, suivi de l'une de ses Utterance-Query Pairs, qui fait pour sa part référence à une question posée par un utilisateur.

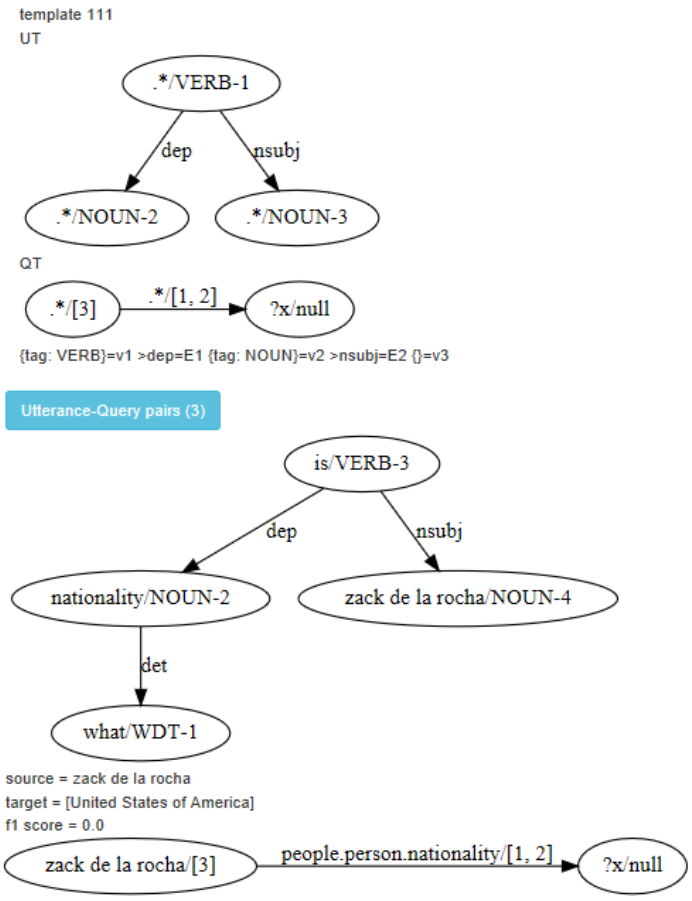


Fig. 4.4.: Exemple de template de réponses utilisé par Abujabal et al. (ABUJABAL et al., 2017a)

4.2.3.1. Extraction depuis les pages Web

Pour être en mesure de réutiliser ces templates de réponses, il était nécessaire de les transformer dans un format que nous sommes en mesure de manipuler. En effet, en nous mettant en relation avec Abujabal et al., nous avons pu obtenir les templates de réponses, mais sous forme de pages Web destinées à la visualisation. Bien que ce format ne soit pas idéal pour extraire les données, il n'en contenait pas moins l'ensemble des informations qui composent les templates de réponses.

Au sein de la page Web, les templates de réponses sont composés de plusieurs éléments :

— un **arbre syntaxique** générique : Il s'agit d'un modèle d'arbre syntaxique qui permet à Abujabal et al. d'identifier quel type de template est le plus approprié lors de l'application de sa méthode de *réponse automatique aux questions*. Cet arbre syntaxique générique n'est pas utile dans le contexte de la génération de questions.

— un **Query Template** générique : Il s'agit d'un ensemble de relations de base de connaissances génériques qui permettent à Abujabal et al. de trouver efficacement la réponse à une question. Ce Query Template générique n'est pas utile dans le contexte de la génération de questions.

— un ensemble d'**Utterance-Query Pairs** : Les Utterance-Query Pairs permettent, au sein du système de *réponse automatique aux questions*, de répondre à une question. Chaque Utterance-Query Pair est composée des éléments suivants :

  — un **arbre syntaxique** : Il est utilisé pour stocker la phrase d'origine de la question. Il est représenté dans la page Web sous la forme d'un graphe SVG.

  — un **Query Template** : Il s'agit des relations qui relient la phrase de l'arbre syntaxique à Freebase. Ces relations se décomposent en triples. Le Query Template est représenté dans la page Web sous la forme d'un graphe SVG.

  — une ou plusieurs **bonnes réponses** : Ces réponses, automatiquement extraites depuis Freebase, indiquent la ou les réponses à la question située dans l'arbre syntaxique.

Afin d'extraire ces templates de réponses depuis les pages Web, nous avons utilisé des expressions régulières pour identifier et extraire chaque sous-partie. Les pages Web ayant été générées automatiquement, les motifs y étaient très réguliers, et donc faciles à extraire.

L'extraction de l'ensemble des templates de réponses s'est ainsi décomposée en trois parties, toutes basées sur les expressions régulières :

1. **Extraction des templates :** La première étape consiste à extraire la totalité du template. Cette opération était possible dans la mesure où une ligne caractéristique marque le début de chaque template. Pour extraire le contenu d'un template de réponses, il fallait ainsi extraire le contenu situé entre cette ligne et la suivante.
2. **Extraire l'ensemble des Utterance-Query Pairs :** Au sein des templates de réponses, seules les Utterance-Query Pairs sont vraiment intéressantes dans le contexte de la génération de questions. En identifiant la ligne qui marque le début de la liste d'Utterance-Query Pairs au sein de chaque template, nous avons pu isoler ces éléments, et les extraire.
3. **Extraire le contenu des Utterance-Query Pairs :** Une fois les Utterance-Query Pairs isolés, l'étape finale consistait à parser leur contenu, situé au sein de graphes SVG. Les expressions régulières ont ici permis ici d'identifier chaque nœud et chaque arête individuellement. Cela nous a permis de reconstruire les différents graphes par la suite.

4.2.3.2. Transformation en modèles de questions

Les templates de réponses de Abujabal et al. n'étaient pas dans un format compatible avec nos modèles de questions. Il a ainsi fallu convertir ces derniers dans un format plus approprié, de façon à pouvoir les utiliser pour générer des questions. Cette transformation s'est faite au fur et à mesure de l'extraction des templates de réponses depuis les pages Web fournies par Abujabal et al..

Les templates de réponses de Abujabal et al. se présentent sous la forme d'un template générique, auquel est associé un ensemble de Utterance-Query Pairs. Les Utterance-Query Pairs contiennent ainsi les informations issues d'une phrase en particulier. Les phrases similaires, du point de vue de Abujabal et al., sont ensuite regroupées dans un template. Ainsi, on devrait théoriquement retrouver le schéma utilisé dans nos modèles de questions, à savoir un sous-graphe issu d'une base de connaissances, auquel correspond un ensemble d'énoncés. Dans la pratique, les templates de réponses de Abujabal et al. ne correspondent pas à ce schéma. En effet, les templates de réponses regroupent les énoncés en fonction de la disposition des mots dans les phrases d'origine (i.e. en fonction de la composition de l'arbre syntaxique). Ainsi, on peut trouver des phrases qui n'ont rien à voir entre elles dans un même template de réponses, et des phrases très proches, avec un Query Template identique, dans deux templates de réponses différents.

Afin de transformer ces templates de réponses en modèles de questions, nous avons ainsi fait abstraction de la notion de template définie par Abujabal et al., pour ne

conserver et traiter que les Utterance-Query Pairs. En effet, l'élément central de nos modèles de questions est le sous-graphe de relations et non l'énoncé de la question. Cette transformation se compose de 3 étapes :

1. **Mettre à jour les arbres syntaxiques** des templates de réponses : Il s'agit de reprendre les arbres syntaxiques utilisés par Abujabal et al. pour les transformer dans un format plus adapté. La section 4.2.3.3 détaille spécifiquement les différentes étapes qui permettent de résoudre cette tâche.
2. **Généraliser les arbres syntaxiques** et les Query Templates : Afin d'être compatibles avec les éléments que nous stockons au sein de notre base de données de modèles de questions, nous avons mis à jour les éléments qui composent les template de réponses de Abujabal et al., pour y intégrer la notion de variable. La section 4.2.3.4 détaille spécifiquement les différentes étapes qui permettent de résoudre cette tâche.
3. **Regrouper les Utterance-Query Pairs** par sous-graphe de relations : La dernière étape a consisté à regrouper les éléments possédant un sous-graphe de relations identique de façon à grouper les énoncés ayant la même signification. Ceci permet de correspondre à notre définition de modèles de questions, qui inclut les paraphrases.

Ces différentes étapes nous ont ainsi permis de transformer les templates de réponses de Abujabal et al. en modèles de questions. On retrouve ainsi tous les éléments qui forment nos modèles de questions :

— Le sous-graphe de relations R : Il est formé à partir des Query Template de une ou plusieurs Utterance-Query Pairs regroupées lors de l'étape (3).

— L'ensemble d'énoncés E : Constitué d'au moins un élément, l'ensemble d'énoncés est formé par les arbres syntaxiques mis à jour à l'étape (1), et regroupés sur la base de Query Template identique à l'étape (3).

— L'ensemble de Variables P : Créé en remplaçant les entités présentes dans l'arbre syntaxique et le Query Template lors de l'étape (2).

4.2.3.3. Alignement

Pour analyser la structure des phrases postées par les utilisateurs, Abujabal et al. ont décomposé ces dernières en arbres syntaxiques. Comme mentionné ci-dessus, ces arbres syntaxiques sont utilisés dans sa tâche de réponse automatique aux questions pour identifier rapidement des questions de même composition. Cependant, les arbres syntaxiques construits par Abujabal et al. ne correspondent pas à ceux que nous avons nous-même mis en place pour stocker les énoncés de nos modèles de

questions, en raison de l'utilisation de deux outils de traitement automatique du langage différents. Le problème réside ici principalement dans le fait que Abujabal et al. ne stockent les énoncés des questions que sous la forme de ces arbres syntaxiques, sans leur associer les phrases d'origines. Il nous était donc impossible d'utiliser ces données en l'état, n'ayant pas d'énoncé valide associé aux questions. Afin de résoudre ce problème, nous avons réutilisé le jeu de questions contenant l'ensemble des phrases saisies par les utilisateurs pour construire des arbres syntaxiques cohérents avec notre définition. Ainsi, pour associer chaque template de réponses avec sa phrase d'origine, nous avons élaboré une méthode ***d'alignement***. Cette méthode consiste à décomposer les phrases d'origine et les phrases des templates en un ensemble de mots, triés alphabétiquement, et à les comparer deux à deux. La mise en application de la méthode a montré que parmi les 1900 phrases qui composent les templates de réponses de Abujabal et al., il n'existe pas deux phrases contenant exactement le même ensemble de mots, ce qui a donc permis de retrouver la phrase d'origine de chaque template de réponses.

Indirectement, le fait de connaître les phrases d'origine liées à chaque template de réponses nous a permis d'évaluer automatiquement la validité de notre approche de génération en langage naturel des énoncés, présenté en section 5.1.4. L'évaluation automatique de ce procédé est présentée en section 6.4.

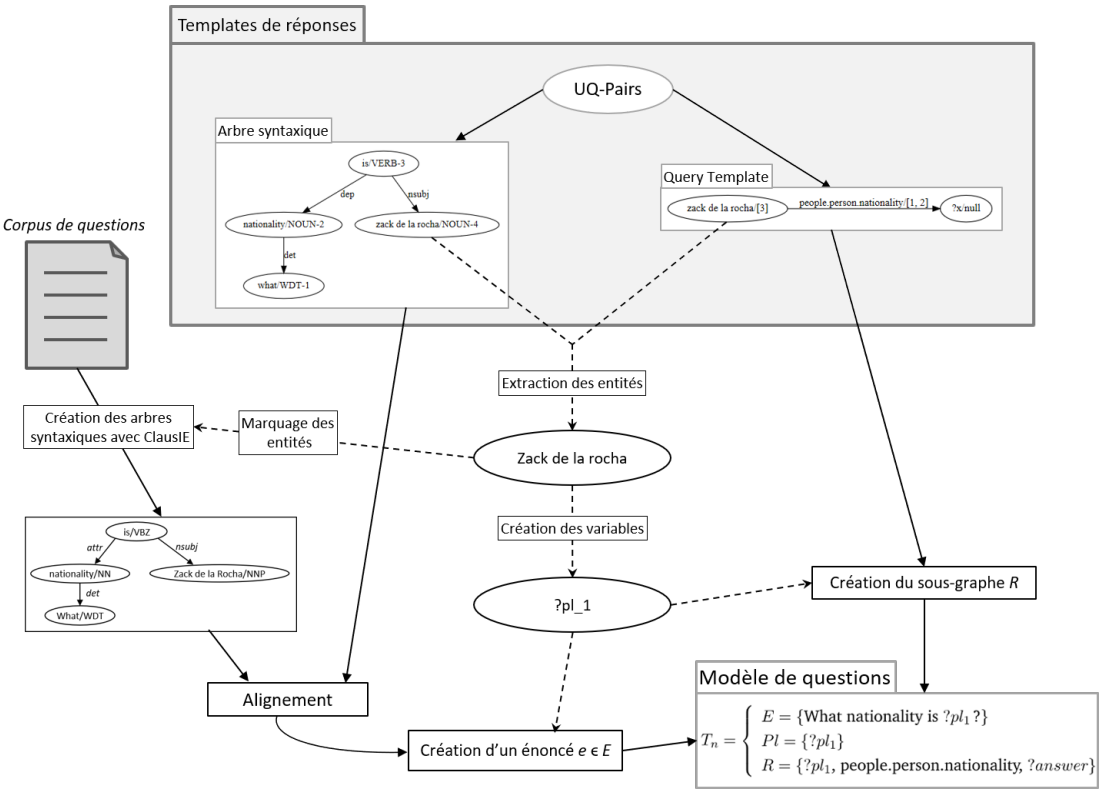


Fig. 4.5.: Processus de transformation des templates de réponses en modèles de questions

4.2.3.4. Positionnement des variables

La dernière étape de l'intégration des templates de réponses de Abujabal et al. au sein de nos modèles de questions a consisté à remplacer les entités présentes au sein de l'énoncé et du Query Template par des identifiants de variables. En se basant sur le contenu des Query Templates des templates de réponses, nous avons ainsi été en mesure, pour chacun d'entre eux, d'identifier la liste exhaustive des entités utilisées dans la question d'origine. Nous avons par la suite substitué automatiquement ces entités dans le Query Template et dans l'arbre syntaxique, tout en ajoutant au fur et à mesure ces variables dans l'ensemble P du modèle de questions correspondant.

La figure 4.5 illustre ce procédé de transformation des templates de réponses en modèles de questions. On note que l'ensemble des entités à remplacer se trouve dans le Query Template. Dans cet exemple $?x$ désigne la bonne réponse, et *Zack de la rocha*, une entité. Pour effectuer la conversion en modèle de questions, on remplacera ces éléments par les variables conformes à notre définition. *Zack de la rocha* est ainsi transformé en $?pl_1$, et $?x$ en $?answer$. Ces modification sont ensuite également répercutées sur notre ensemble d'énoncés E.

4.2.4 La nécessité des modèles de questions

Dans l'approche présentée dans ce manuscrit, les modèles de questions sont nécessaires pour générer des questions à choix multiples. La variété des questions pouvant être générées dépend donc directement du nombre de modèles de questions disponibles. Dans cette optique, nous proposons donc deux solutions permettant d'alimenter la base de modèles de questions, une première automatique, et une seconde plus manuelle.

Dans le contexte de génération de questions thématiques, l'étape suivante consiste à associer les modèles de la base avec des thèmes. En effet, dans un cas comme dans l'autre, cette association n'est pas réalisée manuellement, mais automatiquement en se basant sur les éléments qui composent les modèles de questions. Il est donc important et nécessaire de mettre en place une mesure de corrélation entre modèles et thèmes afin de refléter pour chaque modèle dans quels thèmes il est le plus pertinent, et inversement, pour chaque thème, les modèles les plus pertinents. La mise en place de cette mesure de corrélation est détaillée dans la section suivante.

4.3 Association des modèles de questions aux thèmes

L'objectif des travaux présentés dans cette thèse est de générer des questions en rapport avec un thème donné. Nous avons identifié dans le chapitre 3 un ensemble de thèmes permettant de représenter des thèmes sous la forme d'un ensemble d'entités triées. Pour utiliser nos modèles de questions dans un contexte thématique, il est nécessaire d'établir une corrélation entre modèles de questions et thèmes.

Le calcul de cette corrélation entre modèles et thèmes est en effet une étape très importante de notre générateur de questions thématiques, et ce pour plusieurs raisons. Premièrement, il faut restreindre le choix lors de la sélection du modèle. Sans cette mesure de corrélation, il serait en effet nécessaire de comparer les résultats obtenus avec l'ensemble des modèles à chaque génération de question. Or l'opération consistant à identifier des candidats à partir d'un modèle est particulièrement coûteuse, comme nous le verrons dans la section 5.1.2. Deuxièmement, on veut garantir la qualité des résultats obtenus lors de la phase de génération de questions. En effet, une corrélation incorrecte conduirait à la sélection de modèles inadaptés, ce qui aurait un impact néfaste sur la qualité des questions générées.

Établir cette corrélation entre modèles de questions et thèmes n'est cependant pas une tâche triviale. La difficulté de cette tâche réside principalement dans la quantité limitée d'informations disponibles pour chaque modèle. En effet, les seules informations disponibles pour effectuer cette mesure sont les énoncés, et le sous-graphe de relations. Par ailleurs, nous avons vu dans la section 4.2 que des modèles de questions peuvent être ajoutés progressivement depuis différentes sources, manuelles ou automatiques. De nouveaux modèles pouvant être ajoutés à tout moment, par des enseignants par exemple, il est préférable que la solution permettant d'établir la mesure de corrélation modèle/thème soit capable d'établir cette mesure vis-à-vis des informations présentes dans chaque modèle de questions indépendamment des autres, pour éviter de devoir recalculer la totalité des valeurs à chaque nouvel ajout (en opposition au *Pagerank* par exemple qui doit être recalculé intégralement à chaque ajout).

Nous avons ainsi parallèlement testé deux stratégies indépendantes destinées à mesurer la corrélation entre les thèmes et les modèles, la première étant basée sur les énoncés, et la seconde sur les sous-graphes de relations.

4.3.1 Stratégie 1 : Comparaison des énoncés et des thèmes avec LSA

A l'aide d'une analyse manuelle des phrases composant les énoncés des modèles de questions, nous avons constaté que ces dernières contenaient régulièrement des mots, ou des morceaux de phrases manifestement liés à un thème. Ce constat nous a amenés à formuler l'hypothèse suivante :

Hypothèse 1 : *L'appartenance d'un modèle de questions à un thème peut être déterminée en mesurant la distance sémantique entre les énoncés de ce modèle, et un ensemble de mots représentatifs du thème.*

Pour vérifier cette hypothèse, nous avons choisi de mesurer la similarité entre les différents énoncés d'un modèle de questions et les mots significatifs d'un thème. Cette mesure est effectuée à l'aide de LSA. Latent Semantic Analysis (WIEMER-HASTINGS et al., 2004) (*LSA*) est une mesure permettant de calculer la similarité sémantique entre deux éléments. La *similarité sémantique* entre deux entités est représentée par un indice de corrélation entre les deux entités. Cette similarité est généralement mesurée sur une échelle de 0 à 1, où 0 signifie que les deux éléments n'ont aucun point commun, et 1 qu'ils sont identiques (FENG et al., 2017).

Nous avons choisi LSA, et plus particulièrement la méthode des *distances cosinus*, car cette mesure permet d'obtenir un score de similarité significatif entre deux chaines de caractères, et ce, même si ces dernières sont de longueurs très différentes (SINGHAL, 2001).

Dans notre cas, calculer la similarité sémantique entre les énoncés des modèles de questions, et le contenu des thèmes permet d'attribuer à chaque modèle un score de corrélation avec chaque thème. Ce score détermine ainsi le niveau de pertinence du modèle vis-à-vis du thème. Pour effectuer cette mesure, nous avons utilisé le procédé suivant : nous avons rassemblé l'ensemble des contenus textuels du top-k des articles de Wikipedia les mieux rankés au sein du thème, suivi de l'en-tête des n meilleurs articles suivants. Sur Wikipedia, l'entête est composé par un ou deux paragraphes qui débutent généralement un article, c'est souvent un résumé du contenu. Cet entête contient des informations générales mais importantes pour l'article concerné, ce qui justifie son utilisation dans notre mesure de similarité sémantique. Pour maximiser la pertinence de la comparaison, nous avons filtré et *lemmatisé* [2] le contenu des chaines de caractères afin de ne comparer que les éléments pertinents.

2. La lemmatisation est une opération permettant de substituer chaque mot d'une phrase par son lemme (forme canonique). Dans notre cas, ce procédé a permis de considérer comme similaires les mots ayant des racines communes, et donc d'augmenter la pertinence de la mesure de similarité.

Ainsi, chaque thème t est construit à partir de la formule suivante :

$$Theme(t) = Lemma\left(\sum_{j\to 1}^{k}(entete_j) + \sum_{j\to 1}^{n}(contenu_j)\right)$$

On calcule ensuite le score de corrélation entre modèles de questions et thèmes à l'aide de la fonction $st(m,t)$. Cette mesure calcule la similarité sémantique entre les énoncés E_m d'un modèle m et le texte composant le thème t :

$$st(m,t) = LSA(E_m, Theme(t))$$

Bien que cette stratégie ait donné quelques résultats cohérents, sa précision était manifestement insuffisante pour permettre à elle seule de déterminer l'appartenance d'un modèle à un thème. En effet, une analyse minutieuse des différents scores de corrélation obtenus entre modèles et thèmes nous a permis d'écarter l'utilisation de cette seule stratégie pour déterminer le lien modèle/thème. Nous avons donc exploré une seconde voie indépendante, détaillée dans la stratégie 2.

4.3.2 Stratégie 2 : Mesure du Pagerank des relations du sous-graphe

En observant la composition des relations utilisées dans les différentes bases de connaissances dérivées de Wikipedia, nous avons constaté que les intitulés de ces relations sont généralement explicites dans leur nommage. Sur Freebase par exemple, cet intitulé est composé d'un ensemble hiérarchique d'éléments (exemple : *law.court.judges*). Ce constat nous a amenés à formuler l'hypothèse suivante :

Hypothèse 2 : *L'appartenance d'un modèle de questions à un thème peut être déterminée en se basant sur les intitulés des relations du sous-graphe de relations.*

Cette hypothèse s'appuie sur le fait que les différentes relations utilisées pour représenter les énoncés au sein d'une base de connaissances permettent de traduire sémantiquement ces énoncés au sein de la base. C'est en l'occurence cette sémantique qu'il est intéressant de lier aux thèmes, pour permettre, au moment de la génération, de sélectionner un modèle pertinent vis-à-vis du thème.

Chaque base de connaissances a une représentation des données qui lui est propre. Le nom donné aux classes, ou les intitulés des relations de la base sont définis par un schéma spécifique. Pour que cette approche basée sur les informations présentes dans le sous-graphe de relations puisse fonctionner, il est nécessaire que les mots-clés employés pour décrire la base soient aussi précis que possible. En ce qui concerne les bases de connaissances dérivées de Wikipedia, les différents schémas se présentent de la façon suivante :

— **Freebase :** Le schéma de la base de connaissances est extrêmement détaillé. Les relations et les entités sont décrites par des intitulés hiérarchiques utilisant des mots-clés très précis, et donc parfaitement compatibles avec cette approche. (Exemple : *law.court.judges*)

— **Wikidata :** Les relations sont identifiées au sein de Wikidata par un simple identifiant (exemple : *P398*). Cependant, Ces identifiants font référence à des intitulés complets qui peuvent être utilisés dans le cadre de cette approche (exemple : *child astronomical body*). Chaque propriété est également accompagnée d'une decription qui peut être utilisée pour mesurer l'appartenance au thème.

— **Yago :** Les relations décrites dans Yago ne contiennent pas suffisamment d'informations pour être utilisées individuellement (exemple : *wroteMusicFor*). Cependant, les types utilisés y sont extrêmement précis et peuvent être utilisés en complément de la relation pour déterminer le thème (exemple : *wikicat_Medieval_Tunisian_mathematicians*).

— **DBpedia :** Les relations décrites dans DBpedia contiennent peu d'informations (exemple : *numberOfStudioAlbums*). Par ailleurs, les libellés des types utilisés dans l'ontologie sont insuffisants pour envisager une approche similaire à Yago. Cependant, l'organisation hiérarchique des types permet de se baser sur les types parents pour identifier plusieurs libellés qui peuvent être assemblés pour déterminer le thème. (Exemple : *Event.NaturalEvent.SolarEclipse*).

Une adaptation est nécessaire à cette étape pour chaque base de connaissances utilisée lors de l'association modèles/thèmes. Nous présentons ici les travaux réalisés à partir de Freebase.

Afin de vérifier cette seconde hypothèse, nous avons tout d'abord converti les différentes relations d'un modèle donné en un ensemble de mots-clés, en se basant sur les intitulés de ces relations. Chaque intitulé permet d'obtenir un ou plusieurs mot-clés. Ensuite, nous vérifions si ces différents mots-clés correspondent à des articles de Wikipedia. Lorsque c'est le cas, nous pouvons réutiliser la mesure de Pagerank définie dans le chapitre 3 pour associer à chaque mot-clé un score individuel de corrélation avec le thème, donné par la formule $sa(i,t)$. Le score de corrélation entre un modèle de questions et un thème est enfin déterminé en calculant la moyenne des scores des mots-clés dans le thème.

Illustrons ce procédé a l'aide d'un exemple : le sous-graphe de relations permettant de déterminer le nom du juge suprême d'une cour donnée est représenté dans Freebase par deux relations, respectivement *law.court.judges* et *law.judicial_tenure.judge*. Ces deux relations peuvent être décomposées en un ensemble de mots : "{*law, court, judges, judicial, tenure, judge*}". Dans cet exemple, les mots-clés *court*, *judge*, et *law* référencent directement des articles de Wikipedia, auxquels correspondent pour chacun un score sa par thème. Par ailleurs, les mots *judicial* et *tenure* sont respectivement redirigés vers les articles *judiciary* et *Academic tenure* par Wikipedia, pour lesquels nous avons également un score sa par thème. Au final, seul le mot *judges* ne peut pas être utilisé pour calculer notre coefficient de corrélation modèle/thème.

Dans cet exemple, le score de corrélation st entre notre modèle exemple, que nous appelerons m_{ex}, et un thème t sera ainsi calculé de la façon suivante :

$$st(m_{ex}, t) = \frac{sa(\text{law}, t) + sa(\text{judge}, t) + sa(\text{court}, t) + sa(\text{judiciary}, t) + sa(\text{academic\_tenure}, t)}{5}$$

De façon générale, pour un modèle m et un thème t, on mesure le score de corrélation modèle/thème $st(m, t)$ de la façon suivante :

$$st(m, t) = \frac{\sum_{p \to 1}^{|P|} (sa(p, t))}{|P|}$$

où

— p représente chaque mot-clé de l'ensemble P, construit à partir des relations qui composent le sous-graphe de relations du modèle m

— $sa(p, t)$ est le score donné à chaque mot-clé dans le thème t, précédemment calculé dans la section 3.4.2.

Nous avons constaté que la mise en place de cette seconde stratégie a permis de considérablement augmenter la précision de l'association des modèles avec les thèmes.

4.4 Limites des modèles de questions

Bien qu'offrant un format avantageux pour la génération automatique de questions à choix multiples, les modèles de questions ont toutefois certaines limites qu'il convient d'expliciter. En effet, lors de la mise en place des modèles de questions, nous avons souhaité mettre en place une solution aussi générique que possible, c'est-à-dire adaptable dans un maximum de scénarios de génération de questions à partir de bases de connaissances. C'est donc dans cette optique que nous avons envisagé ce couplage entre des sous-graphes de relations et des arbres syntaxiques. Cependant, certaines contraintes limitent une utilisation générique de ces modèles. Ces limites peuvent se résumer en deux points : 1. la dépendance à une base de connaissances et 2. la corrélation modèle/thème. Les deux sections suivantes les détaillent.

4.4.1 Dépendance à une base de connaissances

Dans la définition actuelle de nos modèles de questions, nous avons intégré un sous-graphe de relations basé sur une et une seule base de connaissances. Ce sous-graphe de relations est au centre du modèle dans la mesure où il permet de représenter la sémantique véhiculée par la question. Il est également au centre de sa relation avec les thèmes, point que nous discutons plus en détails à la section suivante (section 4.4.2).

Le choix de ce format a été fait pour simplifier l'utilisation lors de la phase de génération de questions. Il permet en effet d'anticiper le format des résultats de la requête SPARQL, et ainsi d'utiliser si nécessaire des éléments spécifiques à la base de connaissances spécifiée. En contrepartie, ce format peut effectivement résulter en la création de plusieurs modèles de questions équivalents : un par base de connaissances.

Il est important également de préciser que le nombre de variables pouvant figurer dans les énoncés d'un modèle de questions est fortement dépendant de la structure de la base de connaissances associée à son sous-graphe de relations. En effet, si cette dernière n'est pas en mesure de gérer des relations n-aires, les sous-graphes de relations seront limités à 1 seul triplet, et les énoncés seront ainsi limités à des interrogations sur des faits simples.

Enfin, il n'existe pas de méthode automatique permettant de déterminer si le sens de l'énoncé est effectivement cohérent avec le sous-graphe de relations du modèle.

Cela signifie que, dans le cas de l'édition manuelle de modèles notamment, cette vérification sera à la charge de l'utilisateur, et donc d'appréciation humaine.

4.4.2 Corrélation modèle/thème

Les bases de connaissances dérivées de Wikipedia n'incluent pas nativement la notion de thèmes. Ainsi, pour utiliser les modèles de questions dans un contexte thématique, il est nécessaire d'analyser la totalité des candidats potentiels, et de les filtrer vis à vis du thème choisi. Ces opérations de requête et de filtrage sont particulièrement coûteuses, et nécessitent donc d'être limitées autant que possible. Dans cette optique, il est important d'associer aux modèles une valeur thématique, permettant de limiter les opérations coûteuses aux modèles les plus prometteurs.

Cependant, cette association modèles/thèmes ne peut se faire que sur la base du contenu exclusivement présent au sein de chaque modèle, à savoir les énoncés, et le sous-graphe de relations. C'est pourquoi dans la section 4.3, nous avons comparé plusieurs stratégies basées sur l'un ou l'autre de ces éléments. Ces mesures, qui permettent ainsi de déterminer la corrélation de chaque modèle avec chaque thème, n'ont ainsi pas pour objectif de déterminer précisément la pertinence finale de la question, mais plus de restreindre la recherche aux éléments prometteurs.

On peut cependant noter que la stratégie 2 a donné les meilleurs résultats pour établir cette corrélation modèle/thème, elle se base sur le sous-graphe de relations du modèle. Le sous-graphe de relations est à l'heure actuelle lié à une base de connaissances unique pour chaque modèle, et chaque base de connaissances possède une ontologie et des relations qui lui sont propres. L'enjeu est donc de trouver, au sein de ces données, des éléments suffisamment explicites pour permettre d'établir une corrélation avec le thème. Une des limites de nos modèles de questions est ainsi liée à l'identification de ces données explicites, ce qui nécessite une adaptation spécifique pour chaque base utilisée. Nous avons ainsi listé dans la section 4.3.2 les éléments permettant d'établir une corrélation entre les différentes bases de connaissances dérivées de Wikipedia et les thèmes. Etant dérivées de Wikipedia, ces bases de connaissances ont une ontologie suffisamment explicite pour permettre d'établir cette corrélation modèle/thème. Cependant, la garantie est moindre dans le cas d'une généralisation du processus à d'autres bases de connaissances.

4.5 Conclusion

Les modèles de questions que nous avons définis dans cette approche nous permettent de compléter significativement l'état de l'art dans le domaine de la génération de questions à partir de bases de connaissances. Dans les sections 2.1, et 2.5, nous avions présenté les différentes limites des approches existantes, et notamment les problèmes liés à la verbalisation des prédicats, et à la difficulté d'établir des liens sémantiques entre les différents triplets d'une même entité.

Dans notre cas, chaque modèle de questions regroupe toutes les informations nécessaires à la création de l'énoncé correspondant, ainsi que la représentation sémantique de cet énoncé au sein d'une base de connaissances, à travers un ensemble de relations n-aires. Ce système de modèle offre ainsi un compromis permettant d'apporter une solution à ces problèmes au détriment de la généricité, dans la mesure où les modèles doivent être connus a priori lors de la génération. On peut ainsi considérer que dans cette approche le problème de la génération de questions à choix multiples est ramené au problème de créer des modèles de questions.

Dans cette optique d'approvisionnement de modèles, nous avons proposé deux approches destinées à enrichir l'ensemble de modèles existants. Une première approche manuelle, présentée en section 4.2.2, permettant à un utilisateur de saisir manuellement les différents composants d'un modèle. Nous avons complété cette approche manuelle à l'aide d'une approche automatique, présentée en section 4.2.3, afin de créer automatiquement des modèles de questions en transformant et en adaptant des données existantes initialement destinées au domaine de la réponse automatique aux questions.

Les problèmes de complétude et de validité des énoncés restent des problèmes de recherches ouverts, pour lesquelles aucune solution satisfaisante n'existe à l'heure actuelle. C'est pour cette raison que les modèles de questions que nous proposons dans cette approche sont systématiquement construits à partir de données ayant été manipulées par des humains. En effet, si ces notions sont difficiles à intégrer dans des processus automatisées, elles sont naturelles et intuitives chez les êtres humains. C'est également dans cette optique que nous n'avons par exemple pas souhaité utiliser les réseaux de neurones, ou les approches de modèles classiques, pour lesquelles il n'y a aucune garantie. Ainsi, les modèles de questions importés à partir de l'éditeur que nous proposons, ou importés à partir des template de réponses de Abujabal et al. ne sont pas infaillibles, comme toutes productions humaines, mais nous les considérons plus fiables que des équivalents générés automatiquement.

5 Construction des questions à choix multiples

Dans ce manuscrit, nous présentons notre générateur automatique de questions à choix multiples thématiques. Nous avons détaillé dans les chapitres précédents de quelle façon nous avons identifié automatiquement un ensemble de thèmes afin de regrouper et de classer des entités par thème, et comment nous avons défini et approvisionné des modèles de questions. Ce chapitre détaille le procédé utilisé pour générer les questions à choix multiples en rapport à un thème donné, en se basant sur les modèles de questions définis précédemment.

Dans la littérature, un certain nombre de critères récurrents sont majoritairement utilisés pour juger le succès de la génération de l'énoncé d'une question. Nous avons vu en section 2.4 que la complétude de l'énoncé ainsi que l'exactitude de sa composition grammaticale et syntaxique sont jugés comme important par la plupart des approches (AL-YAHYA, 2014) (BAILEY et al., 1998). En effet, pour permettre à un utilisateur de répondre efficacement à la question, l'énoncé doit être cohérent et compréhensible au vu du langage dans lequel il est généré.

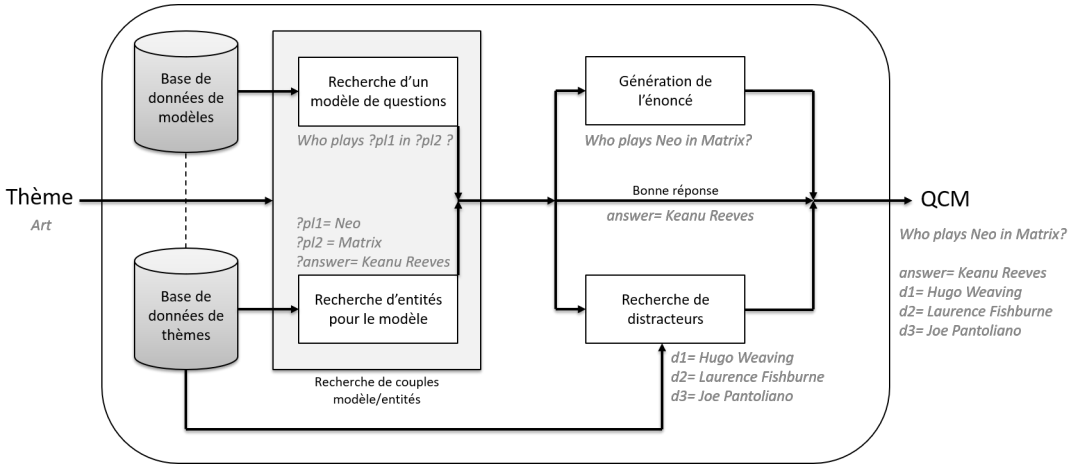


Fig. 5.1.: Étape 3 : Génération de l'énoncé d'une question, et de ses distracteurs

Ce chapitre explique ainsi comment nous utilisons les modèles de questions définis dans le chapitre 4 pour générer en langage naturel un ensemble de questions à choix multiples. Nous expliquons ainsi dans une première partie les procédés utilisés pour générer un énoncé de question dans une thématique donnée, et nous expliquons dans une seconde partie comment nous identifions un ensemble de distracteurs

pertinents. La figure 5.1 donne un aperçu des différentes étapes qui permettent au générateur de construire les questions. On peut ainsi constater le rôle joué par les bases de données de thèmes et de modèles définies dans les étapes précédentes. Un exemple permet d'illustrer le fonctionnement de chacune des étapes de cette figure. On voit ainsi comment à partir du thème "*Art*", on génère la question "*Who plays Neo in Matrix ?*".

5.1 Génération en langage naturel des énoncés

Dans le chapitre 4, nous avons décrit l'intérêt d'utiliser des modèles de questions adaptés à la génération automatique de questions. Cette section décrit les solutions mises en oeuvre pour utiliser ces modèles et plus particulièrement la façon dont ils sont utilisés pour générer un énoncé de question. Nous expliquons ainsi tout particulièrement de quelle façon sont identifiés les candidats qui substitueront les variables dans l'énoncé. En se basant sur la littérature, nous avons sélectionné les critères suivants pour juger de la validité des candidats au sein d'un énoncé :

1. **la correspondance de type :** Les sous-graphes de relations de nos modèles de questions nous permettent d'obtenir des informations sur le type des variables. Ce type doit être respecté par les candidats. Il faut donc limiter la recherche à un type précis, de façon à maximiser la pertinence des candidats. Par exemple, dans la question *Which athlete won the award [...]*, le type *person.athlete* sera plus pertinent que son type parent *person*. Par conséquent, limiter les candidats aux éléments ayant un type proche, voire identique au type d'origine, garantit avec une plus forte probabilité une intégration réussie du candidat dans l'énoncé de la question. Ce critère est probablement le plus important dans la mesure où il est déterminant pour la compréhension globale de la phrase générée.
2. **la correspondance de genre :** Quand les variables représentent des personnes, il est préférable de respecter le genre de l'entité d'origine de la question. En effet, si dans la plupart des cas la tournure permet de substituer les variables de l'énoncé sans que le genre n'impacte son contenu, certaines situations peuvent résulter en un énoncé grammaticalement erroné. C'est par exemple le cas avec un modèle construit à partir de la question "*Where did William Shakespeare perform most of his plays*" qui donnerait le modèle "*Where did* $?pl_1$ *perform most of his plays*". Dans ce cas, l'énoncé nécessite la mise en place d'un candidat masculin pour être en accord avec le possessif "*his*". Ce problème est particulièrement difficile à résoudre, dans la mesure où les bases de connaissances ne contiennent généralement pas de distinction de genre dans les propriétés de leurs entités. En ce qui concerne Freebase notamment, bien que le type *person* permettant de caractériser les êtres humains soit présent dans la base, aucune indication concernant leur genre de ne s'y trouve. Contrairement au critère précédent, si ce critère n'est pas respecté, l'énoncé reste intelligible.
3. **la correspondance de nombre :** Certaines entités peuvent apparaître dans la phrase d'origine sous une forme plurielle. Par exemple la phrase "*In which state are playing the New York Giants*" est formée au pluriel en raison de l'entité "*New York Giants*", dont l'intitulé est exprimé au au pluriel alors qu'il ne correspond qu'à une seule entité de la base de connaissances. Le modèle

obtenu à l'aide de cette phrase serait "*In which state are playing the* $?pl_1$". Dans ce genre de cas, il est important d'identifier au sein de la base de connaissances des éléments soumis aux mêmes propriétés, afin de correspondre avec la conjugaison d'origine de l'énoncé. Là encore, bien que ce critère permette de s'assurer d'une cohésion parfaite entre la phrase d'origine et la phrase générée, il n'est pas essentiel, au regard de la compréhension de la question posée, que ce critère soit validé.

4. **la correspondance des temps :** Dans certains cas, le temps utilisé lors de la formulation de la question d'origine est particulièrement important. C'est le cas notamment lorsque l'on fait référence à une personne décédée, ou à un endroit qui n'existe plus. Dans ce cas la phrase est généralement formulée au passé, et la présence de certains candidats, toujours en vie par exemple, pourraient être inappropriée. Par exemple dans le cas où la phrase d'origine est : "*Which person among the following was the leader of the first Empire ?*", on va mettre en place le modèle "*Which person among the following was the leader of the* $?pl_1$".

5.1.1 Choix des modèles

Lors de la génération de nos questions à choix multiples, nous utilisons les modèles de questions définis précédemment pour identifier des entités à l'aide du sous-graphe de relations, puis pour générer l'énoncé en langage naturel à l'aide d'un arbre syntaxique. Bien que, comme nous le verrons par la suite, le score de corrélation question/thème prenne en compte les entités qui sont insérées dans le modèle, il faut avant tout sélectionner des modèles pertinents pour le thème, de façon à maximiser le score de la question. En effet, la solution permettant de générer la question la plus pertinente pour un thème donné consisterait à tester l'ensemble des modèles avec l'ensemble des entités, et de choisir le couple maximisant le score dans le thème.

Cette solution est cependant coûteuse en temps de calcul, dans la mesure où la sélection des entités de la question à l'aide des modèles est une opération qui nécessite un parcours exhaustif des candidats. Par ailleurs, le score de corrélation entre modèle et thèmes, calculé lors de la section 4.3 nous permet de distinguer les modèles considérés comme pertinents vis-à-vis d'un thème, de ceux qui le sont moins. Nous avons donc utilisé le score ST de corrélation entre modèles et thèmes pour trier les modèles par ordre de pertinence pour le thème choisi, et ainsi sélectionner le top-k des modèles les plus pertinents.

5.1.2 Identification des candidats

Les modèles de questions sont composés des énoncés, des variables et du sous-graphe des relations R. Dans cette section, nous présentons la méthode mise en place pour utiliser ces sous-graphes de relations afin d'identifier dans la base de connaissances des candidats compatibles avec l'énoncé de la question. Le contenu de ces sous-graphes de relations nous permet d'identifier avec une seule requête l'ensemble des entités nécessaires à la génération de toutes les questions correspondant au modèle. Chaque "ligne" résultat de la requête contient une entité bonne réponse et une entité pour chaque variable. Chacune de ces lignes permet de constituer une *instance* de question, et sera désignée par le symbole I dans la suite du manuscrit.

Avec la multitude d'informations qui composent généralement les bases de connaissances, il est fréquent que plusieurs milliers, voire dans certains cas plusieurs dizaines de milliers d'instances I correspondent à un modèle donné. Il est donc nécessaire de filtrer ces résultats pour ne conserver que des instances viables vis-à-vis du thème de référence de la question. En effet, l'objectif étant de générer des questions relatives à un thème prédéfini, certains modèles *transversaux* peuvent être pertinents dans plusieurs thèmes, notamment lorsque le thème dépend principalement des entités présentes dans l'énoncé. Par exemple, c'est le cas d'une question concernant le lieu de naissance d'une personne. La thématique de la question y dépend entièrement de la personne choisie. Un tel modèle sera pertinent d'un point de vue historique si le candidat est *Napoleon*, mais plutôt d'un point de vue artistique si le candidat est *Leonardo da Vinci*.

Il est donc important de faire intervenir la notion de contexte à la fois lors de la sélection du modèle de la question, comme nous l'avons vu dans la section 4.3 mais également lors de la sélection d'une instance I. En ce qui concerne la sélection de cette instance, notre démarche se scinde en deux parties distinctes : (i) l'obtention des instances compatibles avec l'énoncé, (ii) la sélection parmi ces instances du meilleur candidat.

Obtention des instances : La démarche présentée ici consiste à identifier chaque instance I compatible en s'appuyant sur les sous-graphes de relations des modèles de questions. Nous générons ainsi automatiquement des requêtes SPARQL, dans le but d'extraire depuis la base de connaissances des ensembles d'entités conformes aux critères du modèle. Chacun de ces ensembles est formé par les entités candidates, et la bonne réponse, le tout formant une instance valide.

La première étape consiste ainsi à identifier au sein de la base de connaissances l'ensemble des informations de chaque entité nécessaires pour notre processus de génération de questions. En l'occurrence, quatre éléments sont importants :

— **L'identifiant Freebase,**

— **L'alias Freebase**,composé d'une chaîne de caractère correspondant à l'intitulé en anglais,

— **L'identifiant Wikipedia**,

— **Le titre de l'article Wikipedia**.

Nous avons ensuite généré automatiquement, à l'aide du sous-graphe, les requêtes SPARQL permettant d'identifier ces instances. Pour chaque entité, on s'assure d'obtenir la totalité des informations listées ci-dessus. Ces requêtes utilisent les identifiants uniques des variables du modèle ($?pl_1$, $?pl_2$,...) ainsi que $?answer$ pour former les variables de la requête SPARQL. Chacune des variables de la requête est ainsi subdivisée en quatre sous-variables composées par les quatre éléments présentés ci-dessus. La figure 5.2 donne un exemple de transformation de modèle de questions en requête SPARQL. Dans cet exemple, les trois variables sont $?pl_1$, $?pl_2$ et $?answer$. Le corps de la requête est quant à lui composé des éléments de notre sous-graphe de relations R. On note également au bas de la requête la restriction sur le type de la bonne réponse (ici, "*person.actor*"). Nous ajoutons systématiquement le type de la bonne réponse pour limiter le nombre de résultats renvoyés, tout en augmentant leur pertinence.

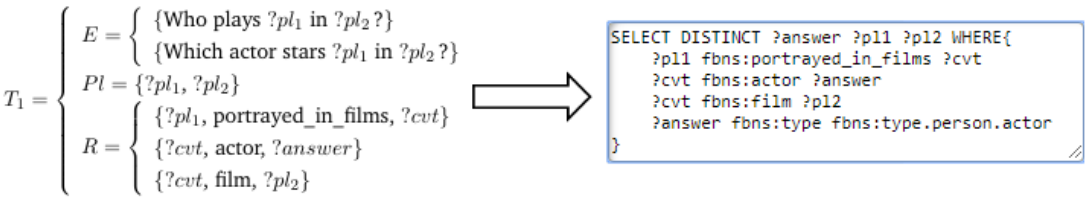


Fig. 5.2.: Exemple de génération de requête SPARQL à partir d'un modèle de questions

Ces requêtes générées automatiquement permettent de récupérer des instances, chancune contenant une bonne réponse et les candidats corespondants pour former une question. Comme précisé dans la définition des modèles de questions en section 4.1.2, les questions à choix multiples peuvent admettre plusieurs bonne réponses. C'est donc à cette étape que nous filtrons systématiquement les instances qui contiennent plusieurs bonnes réponses.

Filtrage des candidats : La multitude des données présentes dans Freebase permet de façon générale d'associer aux modèles un grand nombre d'instances différentes. Le second objectif de cette sélection d'instances consiste par conséquent à filtrer les candidats en ne conservant que ceux qui sont pertinents du point de vue du thème.

Nous avons ainsi considéré la question globalement, en cherchant à maximiser la corrélation de la totalité des entités de chaque instance, plutôt que de traiter séparément chaque élément. Afin d'introduire une mesure de corrélation viable entre une question à choix multiples et un thème, nous considérons deux aspects de cette mesure : tout d'abord le score du modèle de la question, indépendamment de toute entité, et ensuite le score des entités, prenant en compte tous les éléments de l'instance.

On définit dans un premier temps $ST(m, t)$, la fonction de score d'un modèle m de rang r au sein d'un thème t, en reprenant le score et le rang calculés en section 4.3.

Définition 11: Score d'un modèle pour le thème

$$ST(m,t) = \frac{st(m,t)}{st(1,t)} \times \frac{R-r}{R}$$

où

- $st(r,t)$ est le score du modèle pour le thème calculé en section 4.3
- $st(1,t)$ est le score du meilleur modèle pour le thème, soit le modèle de rang 1
- R, le nombre total de modèles ayant un score non nul pour le thème
- r, le rang du modèle pour le thème t
- t, le thème

On normalise ainsi entre 0 et 1 le score et le rang du modèle.

Pour un modèle donné, on utilise ensuite la fonction de score $S(I, t)$, qui donne le score d'une instance I vis-à-vis d'un thème t.

Définition 12: Score des entités

$$S(I,t) = \frac{\sum_{j\to 1}^{|I|} \left(\frac{sa(j,t)}{sa(1,t)} \times \frac{|t|-r_j}{|t|} \right)}{|I|}$$

où

— $sa(j,t)$ est le score de *Pagerank* de chacune des entités de la question au sein du thème t, calculé dans de la section 3.4.2

— $sa1,t$ est le score de l'entité la mieux classée pour le thème t

— I est une instance, composée d'une bonne réponse et de un ou plusieurs candidats. $|I|$ est donc le nombre d'entités au sein de cette instance.

— r_j le rang d'une entité de rang j au sein du thème t

— t est le thème considéré. $|t|$ représente donc le nombre total d'entités ayant un score de *Pagerank* non nul pour le thème considéré.

On normalise entre 0 et 1 le score et le rang de chaque entité. $S(I,t)$ est obtenu en calculant la moyenne de ces valeurs normalisées.

Ces formules nous permettent de calculer séparément le score d'un modèle m et de chaque instance I au sein d'un thème t. Pour calculer la valeur de corrélation d'une question avec un thème, on combine ces deux scores au sein d'une même mesure : l'*indice de confiance* $IC(r,i,t)$. Ce score est défini de la façon suivante :

Définition 13: Indice de confiance

$$IC(I,m,t) = p \times S(I,t) + (1-p) \times ST(m,t)$$

où

— p est le pondérateur permettant de donner une importance relative à chacune des deux valeurs.

— I est une instance, composée d'une bonne réponse et de un ou plusieurs candidats

— m désigne un modèle de questions

— t représente un thème

$IC(r,i,t)$ mesure ainsi le score de pertinence pour un thème t de la composition d'un modèle m et d'une instance I.

$IC(r,i,t)$ correspond à la moyenne pondérée des deux scores qui le composent, à savoir le score donné au modèle m, et le score donné à une instance I regroupant une bonne réponse et un ensemble de candidats. Cette mesure intervient après l'exécution

de la requête SPARQL, et permet de comparer pour un thème donné l'ensemble des scores donnés aux couples $<m,\ I>$. Ce score considère ainsi globalement la pertinence de chaque question pour un thème donné. L'utilisation d'une telle mesure nous permet de trier par ordre de pertinence les questions générées au sein d'un thème, afin de déterminer parmi ces questions lesquelles sont susceptibles d'être les plus intéressantes pour ce thème.

5.1.3 Substitution des variables par les candidats

Avant de procéder à la phase de génération des énoncés en langage naturel, il est nécessaire de substituer les variables par les candidats au sein de la phrase. Cette étape de substitution intervient après que les couples $<m,\ I>$ aient été comparés à l'aide de la fonction $IC(r, i, t)$ introduite à l'étape précédente. Ainsi, on a sélectionné un modèle et une instance pour générer la question. L'objectif ici est de préparer le contenu d'un arbre syntaxique du modèle issu de E en y remplaçant ses variables par les entités sélectionnées.

Tout d'abord, il est nécessaire de choisir un énoncé parmi l'ensemble d'énoncés E du modèle. Dans notre cas, nous considérons l'ensemble des énoncés du modèle comme équivalents (un score de difficulté pourrait leur être attribué à terme). Un tirage aléatoire équiprobable est donc réalisé pour sélectionner l'énoncé qui sera utilisé pour la génération de la question.

Une fois l'énoncé déterminé, l'ensemble de ses variables sont substituées par l'ensemble des candidats déterminés à l'étape précédente. La structure générique de nos modèles nous permet d'identifier rapidement, pour chaque candidat, son emplacement au sein de l'énoncé.

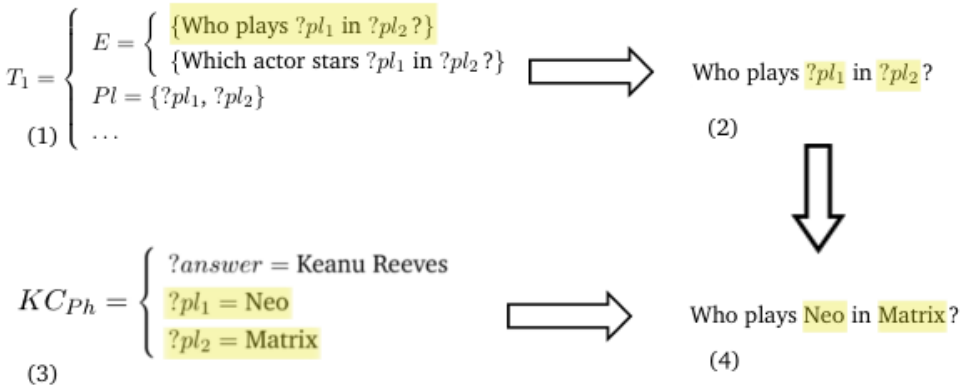


Fig. 5.3.: Substitution des variables au sein d'un énoncé du modèle

La figure 5.3 illustre ce procédé de substitution des variables à l'aide d'un exemple. (1) On y retrouve le modèle T_1 composé d'un ensemble d'énoncés, et notre ensemble

d'instances candidates. (2) Un énoncé est aléatoirement choisi au sein de E ; ici il s'agit de l'énoncé "*Who plays* $?pl_1$ *in* $?pl_2$ *?*". (3) Lors de la recherche de candidats, on connait grâce à notre modèle de questions la correspondance entre variables et candidats. Ces informations sont donc également explicites au sein de I. (4) Les variables de l'énoncé sélectionné sont finalement substituées par les éléments correspondants issus de I.

5.1.4 Génération à partir des arbres syntaxiques

A ce stade, les variables du modèle ont été substituées par les candidats issus de I. La dernière étape de la génération de l'énoncé en langage naturel consiste donc à transformer son arbre syntaxique en phrase. L'utilisation des arbres syntaxiques permet une modularité intéressante des phrases. Il est ainsi possible, grâce aux méta-données qui décrivent chaque mot, de modifier certains aspects de la phrase, comme son temps, ou le genre d'un possessif, de façon à s'assurer que le contenu de l'énoncé est cohérent avec les entités sélectionnées. Nous avons mis en place une solution permettant de générer des phrases à partir des arbres syntaxiques.

L'objectif de notre générateur en langage naturel est donc de de convertir un arbre syntaxique en une phrase intelligible pour un être humain. Initialement, les arbres syntaxiques sont utilisés en Traitement Automatique du Langage pour analyser automatiquement le contenu d'une phrase, et identifier précisément ses composants. Combiné à d'autres outils de ce domaine, les arbres syntaxiques permettent à une machine de *comprendre* sémantiquement le contenu d'une phrase.

Dans notre cas, l'enjeu est inversé, dans la mesure où il s'agit de générer une phrase en langage naturel à partir d'un arbre syntaxique donné. Nous avons ainsi mis en place un générateur sur la base d'une étude des positionnements conditionnels des tags des arbres syntaxiques les uns par rapport aux autres. En effet, l'analyse de ces tags issus de phrases diverses, couplée à la documentation technique fournie par Stanford (De Marneffe et Manning, 2008) et le Penn Treebank (Marcus et al., 1994), nous a permis de mettre en place un ensemble de règles conditionnelles de positionnement des mots en fonction de leurs POS tags.

On parle ici de règles conditionnelles, car ces POS tags étant initialement destinés à l'extraction de données plutôt qu'à leur restitution, il ne suivent pas forcément de logique implacable à ce sujet. Ainsi, on distingue plusieurs catégories de placement conditionnel :

— **Place fixe :** Le mot correspondant se situe toujours avant, ou après le mot précédent. Par exemple, si la dépendance est *DET*, c'est-à-dire une relation entre un déterminant et un sous-arbre, ce dernier se placera toujours avant.

— **Sur place :** Le mot correspondant se situe à l'emplacement où il est rencontré dans la phrase.

— **Placement conditionnel de tags :** Le mot correspondant est placé en fonction des tags des mots qui le suivent ou le précèdent. Par exemple, un adjectif attribut, désigné par la dépendance *ATTR*, se situera toujours avant le mot qu'il modifie, sauf s'il est notamment associé avec un autre adjectif qualificatif (*NSUBJ*) : "*The most beautiful cat*".

— **Placement conditionnel sur type :** En plus de son tag, le type du mot (verbe, nom, etc.) est déterminant pour son placement. Par exemple un adjectif, désigné par la dépendance *NSUBJ*, se place généralement avant le mot suivant, sauf par exemple si le type du mot précédent est un *WH-pronoun* (i.e. "*How fine...* ").

— **Placement conditionnel complexe :** Certains tags génériques sont sur-représentés. Il a fallu mettre en place un ensemble de règles complexes prenant en compte les éléments précédents pour obtenir un résultat viable.

Au total, on obtient un ensemble de 85 règles permettant de déterminer la position des mots dans la phrase générée. Ces règles s'organisent sous la forme d'une machine à états finis conditionnelle, qui, en fonction des dépendances, va statuer sur le positionnement des mots à placer. La figure 5.4 donne un aperçu du raisonnement effectué par la machine à états finis pour statuer sur le positionnement. On y voit ainsi qu'à partir du mot considéré, un ensemble de critères sont étudiés pour déterminer sa position relative par rapport au mot suivant de l'arbre. Les variables *after* et *before* nous permettent d'étendre l'analyse conditionnelle aux dépendances des mots suivants et précédents dans l'arbre. La variable *tag* nous permet de prendre en compte la nature du mot considéré en plus de son lien avec les autres mots du graphe.

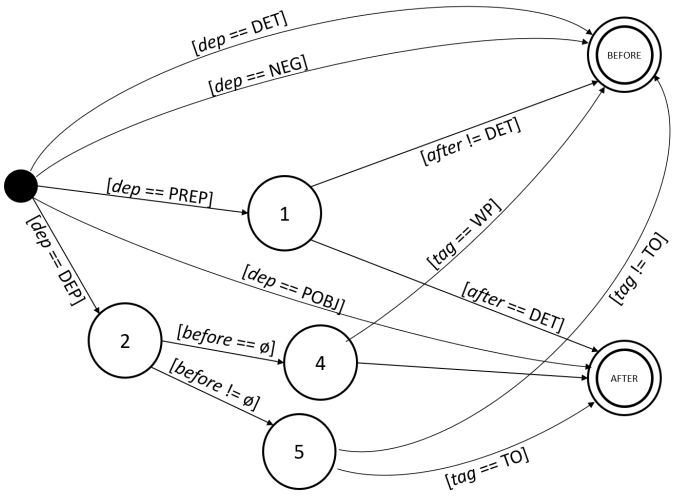


Fig. 5.4.: Machine à états finis partielle représentant la décision conditionnelle de positionnement des dépendances en fonction des tags de l'arbre syntaxique

La section 6.4 présente les résultats obtenus lors de l'évaluation de cette approche de transformation des arbres syntaxiques en langage naturel.

5.2 Génération des distracteurs

Une fois que l'énoncé de la question est généré en langage naturel, la dernière étape de notre générateur de questions à choix multiples consiste à trouver un ensemble de distracteurs à notre question. En se référant à la littérature (SEYLER et al., 2017)(GRAESSER et WISHER, 2001), on considère qu'un distracteur doit remplir deux critères principaux pour que la personne interrogée les considère comme une alternative viable à la bonne réponse. Premièrement, *le type* de l'élément proposé doit être le même que celui de la bonne réponse, et deuxièmement, le *contexte* doit se rapprocher autant que possible de la réponse initiale.

Nous illustrons l'importance de ces deux éléments à travers un exemple. Considérons la question "*Who is the author of the song Imagine ?*", et sa bonne réponse "*John Lennon*". Si on propose avec cet énoncé le distracteur "*Guitar*", qui respecte la contrainte du contexte, mais pas celle du type, la personne interrogée comprendra immédiatement qu'il ne s'agit pas de la bonne réponse. Inversement, si on propose le distracteur "*Nelson Mandela*" ne validant que la contrainte du type mais pas celle du contexte, le distracteur sera également écarté facilement. Ces deux éléments sont donc complémentaires pour obtenir des distracteurs de qualité. Une proposition pertinente pour cette question serait "*Paul McCartney*", qui valide à la fois le contexte et le type de la bonne réponse.

La difficulté liée à l'identification des distracteurs au sein des bases de connaissances vient principalement du fait que ces dernières ne contiennent pas d'informations contextuelles. Ainsi, les critères du type et du contexte des distracteurs doivent être étudiés séparément. Nous présentons ainsi dans une première partie la méthode utilisée pour résoudre le problème du *type*, et dans une seconde partie, celle permettant de résoudre le problème du *contexte*.

5.2.1 Type des distracteurs

Cette première étape de génération de distracteurs consiste à identifier au sein de la base de connaissances un ensemble de candidats de distracteurs qui remplissent les critères de type de la bonne réponse. On utilisera dans cette section les termes de *candidats de distracteurs* pour désigner les éléments qui, au fur et à mesure du processus de génération de distracteurs, forment un ensemble de candidats au sein desquels seront finalement sélectionnés les distracteurs.

Dans Freebase, les types offrent l'avantage d'être organisés hiérarchiquement en fonction de leur niveau de précision. Par exemple, "*person.actor*" est un sous-type de "*person*" dans la mesure où un acteur est forcément une personne, alors que la réciproque ne s'applique pas. On considère donc que "*person.actor*" est plus précis que *person*. Ainsi, nous avons émis l'hypothèse que plus un distracteur a de points communs avec la bonne réponse, plus il sera crédible en tant que tel. Nous n'avons pas limité la recherche au seul type de la bonne réponse, mais aussi inclus la sémantique inhérente au sous-graphe de relations R de notre modèle. Illustrons cette hypothèse par un exemple. Si la question est "*Which actor has the main role in the movie* $?pl_1$", limiter la sélection des distracteurs au type *person.actor* ne sera pas suffisant pour les rendre crédibles. Dans cet exemple, il faudrait de plus que le distracteur soit un acteur ayant au moins un rôle principal dans sa carrière.

Nous avons ainsi repris et généralisé les sous-graphes de relations R de nos modèle de questions, dans l'optique de construire automatiquement des requêtes SPARQL permettant d'identifier des candidats respectant l'ensemble des contraintes de R . La totalité des éléments présents dans le sous-graphe de relations du modèle de

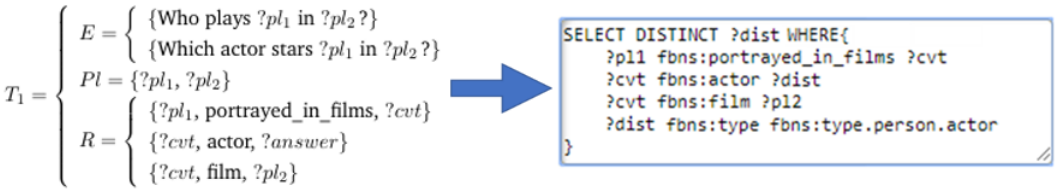


Fig. 5.5.: Exemple de génération d'une requête SPARQL destinée à identifier des distracteurs à partir d'un modèle de questions

questions sont utilisés pour construire la requête SPARQL permettant rechercher des éléments similaires à la bonne réponse. Cependant, vis-à-vis du sous-graphe de relations du modèle, seule l'entité $?answer$ qui indique la réponse de la question nous intéresse. C'est en effet à partir de cette variable que sont identifiés en premier lieu les candidats de distracteurs. Dans la figure 5.5, nous illustrons la génération de cette requête SPARQL à l'aide d'un exemple. La variable $?answer$ a été substituée dans le corps de la requête par la variable $?dist$, chargée de chercher des candidats au sein de la base de connaissances. Les autres éléments variables ($?pl_1$, $?pl_2$, $?cvt$) ne sont pas explicitement interrogés dans le sens où les entités correspondantes ne seront pas affichées, mais ces variables restent nécessaires pour valider le bon fonctionnement de la requête.

Dans certains cas, les contraintes liées au modèle de questions du modèles de questions ne permettent pas d'identifier de candidats de distracteurs (eg. la requête SPARQL ne renvoie pas, ou pas assez de résultats). Nous avons donc mis en place des stratégies alternatives pour trouver malgré tout des candidats potentiels. Nous

avons choisi de relâcher progressivement les contraintes imposées par le modèle pour agrandir progressivement l'ensemble des candidats potentiels. Dans le cas extrême, seul le type de la bonne réponse est utilisé. La figure 5.6 montre les étapes

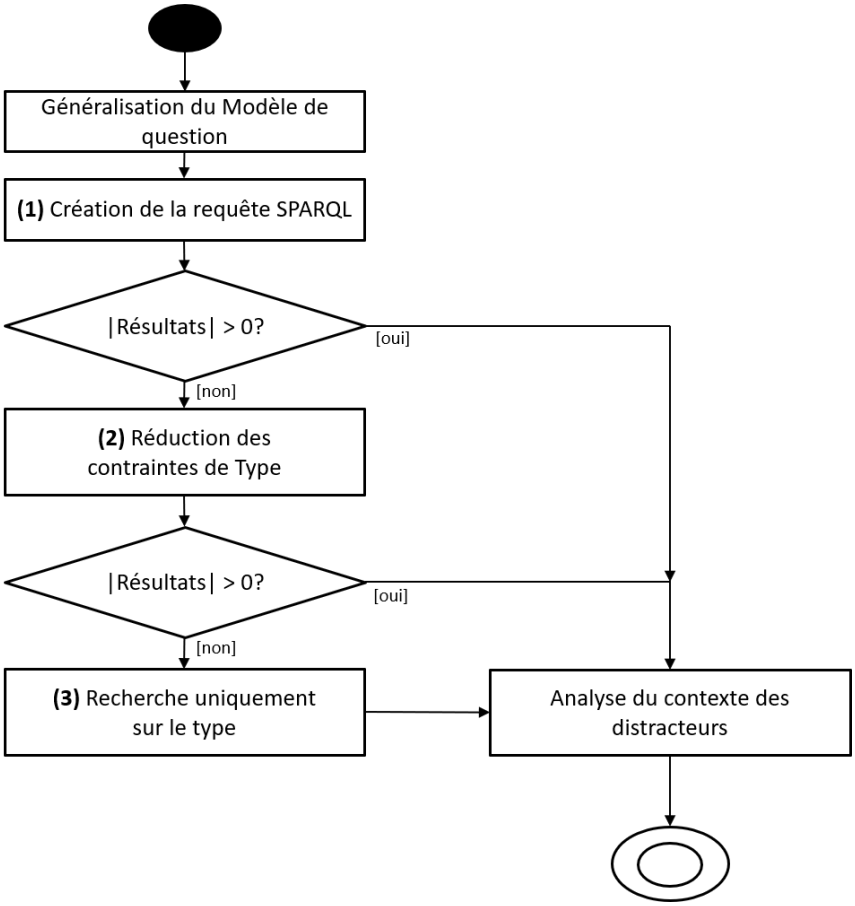


Fig. 5.6.: Diagramme d'activité explicitant les étapes de réduction des contraintes lors de la recherche des candidats de distracteurs

utilisées pour construire et modifier la requête générique chargée de trouver des candidats de distracteurs à une question. Tant qu'aucun candidat n'est renvoyé, on élargit le champs de recherche, jusqu'à se limiter au type seul. Pour illustrer cette figure, reprenons l'exemple "*Which actor has the lead role in the movie* $?pl_1$ *?*". Dans cet exemple, on demande à l'utilisateur quel acteur a le rôle principal pour un film donné. Pour minimiser le nombre de distracteurs potentiels tout en maximisant leur crédibilité, nous générons tout d'abord la requête *(1)* la plus contraignante, à savoir celle qui recherche les candidats qui sont à la fois des personnes ayant eu un rôle principal dans un film, et qui sont également des acteurs. Si cette requête ne donne pas de résultats, ou pas en quantité suffisante, nous générons une seconde requête *(2)* moins contraignante qui recherche des personnes ayant eu un rôle dans un film, et qui sont des acteurs. De même que précédemment, si la quantité de résultats est insuffisante, une nouvelle requête est générée *(2)* afin de réduire la recherche uniquement aux résultats de type acteurs. Dans l'hypothèse faible où le nombre de résultats serait insuffisant (pour des types potentiellement très spécifiques),

nous élargissons dans une dernière requête *(3)* la recherche aux candidats du type générique, dans notre exemple, toutes les personnes. Il est important de préciser que dans une base de connaissances comme Freebase, les requêtes *(2)* et *(3)* ne sont presque jamais utilisées. En revanche, la solution étant générique, nous avons pris en compte son éventuelle adaptation sur des bases de connaissances plus modestes.

5.2.2 Contexte des distracteurs

Nous avons à l'étape précédente identifié dans notre base de connaissances tous les candidats de distracteurs validant le sous-graphe de relations R d'un modèle de questions donné. Il s'agit maintenant de filtrer cet ensemble en fonctions des données contextuelles à notre disposition. Cette seconde partie de la génération de distracteurs consiste à ne retenir parmi les candidats de distracteurs que ceux qui sont pertinents dans le contexte de la question. Comme mentionné précédemment, les bases de connaissances que nous avons utilisées ne permettent pas de définir à elles seules des informations relatives au contexte, en raison de l'absence de méta-données spécialisées. Cette section détaille donc les méthodes utilisées pour identifier les distracteurs les plus pertinents dans le contexte de la question.

Dans l'étape précédente, nous avons vu que nous appliquons en priorité un maximum de contraintes à la requête. Ces contraintes ne sont progressivement levées que dans le cas où il n'y a pas, ou pas assez de candidats de distracteurs. Ces contraintes sont volontairement aussi précises que possible, car elles ont l'avantage de réduire la recherche aux candidats les plus pertinents tout en limitant ce nombre de candidats. Malgré ce filtre, et compte tenu du nombre important d'entités présentes dans les bases de connaissances, il arrive fréquemment que le nombre de candidats de distracteurs soit extrêmement élevé. Nous avons donc mis en place une solution destinée à extraire le top-k des éléments les plus proches du contexte choisi, suffisamment efficace pour prendre en compte le nombre potentiellement important de candidats.

Pour déterminer les top-k distracteurs, nous avons ainsi défini une combinaison de deux approches, une basée sur le *Pagerank*, et l'autre sur la similarité sémantique.

Méthode basée sur le *Pagerank* : En reprenant notre mesure de *Pagerank* définie dans le chapitre 3, nous avons pu affecter un score de pertinence à chaque candidat de distracteurs dans le thème de la question. Ce score nous permet de classer les candidats identifiés selon leur corrélation avec le thème. On considère que plus le *Pagerank* d'un candidat est élevé, plus il est lié au contexte de la question, et donc pertinent en tant que distracteur.

Méthode basée sur la similarité sémantique : Mesurer la corrélation des candidats de distracteurs avec la thématique de la question n'est pas suffisant pour identifier des distracteurs pertinents. En effet, des questions similaires auraient systématiquement des distracteurs similaires. Nous avons donc complété la méthode décrite ci-dessus avec une mesure se basant sur la proximité des distracteurs avec la bonne réponse. En mesurant la similarité sémantique entre chaque candidat et la bonne réponse à l'aide de LSA, nous identifions parmi les candidats retenus ceux qui possèdent une corrélation maximale avec la bonne réponse.

En combinant et pondérant les approches précédentes, on calcule ainsi $sd(d,t)$, le score d'un distracteur d pour la question, en prenant en compte le thème t prédéfini :

$$sd(d,t) = p \times sa(d,t) + (1-p) \times LSA(answer,d)$$

où

— $sa(d,t)$ est le score de *Pagerank* de l'entité d pour le thème t, calculé à la section 3.4.2,

— $lsa(answer,d)$ est la similarité sémantique entre d et la bonne réponse de la question,

— p est un pondérateur entre 0 et 1,

— t est le thème prédéfini qui conditionne le processus de génération de la question.

On obtient ainsi une mesure capable de trier les distracteurs par importance, tout en prenant à la fois en compte le contexte de la question, et la proximité du distracteur avec la bonne réponse.

5.2.3 Diversité des distracteurs

Afin d'augmenter la diversité des distracteurs générés, nous avons introduit un tirage aléatoire parmi le top-k des candidats de distracteurs obtenus à l'aide de notre fonction $sd(d,t)$, calculée précédemment. Nous appliquons ainsi la règle suivante : Soit n le nombre des distracteurs nécessaires pour une question donnée (dans notre cas, $n = 3$), on recherche les top-k entités, où :

$$k = 2n + 3$$

Un tirage aléatoire sans remise est ensuite réalisé parmi l'ensemble obtenu. Ces valeurs arbitraires ont été établies suite à l'analyse expérimentale du fait que la pertinence des distracteurs décroit rapidement quand k augmente. Nous avons donc testé plusieurs valeurs de k et de n, et validé que le meilleur compromis était trouvé quand $k = 2n + 3$. Dans l'hypothèse où l'ensemble de candidats de distracteurs a une taille inférieure à k, le top-n est systématiquement sélectionné.

5.2.4 Choix des distracteurs

Ainsi, pour synthétiser l'approche de sélection de distracteurs présentée dans ce document, on peut résumer cette étape de choix des distracteurs à un ensemble de filtres opérant successivement à partir des entités issues de la base de connaissances. Sont ainsi filtrées en premier lieu les entités de types similaires, puis celles qui sont pertinentes dans le contexte de la question, et pour finir, celles qui sont sémantiquement proches de la bonne réponse.

En effet, les bases de connaissances ne contiennent pas d'informations permettant de restreindre la recherche au contexte de la question. Il est donc nécessaire d'extraire depuis la base de connaissances *toutes* les entités ayant un type similaire à la bonne réponse. C'est pourquoi il est important de choisir le type le plus spécifique possible, permettant ainsi de limiter le nombre de candidats.

Dans un second temps, il faut choisir parmi cet ensemble de candidats quels sont les éléments les plus crédibles pour la question. En se basant sur les données d'une base de connaissances de la taille de Freebase, il y a à ce stade parfois plusieurs milliers, voire plusieurs dizaines de milliers de candidats. Nous effectuons donc un premier filtre contextuel en se basant sur le score de *Pagerank* défini précédemment. Ce score définit en effet l'importance de chaque élément au sein d'un thème. C'est notre mesure de contexte la plus fiable.

Cependant, ce score seul ne suffit pas pour trouver des distracteurs pertinents. En effet, en se limitant à cette mesure, des questions possédant des énoncés différents mais un type de réponse et un thème similaires auraient toutes les mêmes distracteurs. Ainsi, afin de renforcer la crédibilité des distracteurs, nous avons couplé cette mesure de *Pagerank* avec une mesure de similarité sémantique avec la bonne réponse. Cette mesure de similarité sémantique vient compléter le *Pagerank*, en identifiant des potentiels distracteurs qui sont à la fois dans le contexte de la bonne réponse, mais aussi sémantiquement proches d'elle.

Ainsi, les différentes étapes qui filtrent progressivement l'ensemble des candidats, pour ne retenir au final que les distracteurs, peuvent se résumer comme suit :

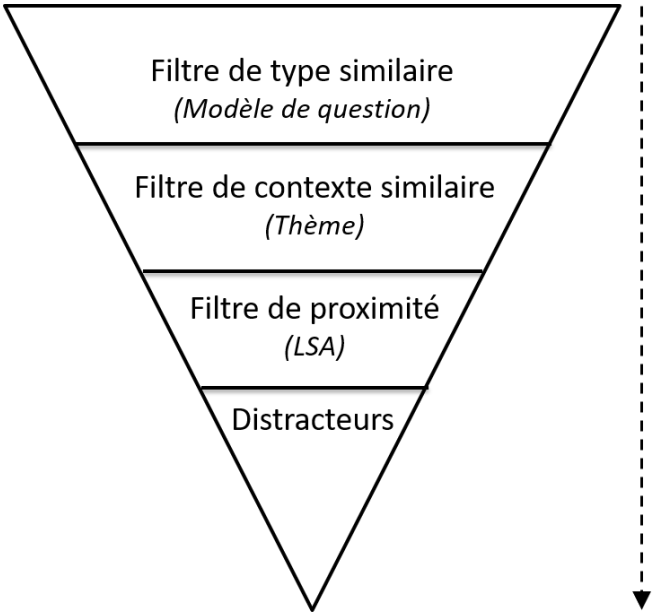


Fig. 5.7.: Filtre progressif des candidats de distracteurs

1. **Filtrage des entités de type similaire :** Au sein de la base de connaissances, les entités ayant un type similaire à la bonne réponse sont conservées, les autres sont filtrées.
2. **Filtrage des entités sur le contexte du thème :** En se servant de la mesure de *Pagerank*, on filtre les entités n'ayant pas de lien avec le thème (*Pagerank* nul).
3. **Filtrage des entités sur le contexte de la bonne réponse :** En mesurant la similarité sémantique (LSA) entre les entités restantes et la bonne réponse de la question, on ne retient que les entités qui ont un lien fort avec cette dernière.
4. **Sélection aléatoire dans les candidats restants :** Pour garantir une certaine diversité au sein des distracteurs proposés, on ne se contente pas de choisir les n premiers éléments, où n correspond au nombre de distracteurs souhaité, mais on va sélectionner n éléments parmi l'ensemble des candidats potentiels.

Ces différentes étapes de filtrage sont illustrées par la figure 5.7.

5.3 Des questions multi-sujets

Une fois générée, une question à choix multiples se présente sous la forme d'un énoncé, d'une bonne réponse, et d'un ensemble de distracteurs. Bien que cette question ait été initialement générée pour un thème donné, nous avons constaté que cette question pouvait aussi être jugée pertinente pour d'autres thèmes.

En se basant sur ce constat, nous avons ajouté une procédure supplémentaire permettant de déterminer, une fois la question générée, tous les thèmes pour lesquels elle est pertinente. Pour illustrer cette procédure, prenons l'exemple de la question suivante, générée dans le thème *History* : "*Who was the predecessor of John F. Kennedy*" et sa bonne réponse : *Dwight D. Eisenhower*. Bien que cette question reste tout à fait pertiente pour l'histoire, en mesurant sa pertinence pour les différents thèmes disponibles, on pourra constater que cette question est plus pertinente pour la politique. Elle est ainsi mise temporairement de côté, et sera utilisée la prochaine fois que le générateur générera une question politique.

Il est important de noter que ce genre de réaffectation n'est utile que lorsque que l'on souhaite utiliser ce générateur de questions pour générer massivement des questions de différents thèmes. Lorsque les paramètres de génération sont limités à 1 thème unique, cette stratégie de mise en cache n'est pas utilisée.

Pour appliquer cette procédure, nous comparons la similarité sémantique de l'énoncé et de la bonne réponse de chaque question avec nos différents thèmes en utilisant une méthodologie similaire à celle présentée en section 4.3 par la stratégie 1. Bien que cette stratégie ne se soit pas révélée très utile pour calculer l'appartenance d'un modèle à un thème, son utilisation se justifie beaucoup plus facilement pour une question générée. En effet, les entités de la question jouent un rôle déterminant dans son association avec un thème et ces dernières n'étaient pas présentes dans le modèle de questions (composé uniquement de variables), mais sont présentes dans la question.

Si la différence de score obtenue par la mesure de similarité ne permet pas de déterminer significativement un meilleur thème, nous calculons une seconde mesure de similarité basée uniquement sur Wikidata en utilisant une approche similaire à (Benedetti et al., 2016). Pour cette seconde mesure, nous n'utilisons pas la totalité des thèmes disponibles, mais seulement ceux ayant eu les meilleurs scores à l'étape précédente. Cette seconde mesure consiste à calculer la distance entre les noeuds correspondant aux entités de la question et les noeuds correspondant aux différents thèmes à évaluer au sein de Wikidata. Si à l'issue de cette seconde opération, nous n'arrivons toujours pas à déterminer un thème significativement meilleur, le thème

dans lequel la question à été initialement générée est considéré comme étant le plus pertinent.

5.4 Conclusion

Dans ce chapitre, nous avons présenté les travaux réalisés pour mettre en place un générateur de questions à choix multiples fonctionnel. Ce dernier repose sur les travaux présentés dans les chapitres précédants de cette thèse, à savoir la notion de thèmes au sein de bases de connaissances, et les modèles de questions.

Ainsi, en corrélant ces modèles de questions avec les thèmes, nous avons été en mesure de générer des questions à choix multiples à partir d'une base de connaissances, en prenant en compte un thème prédéterminé. Cette contribution est originale par rapport aux approches existantes qui cernent difficilement cette notion au sein des bases de connaissances.

Nous avons présenté en section 2.3 les différents travaux existants faisant mention de la notion de difficulté des questions générées. Cette notion de difficulté est relativement centrale dans le processus de génération de questions à choix multiples, pour permettre d'adapter les questions générées en fonction des personnes interrogées. Nous n'avons cependant pas, à l'heure actuelle, de corrélation entre les questions produites et leur difficulté. Plusieurs solutions sont cependant envisagées pour permettre d'intégrer à terme cette notion dans notre processus de génération de questions :

— Associer des difficultés différentes aux énoncés d'une questions.

— Utiliser par exemple la fonction de score des entités dans le contexte $sa(i, t)$ pour chercher des entités plus ou moins connues suivant la difficulté souhaitée.

— Augmenter/réduire la pertinence des distracteurs proposés proportionnellement à la difficulté recherchée.

La qualité des résultats obtenus pour générer des questions à choix multiples sont évalués dans le chapitre 6.

6 Expérimentation

Dans ce manuscrit, nous présentons les travaux réalisés pour mettre en place notre générateur de questions à choix multiples thématiques à partir de bases de connaissances. L'objectif de ce générateur est de construire des questions à choix multiples de qualité pour un thème donné. Nous avons donc testé son efficacité en générant un ensemble de questions à choix multiples, et en procédant à une évaluation d'une partie de ces questions. Cette section détaille les conditions dans lesquelles a été réalisée cette évaluation, et présente les résultats obtenus.

6.1 Conditions d'évaluation

Nous avons généré automatiquement 1200 questions à choix multiples dans des thèmes prédéterminés à l'aide de notre générateur de questions. Les thèmes identifiés dans le chapitre 3 étant nombreux, et pour certains, relativement spécifiques, nous avons choisi de simplifier l'évaluation en ne proposant aux personnes évalués qu'un ensemble restreint. L'objectif ici étant de trouver un compromis acceptable pour réduire le temps passé par les évaluateurs sur chaque question, et donc permettre d'évaluer plus de questions, tout en effectuant une évaluation pertinente vis à vis de la problématique.

Nous avons ainsi restreint la liste complète des 52 thèmes extraits automatiquement, à une liste partielle de 7 thèmes, issus de cette dernière, sous lesquels se situent les 45 autres thèmes de la liste. Ce regroupement a pu être réalisé automatiquement en se basant sur le système de catégories de Wikipedia. Nous identifions grâce aux liens hiérarchiques les catégories de Wikipedia les plus élevés dans l'arbre, qui couvrent l'ensemble des autres catégories présentes dans la liste de thèmes. Ces *super-thèmes* sont utilisés par les évaluateurs. Ce procédé est illustré par la figure 6.1.

La restriction de l'évaluation à ce sous-ensemble de thèmes offre deux avantages majeurs pour cette évaluation : premièrement, cela simplifie le travail des évaluateurs, qui ne sont pas perdus au sein d'une liste déroulante de 52 thèmes. Deuxièmement, cela limite l'intersection entre les thématiques, limitant ainsi la confusion des évaluateurs lors du choix du thème. Par exemple, une question comme "*Dans quel pays se trouve Stonehenge ?*" peut être classée à la fois dans les thèmes de l'Angleterre,

de l'Union Européenne, ou de la Géographie. En dehors de contexte supplémentaire, il est difficile pour l'évaluateur de décider laquelle de ces trois propositions est la meilleure. En regroupant ces thèmes sous le thème *Géographie*, la confusion disparait.

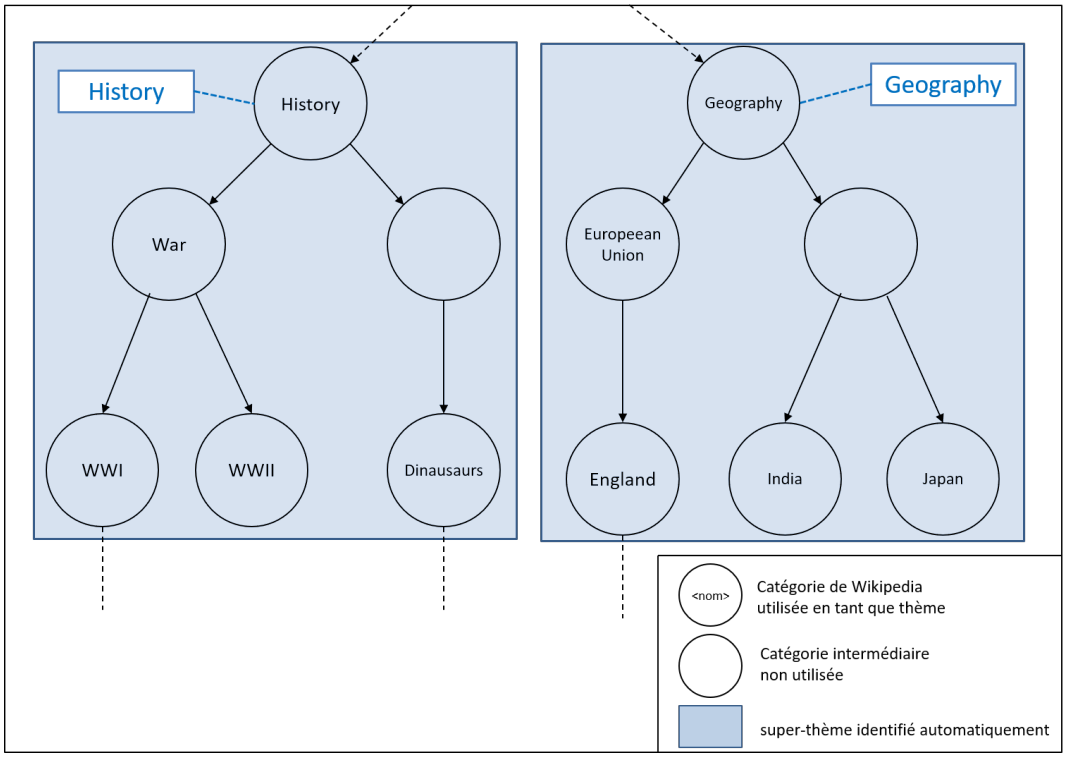


Fig. 6.1.: Extraction automatique des thèmes à partir des thèmes en se basant sur la hiérarchie des catégories de Wikipedia

La liste exhaustive des thèmes, ainsi que les super-thèmes identifiés pour les regroupements thématiques sont présentés dans l'annexe B. Ces super-thèmes peuvent ainsi être énumérés de la façon suivante :

— **Arts :** regroupe au sens large tous les éléments qui sont liés au domaine artistique, incluant l'architecture, la photographie, le cinéma, etc.

— **Géographie :** regroupe plus particulièrement tout ce qui est lié à une notion de localisation, comme l'emplacement des monuments, des pays, etc. Tous les aspects plus géopolitiques sont traités par le thème **société**.

— **Religion :** regroupe les éléments liés au domaine religieux, toutes religions confondues. On considère également l'athéisme et les sectes comme relevant de ce thème.

— **Histoire :** regroupe tous les éléments liés à l'histoire de la Terre, dont les éléments relatifs à l'histoire humaine, mais également les éléments concernant les dinosaures ou les modes de vie préhistoriques.

— **Sciences :** regroupe tous les éléments liés de près ou de loin au domaine scientifique. Il regroupe ainsi les grands domaines comme les mathématiques, la médecine, les sciences naturelles, etc.

— **Société :** regroupe tous les éléments qui sont liés à la vie en communauté. Il inclut ainsi la politique, l'éducation ou la justice. Ce thème inclut également les notions de communautés virtuelles, disponibles à travers Internet par exemple.

— **Sports :** inclut tous les éléments liés au domaine sportif. Il regroupe ainsi toutes les disciplines de ce domaine, ainsi que les grands événements qui y sont associés (ex : Jeux olympiques).

L'évaluation a été réalisée à l'aide d'une application Web, permettant aux évaluateurs de répondre efficacement aux différents critères sélectionnés. Cette application Web offre l'avantage de permettre aux évaluateur de procéder à une évaluation discontinue des questions. En effet, étant authentifiés à l'aide d'une adresse e-mail unique, les utilisateurs peuvent à tout moment stopper et reprendre leur évaluation là où ils se sont arrêtés. Par ailleurs cette application Web permet de garantir que les évaluateurs ont bien évalué les mêmes questions, permettant de comparer l'efficacité du système de façon plus objective.

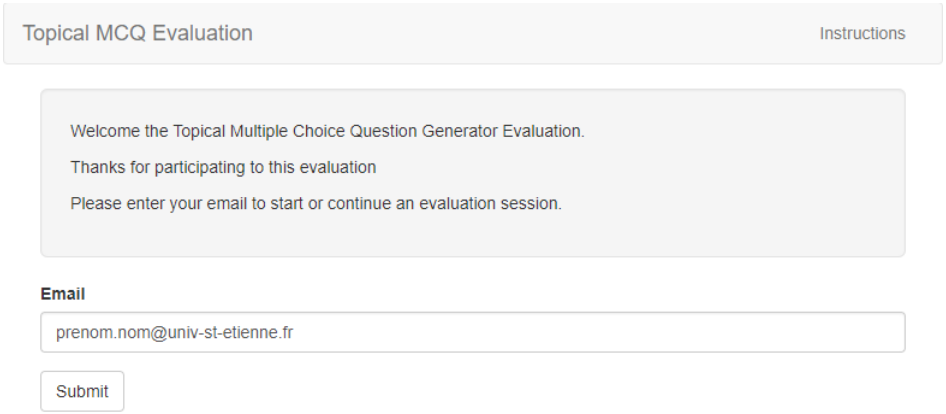


Fig. 6.2.: Interface d'authentification pour l'application Web d'évaluation des questions.

La page d'authentification des utilisateurs est présentée sur la figure 6.2. Il est important de noter que pour cette évaluation, seule l'adresse e-mail était demandée pour accéder à l'évaluation. Si cette dernière est déjà connue de l'application, l'évaluation est reprise à la dernière question évaluée ; dans le cas contraire, une nouvelle session d'évaluation est créée. En effet, dans le cadre de cette évaluation, nous avons sollicité un nombre restreint de personnes connues à l'avance et volontaires pour évaluer les questions générées. Il n'était ainsi pas nécessaire d'obtenir plus d'informations sur les évaluateurs pour établir des statistiques sur l'âge ou la profession de ces derniers. Nous avons ainsi demandé à 3 évaluateurs d'évaluer nos données. Nous avons ciblé pour cela des profils compétents dans l'enseignement, de façon à pouvoir juger objectivement de la qualité des questions générées en se basant sur leur expérience

professionnelle. Au total, chaque évaluateur a évalué 300 questions individuellement. Ces questions ont été sélectionnées aléatoirement parmi les 1200 questions générées, mais restent les mêmes pour tous les évaluateurs.

6.2 Critères d'évaluation

Afin de refléter de la façon la plus complète possible la qualité des questions générées, nous avons évalué chaque composant des questions séparément. Ainsi, en nous basant sur les évaluations de questions à choix multiples existantes (Bailey et al., 1998)(Al-Yahya, 2014), nous avons défini 7 critères, listés dans la table 6.1, permettant aux évaluateurs de donner une estimation de la qualité pour chacun des éléments suivants : l'énoncé, la bonne réponse, les distracteurs, et la question dans son ensemble.

Critère	Description
Énoncé	
Q1	L'énoncé est cohérent et grammaticalement correct
Q2	L'énoncé est en relation avec le thème présenté
Bonne réponse	
K1	La réponse est effectivement une bonne réponse
Mauvaises réponses (Distracteurs)	
D1	La réponse est effectivement une mauvaise réponse
D2	La réponse est plausible en tant que bonne réponse
Evaluation globale	
G1	Estimation de la difficulté globale de la question
G2	Estimation de la qualité globale de la question

Tab. 6.1.: Critères d'évaluation d'une question à choix multiples. Énoncé, bonne réponse et distracteurs y sont évalués séparément.

Les critères d'évaluation des questions sont mesurés de la façon suivante :

— **Q1-L'énoncé est cohérent et grammaticalement correct :** l'évaluateur juge la validité grammaticale de l'énoncé sur une échelle de 1 à 5, où *1=la question est illisible ou incompréhensible* et *5=la forme et l'objectif de la question sont parfaitement corrects*.

— **Q2-L'énoncé est en relation avec le thème présenté :** l'évaluateur juge la pertinence de la question pour le thème présenté sur une échelle de 1 à 5, où *1=la question n'a rien à voir avec le thème* et *5=la question est parfaitement adaptée au thème*. Nous avons choisi ici de mettre une note plutôt qu'une variable binaire (appartient ou n'appartient pas) dans la mesure où certaines entités peuvent être associées à plusieurs thèmes. Toutefois, il est demandé à

l'utilisateur, si ce dernier n'est pas satisfait du thème proposé, d'en sélectionner un autre plus adéquat dans une liste déroulante.

— **K1-La réponse est effectivement une bonne réponse :** présenté sous la forme d'une variable binaire, ce critère permet de valider que la réponse proposée est effectivement une bonne réponse à la question posée.

— **D1-La réponse est effectivement une mauvaise réponse :** présenté sous la forme d'une variable binaire, l'évaluateur doit confirmer que le distracteur proposé est effectivement une mauvaise réponse. Chaque distracteur est évalué individuellement à l'aide de ce critère.

— **D2-La réponse est plausible en tant que bonne réponse :** l'évaluateur juge si le distracteur présenté est crédible ou non en tant que réponse potentielle à la question. Chaque distracteur est évalué individuellement à l'aide de ce critère.

— **G1-Estimation de la difficulté globale de la question :** En fonction de l'énoncé de la question et des distracteurs qui lui sont présentés, l'évaluateur donne une note globale qui représente le niveau de difficulté qu'il a éprouvé en répondant à la question. Cette note se situe sur une échelle de 1 à 5, où *1=Trop facile/Tout ou partie de la réponse est contenu dans l'énoncé*, et *5=Trop difficile : il faut être un expert du domaine pour répondre à la question*.

— **G2-Estimation de la qualité globale de la question :** En se basant sur son ressenti personnel en fonction des éléments qui lui sont présentés, l'évaluateur juge globalement la question sur une échelle de 1 à 5, où *1=La qualité est catastrophique* et *5=On ne voit pas la différence avec une question faite par un humain*.

Ces différents critères sont demandés à l'utilisateur au travers d'une page Web spécifique, permettant une évaluation unitaire et séquentielle des questions à évaluer. La figure 6.3 donne un aperçu de la page Web utilisée par les évaluateurs. L'énoncé, le thème et les différentes réponses (bonne et mauvaises) sont présentées sur la partie gauche de l'écran, et l'évaluation s'y fait élément par élément sur la partie droite. Les critères notés sur une échelle à plusieurs valeurs sont présentés sous forme de curseurs, alors que les critères binaires sont présentés sous forme de commutateurs. Par défaut, les curseurs se situent en leur milieu, correspondant à une valeur mitigée, et les commutateurs sont situés sur leur coté positif.

A l'heure actuelle, les questions générées sont indépendantes les unes des autres, même lorsqu'elles concernent le même thème. Chaque évaluation ne concerne par conséquent qu'une et une seule question. Nous n'avons pas à l'heure actuelle d'évaluation portant sur des questionnaires (ensemble de questions liées). La couverture du thème et le degré d'intersection entre les questions ne sont donc pas mesurés au travers de cette évaluation.

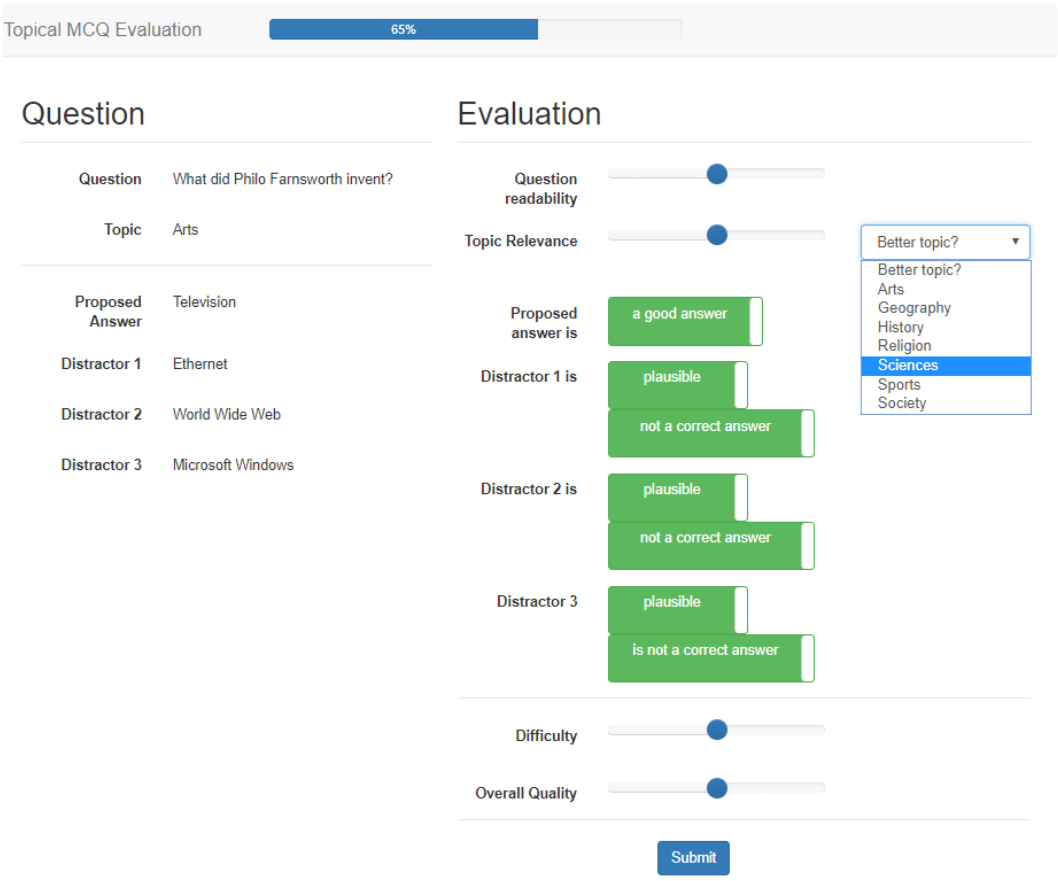


Fig. 6.3.: Page d'évaluation d'une question à choix multiples

Les critères utilisés pour cette évaluation ont été définis dans l'optique d'offrir un compromis acceptable entre le temps passé par les évaluateurs, et la précision des résultats obtenus. On évaluera ainsi globalement l'énoncé sans utiliser de distinction permettant de caractériser les éventuels problèmes liés à la grammaire ou au sens de ce dernier. De même, nous ne demandons pas aux évaluateurs de comparer les distracteurs, sur des critères de longueur, de redondance ou de diversité par exemple. Cependant, nous demandons aux évaluateurs un retour sur chaque composant des questions, et les distracteurs doivent être évalués indépendamment. On obtient ainsi, à partir de ces critères, une vue d'ensemble précise sur la qualité de chaque question évaluée, tout en permettant aux évaluateurs de traiter un maximum de questions.

6.3 Résultats et discussion

En se basant sur les évaluations collectées par l'application Web, nous avons intégré à cette dernière la possibilité de calculer en temps réel les statistiques des notations attribuées par l'ensemble des évaluateurs. Pour être utiles, ces statistiques ne prennent en compte que les questions qui ont été évaluées par l'ensemble des évaluateurs (Par

exemple, si l'évaluateur 1 a évalué toutes les questions, et que l'utilisateur 2 n'en a évalué que 50, les statistiques ne seront calculées que sur ces 50 questions).

Ces statistiques sont disponibles en temps réel sur l'application Web d'évaluation (voir figure 6.4). Il est ainsi possible d'y trouver le nombre de questions évaluées par l'ensemble des évaluateurs n_Q (celles sur lesquelles les statistiques sont calculées), le nombre d'évaluateurs total ayant participé à l'étude n_P, et le nombre total d'évaluations n_E avec $n_E \geq n_P \times n_Q$.

La page de statistiques nous permet ainsi d'observer la moyenne et l'écart type des réponses des évaluations. Cette moyenne est calculée sur l'échelle de 1 à 5 pour les résultats admettant plusieurs valeurs, et entre 0 et 1 dans le cas des évaluations binaires. Nous normalisons par la suite ces résultats avec des valeurs entre 0 et 1, reportées dans la table 6.2.

Topical MCQ Evaluation

- Questions evaluated: 300
- Raters: 3
- Evaluations: 900

Measure	Average	STD
QUESTION_READIBILITY	4.6068	0.7733366383908061
TOPIC_RELEVANCE	3.4301	1.6937446549524224
OVERALL_QUALITY	3.8955	1.1188217149396948
DIFFICULTY	3.1797	1.257830985004056
DISTRACTOR_PLAUSIBLE	0.92677931	0.2603814646368276
DISTRACTOR_CORRECT	0.98669616	0.03910938341818392
GOODANSWER	0.9647	0.1846132004617342

Measure	PercentageAgreement	HubertKappaAgreement
QUESTION_READIBILITY	0.95578231292517	0.91156462585034
TOPIC_RELEVANCE	0.826530612244898	0.653061224489796
OVERALL_QUALITY	0.7925170068027211	0.5850340136054422
DIFFICULTY	0.7346938775510204	0.4693877551020409
DISTRACTOR1_PLAUSIBLE	0.8741496598639455	0.7482993197278911
DISTRACTOR2_PLAUSIBLE	0.9047619047619048	0.8095238095238095
DISTRACTOR3_PLAUSIBLE	0.891156462585034	0.782312925170068
DISTRACTOR1_CORRECT	0.9932088285229211	0.9780457764176579
DISTRACTOR2_CORRECT	0.9964177014426984	0.9941369770796267
DISTRACTOR3_CORRECT	0.9881554504043756	0.9879057239050791
GOODANSWER	0.9285714285714286	0.8571428571428572

Fig. 6.4.: Résultats calculés automatiquement par l'application Web d'évaluation.

Nous avons complété les mesures de moyenne et d'écart type avec des mesures permettant de calculer les différences de notation entre les évaluateurs. Ces mesures permettent ainsi de juger si les évaluateurs émettent des avis similaires ou non sur les critères d'évaluation. Dans cette optique, nous avons retenu deux mesures, le coefficient de corrélation (LIN, 1989) et le Kappa de Hubert (HUBERT, 1977). Nous

avons choisi de retenir ici le Kappa de Hubert dans la mesure où il s'agit d'une extension compatible avec plus de deux évaluateurs du Kappa de Cohen (COHEN, 1960). Cette mesure a été sélectionnée car elle semble largement utilisée par la communauté scientifique lorsqu'il est nécessaire de mesurer l'accord inter-évaluateurs avec plus de trois évaluateurs (WARRENS, 2010).

Chacun des 3 évaluateurs a évalué les 300 questions (l'interface étant configuré pour ne pas admettre d'évaluation supplémentaire). Une fois l'évaluation terminée, nous avons calculé à partir de ces résultats les mesures de moyenne, écart type, kappa de Hubert et concordance. Ces résultats sont présentés dans la table 6.2.

Critère	Moyenne	Écart type	Hubert κ	%Concordance
Énoncé				
Q1	0,90	0,19	0,89	0,94
Q2	0,61	0,42	0,63	0,82
Bonne réponse				
K1	0,96	0,20	0,84	0,92
Mauvaises réponses (Distracteurs)				
D1	0,99	0,04	0,99	0,99
D2	0,90	0,30	0,88	0,75
Evaluation globale				
G1	0,55	0,31	0,71	0,43
G2	0,72	0,28	0,49	0,74

Tab. 6.2.: Table de résultats de l'évaluation humaine. Les résultats sont normalisés entre 0 et 1

Discussion : Concernant les résultats de cette évaluation, plusieurs points peuvent être commentés : (i) la qualité des questions générées, (ii) l'aspect thématique des questions générées et (iii) la difficulté des questions générées.

(i) La qualité des questions générées : Les différents critères permettant d'évaluer la qualité des questions générées affichent tous un retour extrêmement positif. L'évaluation révèle ainsi que 90% des questions sont lisibles et compréhensibles. Par ailleurs, concernant le choix de la bonne réponse et des distracteurs, les mesures $K1$ et $D1$ indiquent toutes les deux un résultat proche de 100% correspondant à un taux d'erreur très faible sur la validité des réponses proposées. Les mesures d'écart type et de concordance ("*almost perfect agreement*" (VIERA et GARRETT, 2005)) qui accompagnent ces valeurs semblent également confirmer que l'avis des évaluateurs sur ces critères convergent fortement, ce qui renforce la crédibilité de ces pourcentages extrêmement élevés. On peut ainsi dire que le fait de combiner des modèles

de questions avec des entités extraites depuis des bases de connaissances offre une réelle efficacité concernant la formation de la structure des questions générées.

(ii) L'aspect thématique des questions générées : Les résultats concernant l'aspect thématique des questions générées sont plus discutables. En effet, la moyenne de la pertinence de la question au sein du thème a été évaluée à 61%, ce qui constitue un résultat encourageant, mais améliorable. Par ailleurs, avec un écart type de 42% et un Hubert kappa de 63% ("*substantial agreement*" (VIERA et GARRETT, 2005)), on note une grande disparité entre les évaluations du thème. Ces résultats concernant la corrélation entre questions et thèmes peuvent être expliqués en prenant en considération l'absence de support thématique à la question. En effet, la notion de thème reste très subjective dans la mesure où les entités peuvent généralement être décrites par plusieurs facettes. Par exemple, la question "Dans quelle université a étudié Niels Bohr ?" est-elle liée au domaine de la physique, ou de l'éducation. Suivant l'aspect que considèrent les évaluateurs, les deux réponses sont possibles. Des interviews des évaluateurs ont confirmé que l'attribution d'un thème a parfois été problématique, plusieurs thèmes pouvant correspondre à la question traitée.

(iii) La difficulté des questions générées : Les évaluateurs ont reporté que la difficulté des questions générées était proche de 50%, ce qui dans la notation de 1 à 5, correspond à la notion de difficulté "*équilibrée*". On peut toutefois noter qu'avec un écart type de 31%, et une mesure de concordance de 43% ("*moderate agreement*" (VIERA et GARRETT, 2005)), cette mesure admet une variation très élevée entre les évaluateurs. Cette mesure doit ainsi être considérée avec un certain recul. Il est par ailleurs important de prendre en compte le fait que les évaluateurs possèdent tous les trois un niveau d'éducation élevé (Master et plus), et que par conséquent, les résultats de cette mesure seraient probablement très différents venant de la part d'élèves de collèges ou lycées.

Avec une moyenne de satisfaction globale de 72%, on peut finalement conclure cette discussion en affirmant que les résultats du générateur de questions sont encourageants. La structure des questions générées est cohérente, les réponses proposées sont adaptées, et bien que pouvant être améliorée, la corrélation entre questions et thèmes est tout de même significative.

6.4 Évaluation automatique du générateur d'énoncé

Parallèlement à l'évaluation des questions à choix multiples thématiques générés, présentée dans la section précédente, nous avons également mis en place une évaluation

automatique pour mesurer l'efficacité du générateur de phrases en langage naturel, utilisé notamment pour construire les énoncés à partir des arbres syntaxiques comme présenté en section 5.1.4. Nous y présentons un ensemble de règles permettant de convertir un arbre syntaxique en phrase grammaticalement correcte.

Ainsi, pour tester l'efficacité de notre générateur de phrases, nous avons utilisé des jeux de données de phrases afin de les convertir en arbres syntaxiques. A partir de ces arbres syntaxiques, nous avons utilisé notre générateur de phrases pour recréer les phrases d'origine. Nous avons ensuite comparé les phrases originales et les phrases obtenues après génération pour évaluer l'efficacité du générateur.

Cette évaluation a été lancée sur un total de 1430 phrases issues de deux jeux de données différents. Au total, 1394 phrases ont été regénérées de façon exactement identiques aux phrases originales, soit 97,48%. Nous avons cependant remarqué que parmi les 36 phrases qui différaient de leurs phrases d'origine, 14 d'entres elles avaient en réalité une forme grammaticale valide et équivalente à la phrase originale. C'est notamment le cas pour la phrase originale "*where the Seneca indians lived ?*", qui a été regénérée en "*where lived the Seneca indians ?*", ou pour la phrase "*what type of government does Australia currently have ?*", regénérée en "*what type of government does Australia have currently ?*". Ainsi, parmi l'ensemble de phrases utilisées pour mener cette évaluation, on considère que 1408 d'entres elles ont donné lieu à la génération d'une phrase valide, soit un taux de réussite de 98,46%.

Discussion : Les résultats obtenus avec cette évaluation ont permis de valider le fait que le générateur de phrases utilisé pour construire l'énoncé des questions est extrêmement fiable en ce qui concerne la structure grammaticale des phrases générées. On peut d'ailleurs remarquer que ce résultat est fortement corrélé aux résultats de l'évaluation précédente (voir section 6.3), et plus particulièrement avec le critère $Q1$, qui évalue la validité grammaticale de la phrase.

6.5 Conclusion

Dans ce chapitre, nous avons présenté les expérimentations menées dans l'optique d'évaluer les résultats produits par notre générateur de questions à choix multiples thématiques. Ces expérimentations englobent ainsi une validation intermédiaire, utilisée pour évaluer automatiquement les résultats de notre module de génération de phrases en langage naturel, et une évaluation finale, au cours de laquelle nous avons demandé à plusieurs juges d'évaluer la qualité de questions générées automatiquement. Ces expérimentations nous amènent à conclure que la solution proposée

montre des résultats prometteurs, tout particulièrement en ce qui concerne la qualité linguistique et grammaticale des phrases générées.

7 Conclusion et perspectives

7.1 Synthèse des contributions

Dans cette thèse, nous avons présenté notre générateur automatique de questions à choix multiples thématiques à partir de bases de connaissances. L'objectif de ce générateur est en effet de générer des questions à choix multiples relatives à un sujet donné. Nous avons présenté les différentes méthodes de l'état de l'art concernant la génération automatique de questions, et montré qu'à l'heure de la rédaction de ce mémoire, ce problème est encore ouvert. Nous avons ensuite proposé une solution de bout en bout, traitant séparément les différentes problématiques nécessaires à la mise en place de ce générateur de questions à choix multiples.

Dans une première partie, nous avons introduit la notion de thèmes au sein de bases de connaissances, afin de pouvoir utiliser les données contenues dans ces dernières pour générer nos questions à choix multiples. Pour cela, nous avons présenté notre solution permettant d'utiliser la structure de Wikipedia, et plus particulièrement, son double graphe d'articles et de catégories, pour créer automatiquement un ensemble de thèmes. Toujours à l'aide de la structure de Wikipedia, nous avons approvisionné ces thèmes avec les articles issus de l'encyclopédie, tout en classant ces derniers par ordre d'importance au sein de chaque thème. Cette première étape nous a ainsi permis d'obtenir un ensemble d'entités, classées au sein de thématiques, et utilisables avec toutes les bases de connaissances dérivées de Wikipedia, comme Yago, Freebase ou DBpedia.

Dans une seconde partie, nous avons détaillé notre approche consistant à utiliser des modèles pour générer l'énoncé des questions. La définition de ces modèles nous a permis d'obtenir des résultats significatifs dans la précision orthographique et grammaticale des phrases générées, tout en utilisant des modèles suffisamment génériques pour être adaptables à une thématique spécifique. En effet, la présence de variables au sein de nos modèles de questions nous permet d'y insérer des entités de types prédéfinis. Ces entités sont extraites à partir des bases de connaissances issues de Wikipedia, au sein desquelles nous avons apporté la notion de thème, nous permettant ainsi de sélectionner des entités relatives à un thème prédéterminé. Ces modèles de questions résolvent le problème de la génération en langage naturel, en permettant la génération d'énoncés basés sur des modèles prédéterminés, mais

en sacrifiant toutefois partiellement la généricité de la solution. Les modèles sont en effet limités aux modèles existants, qui doivent être soit créés manuellement, soit extraits de questions existantes. Nous avons également présenté notre approche consistant à intégrer massivement des modèles à partir des travaux de (ABUJABAL et al., 2017a), de façon à proposer un ensemble initial de modèles opérationnels pour la génération de questions à choix multiples.

Dans une troisième partie, nous présentons de quelle façon nous avons combiné les données issues des bases de connaissances dérivées de Wikipedia avec nos modèles de questions afin de mettre en place notre générateur de questions à choix multiples. Ainsi, à partir d'un thème prédéterminé, notre générateur est en mesure d'identifier un modèle et un ensemble d'entités afin de construire l'énoncé de la question, et d'y associer la bonne réponse. Nous présentons également dans cette partie les procédés utilisés pour associer à chaque question un ensemble de distracteurs. Nous présentons tout particulièrement les étapes utilisées pour sélectionner au sein des bases de connaissances des candidats qui soient à la fois pertinents vis à vis du contexte de la question (thématiques), mais également du point de vue de la bonne réponse, afin de renforcer leur crédibilité. La combinaison entre cette génération d'énoncés thématiques, et l'intégration de distracteurs prenant en compte le contexte, nous permet finalement de réaliser l'objectif initial : *Générer automatiquement des questions à choix multiples thématiques à partir de bases de connaissances*.

Cette thèse a donné lieu à un prototype fonctionnel. Ce prototype, voulu avec un maximum de généricité, suit également la décomposition en 3 sous-parties indépendantes présentées ci-dessus. Ainsi, les questions peuvent-être générées à partir de tout ensemble de modèles et d'entités thématiques, qui elles-même peuvent être extraites à partir de n'importe quel thème choisi.

Nous avons également présenté dans ce manuscrit les évaluations qui ont été mises en place pour valider l'efficacité du générateur de questions à choix multiples, et donc la qualité des questions générées. Ces expériences ont ainsi permis de conclure que les résultats sont satisfaisants, tout particulièrement en ce qui concerne la qualité orthographique et grammaticale des énoncés générés. Les résultats ont également montré que l'aspect thématique des questions générées était encourageant, bien que certains aspects puissent être améliorés.

Pour terminer la synthèse des travaux qui composent cette thèse, on peut conclure que le générateur de questions à choix multiples thématiques que nous proposons permet de compléter efficacement les approches existantes dans le domaine de la génération de questions. En effet, la solution proposée est capable de générer des questions à partir de tout ensemble d'entités classées par thèmes, et tout ensemble de modèles de questions compatibles. Ces travaux de thèse comprennent également

des expérimentations permettant, à partir d'une évaluation manuelle des résultats obtenus, de valider l'approche présentée dans ce manuscrit.

7.2 Améliorations et travaux futurs

Certaines améliorations pourraient renforcer la qualité des travaux effectués, mais n'ont pas pu être conduites pour des raisons de temps ou de contraintes légales notamment. C'est le cas de l'évaluation, qui pourrait être imaginée dans un contexte plus large. A l'heure actuelle, seuls trois évaluateurs ont évalué le résultat final de ces travaux. Par ailleurs, bien que de grades différents, ces trois évaluateurs travaillent dans le même domaine, l'informatique. Une étude à grande échelle, regroupant des personnes avec différents niveaux scolaires, différents âges, différentes spécialités pourrait apporter un avantage considérable pour une évaluation complète, et l'établissement de statistiques pour chacune de ces catégories. Il est cependant important de noter qu'une telle évaluation est compliquée pour plusieurs raisons, la plus importante étant la logistique. En effet, même en utilisant l'application Web d'évaluation, trouver des contacts acceptant de la diffuser au sein des collèges, lycées et universités est une tâche compliquée. Autre raison, la législation française, avec la CNIL notamment, et la législation européenne, avec la nouvelle loi GDPR, compliquent significativement les procédures de collecte et de traitement des données. Ainsi, l'intégration d'une évaluation à grande échelle reste à l'ordre du jour, même si à l'heure actuelle nous l'avons substituée par une évaluation plus restreinte.

Une seconde amélioration qui pourrait être pertinente réside dans la gestion de la difficulté des questions. En effet, générer des questions dans un niveau de difficulté donné serait un avantage considérable pour utiliser les questions générées dans un contexte éducatif. C'est d'ailleurs dans cette optique que nous avons intégré la notion de difficulté dans notre évaluation, à travers le critère $G1$, à l'aide duquel les évaluateurs doivent donner une note selon la difficulté éprouvée vis-à-vis de la question. L'objectif ici était à terme de corréler la difficulté des questions, exprimée par les utilisateurs, avec un ensemble de niveaux scolaires à définir. La gestion des niveaux de difficulté est une tâche beaucoup plus compliquée, comme le montre notamment (Seyler et al., 2017).

La génération des questions et énoncés par modèles permet d'obtenir une grande qualité des résultats, mais cette approche est limitée par le fait de devoir définir les modèles manuellement. L'utilisation détournée de leur but initial des templates de (Abujabal et al., 2017a) a permis d'obtenir un grand nombre de modèles automatiquement. Cependant il est nécessaire de développer de nouvelles approches pour alimenter ce jeu de templates. Une piste de recherche serait d'utiliser en entrée des

questions générées par un système de réseaux de neurones puis de rechercher dans la base de connaissances les entités correspondantes pour essayer de généraliser les relations entre ces entités pour en déduire des modèles de la même forme que ceux que nous utilisons aujourd'hui. Une problématique posée par cette approche est celle de la détermination des thèmes pour ces questions. Actuellement ce sont les labels des relations dans la base de connaissances qui servent à déterminer ces thèmes avec succès sur Freebase. Les principales bases de connaissances (DBpedia, Wikidata) possèdent toutes des labels pour leurs relations, nous estimons donc que notre approche serait généralisable.

Enfin, il est important de noter qu'à l'heure actuelle, les questions sont générées indépendamment les unes des autres, y compris au sein d'un même thème. Pour générer des questionnaires complets, il est important que les questions soient cohérentes entres elles. Ainsi, une amélioration notable envisagée dans les futurs travaux serait de prendre en compte les notions de *couverture* et d'*intersection* pour transformer cette génération de questions indépendantes en génération de questionnaires. Ainsi, on entend par **couverture** le fait de générer des questions qui permettent de traiter d'un maximum de sujets différents au sein du thème passé en paramètre, et par **intersection** le fait de minimiser les redondances entre chaque pair de questions d'un questionnaire. Bien qu'étroitement liés, ces deux aspects doivent être traités simultanément pour garantir la qualité des questionnaires générés. Cette amélioration aurait pour but de refléter de façon plus précise les connaissances des personnes interrogées sur le thème choisi, en reflétant une connaissance globale de ce dernier, tout en garantissant que les questions générées soit suffisamment différentes entre elles pour qu'un sujet n'ayant qu'une connaissance partielle du thème ne puisse pas répondre à toutes les questions. Ce passage de questions indépendantes en questionnaires permettrait une utilisation plus large de la solution proposée, notamment dans les milieux éducatifs pour lesquels il est important que les questions proposés aux étudiants soient liés. Une évaluation spécifique aux critères de couverture et d'intersection sera alors nécessaire pour juger de la viabilité de la solution dans un contexte éducatif.

Parallèlement, dans une optique de visibilité et de reproductibilité des travaux réalisés au cours de cette thèse, nous avons prévu de rendre public l'ensemble des résultats obtenus. Ces résultats comprennent notamment le code source, qui sera publié prochainement de façon Open Source, mais aussi l'ensemble des résultats intermédiaires qui pourraient être réutilisés par la communauté. Ainsi, sont actuellement disponibles pour une consultation libre les éléments suivants :

— **La démonstration de Fouilla** [1] **:** Une application Web sur laquelle l'utilisateur peut naviguer et tester par lui-même l'intérêt d'un filtre par thème lors de l'exploration de DBpedia.

— **Les jeux de données thématiques** [2] **:** Une page Web permettant de télécharger librement une version RDF des thèmes extraits dans le chapitre 3. Le format RDF permet d'utiliser ces thèmes en complément d'autres travaux qui pourraient avoir besoin d'intégrer une dimension thématique.

— **La liste des thèmes** [3] **:** Une page Web qui présente modestement la liste des différents thèmes extraits dans le chapitre 3.

Le code source n'est pas disponibles à l'heure actuelle en raison de sa structure complexe, composés de nombreux sous modules (construction des thèmes, extraction des templates, Génération de questions, etc.) qui nécessite une documentation détaillée.

7.3 Perspectives

Ces travaux de thèse offrent de nombreuses perspectives, et un vaste champs d'applications qui ne se limite pas au domaine strict de la génération automatique de questions à choix multiples. En effet, ramené à un contexte éducatif, ces travaux permettent de générer des questions qui sont en lien avec des chapitres ou des leçons existants. On pourrait tout particulièrement adapter ces travaux dans le cadre de MOOCs (*Massive Open Online Courses*), pour lesquels il est souvent difficile d'évaluer traditionnellement les acquis de connaissances. Le système proposé ici pourrait en effet, à partir d'un MOOC mis en ligne par un enseignant, générer automatiquement un ensemble de questions correspondant à la thématique de ce dernier. Cette évaluation automatique aurait ainsi le double avantage de (1) permettre aux élèves d'avoir un retour sur leur compréhension du cours, et (2) permettre aux enseignants de cibler les points qui sont bien intégrés par les élèves, et retravailler ceux qui le sont moins.

Toujours dans le domaine des MOOCs, on pourrait également imaginer la combinaison de ce système avec des algorithmes de *machine learning* permettant aux étudiants un enseignement sur mesure. En effet, à l'aide de questions personnalisées, dépendant des réponses faites aux questions précédentes, on peut cerner les acquis et les lacunes des étudiants, et ainsi les rediriger vers des cours, ou morceaux de cours leur permettant de retravailler les points mal compris.

1. `http://demo-satin.telecom-st-etienne.fr/fouilla`
2. `http://datasets-satin.telecom-st-etienne.fr/traynaud/fouilla`
3. `http://datasets-satin.telecom-st-etienne.fr/traynaud/Topics.html`

Le projet Mooctab [4] est un projet ITEA3 [5] visant à promouvoir le déploiement massif de MOOCs sur un support tablette. Dans le cadre de ce projet, la totalité des cours est stocké sur un serveur central, et transmis aux salles de classes par l'intermédiaire d'une Mooctab-Box. Le système que nous proposons peut s'intégrer à ce type d'architecture en proposant de compléter les cours créés par les enseignants par des questions à choix multiples générés automatiquement.

4. http://mooctab.com/
5. https://itea3.org/project/mooc-tab.html

Bibliographie

ABUJABAL, Abdalghani, Mohamed YAHYA, Mirek RIEDEWALD et Gerhard WEIKUM (2017a). „Automated Template Generation for Question Answering over Knowledge Graphs“. In : *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, p. 1191-1200 (cf. p. 96, 97, 146, 147).

ABUJABAL, Abdalghani, Rishiraj Saha ROY, Mohamed YAHYA et Gerhard WEIKUM (2017b). „QUINT : Interpretable Question Answering over Knowledge Bases“. In : *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017 - System Demonstrations*, p. 61-66 (cf. p. 96).

AFZAL, Naveed et Ruslan MITKOV (2014). „Automatic generation of multiple choice questions using dependency-based semantic relations“. In : *Soft Comput.* 18.7, p. 1269-1281 (cf. p. 35).

AGARWAL, Manish et Prashanth MANNEM (2011). „Automatic Gap-fill Question Generation from Text Books“. In : *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2011, Portland, Oregon, USA, June 24, 2011*, p. 56-64 (cf. p. 14).

AHO, Alfred V (2003). *Compilers : principles, techniques and tools (for Anna University), 2/e*. Pearson Education India (cf. p. 89).

AL-YAHYA, Maha (2014). „Ontology-based multiple choice question generation“. In : *The Scientific World Journal* 2014 (cf. p. 18, 24, 29, 30, 33-35, 38, 111, 136).

ALDABE, Itziar et Montse MARITXALAR (2010). „Automatic Distractor Generation for Domain Specific Texts“. In : *Advances in Natural Language Processing, 7th International Conference on NLP, IceTAL 2010, Reykjavik, Iceland, August 16-18, 2010*, p. 27-38 (cf. p. 21).

ALSUBAIT, Tahani, Bijan PARSIA et Ulrike SATTLER (2013). „A similarity-based theory of controlling mcq difficulty“. In : *2013 Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*. IEEE, p. 283-288 (cf. p. 30, 32).

ALSUBAIT, Tahani, Bijan PARSIA et Uli SATTLER (2014). „Generating Multiple Choice Questions From Ontologies : Lessons Learnt“. In : *Proceedings of the 11th International Workshop on OWL : Experiences and Directions (OWLED 2014) co-located with 13th International Semantic Web Conference on (ISWC 2014), Riva del Garda, Italy, October 17-18, 2014*. P. 73-84 (cf. p. 17, 30, 33, 36, 38).

AUER, Sören, Christian BIZER, Georgi KOBILAROV et al. (2007). „DBpedia : A Nucleus for a Web of Open Data“. In : *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. T. 4825. Lecture Notes in Computer Science. Springer, p. 722-735 (cf. p. 53).

AWAD, Ahmed Ezz et Mohamed Yehia DAHAB (2014). „Automatic Generation of Question Bank Based on Pre-defined Templates“. In : *?* (Cf. p. 14).

BAILEY, Charles D, Julia N KARCHER et Barbara CLEVENGER (1998). „A comparison of the quality of multiple-choice questions from CPA exams and textbook test banks“. In : *The Accounting Educators' Journal* 10.2 (cf. p. 33, 34, 38, 111, 136).

BECKER, Lee, Sumit BASU et Lucy VANDERWENDE (2012). „Mind the Gap : Learning to Choose Gaps for Question Generation“. In : *Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, p. 742-751 (cf. p. 15, 16, 41).

BENEDETTI, Fabio, Domenico BENEVENTANO et Sonia BERGAMASCHI (2016). „Context Semantic Analysis : A Knowledge-Based Technique for Computing Inter-document Similarity“. In : *Similarity Search and Applications - 9th International Conference, SISAP 2016, Tokyo, Japan, October 24-26, 2016. Proceedings*, p. 164-178 (cf. p. 129).

BERNHARD, Delphine, Louis De VIRON, Véronique MORICEAU et Xavier TANNIER (2012). „Question Generation for French : Collating Parsers and Paraphrasing Questions“. In : *D&D* 3.2, p. 43-74 (cf. p. 41).

BHATIA, Arjun Singh, Manas KIRTI et Sujan Kumar SAHA (2013). „Automatic Generation of Multiple Choice Questions Using Wikipedia“. In : *Pattern Recognition and Machine Intelligence - 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings*, p. 733-738 (cf. p. 24, 29, 35, 36).

BLEI, David M., Andrew Y. NG et Michael I. JORDAN (2003). „Latent Dirichlet Allocation“. In : *Journal of Machine Learning Research* 3, p. 993-1022 (cf. p. 45, 51).

BLOOM BENJAMIN, S et David R KRATHWOHL (1956). *Taxonomy of Educational Objectives : The Classification of Educational Goals, by a committee of college and university examiners. Handbook I : Cognitive Domain* (cf. p. 29, 31).

BOLLACKER, Kurt D., Colin EVANS, Praveen PARITOSH, Tim STURGE et Jamie TAYLOR (2008). „Freebase : a collaboratively created graph database for structuring human knowledge“. In : *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, p. 1247-1250 (cf. p. 18, 54).

BORDES, Antoine, Jason WESTON, Ronan COLLOBERT et Yoshua BENGIO (2011). „Learning Structured Embeddings of Knowledge Bases“. In : *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*.

BOUGOUIN, Adrien, Florian BOUDIN et Béatrice DAILLE (2013). „TopicRank : Graph-Based Topic Ranking for Keyphrase Extraction“. In : *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, p. 543-551 (cf. p. 48).

BRIN, Sergey et Lawrence PAGE (1998). „The Anatomy of a Large-Scale Hypertextual Web Search Engine“. In : *Computer Networks* 30.1-7, p. 107-117 (cf. p. 47, 68).

BROWN, Jonathan, Gwen A. FRISHKOFF et Maxine ESKÉNAZI (2005). „Automatic Question Generation for Vocabulary Assessment“. In : *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics, p. 819-826 (cf. p. 23).

CHEPELIANSKII, A. D. (2010). „Towards physical laws for software architecture“. In : *CoRR* abs/1003.5455. arXiv : 1003.5455 (cf. p. 47).

COHEN, Jacob (1960). „A coefficient of agreement for nominal scales“. In : *Educational and psychological measurement* 20.1, p. 37-46 (cf. p. 140).

CORREIA, Rui, Jorge BAPTISTA, Nuno J. MAMEDE, Isabel TRANCOSO et Maxine ESKENAZI (sept. 2010). „Automatic Generation of Cloze Question Distractors“. In : *Second Language Studies : Acquisition, Learning, Education and Technology*. Waseda University, Tokyo, Japan : SLaTE : the ISCA SIG on Speech et Language Technology in Edu (cf. p. 21, 38).

COURSEY, Kino, Rada MIHALCEA et William E. MOEN (2009). „Using Encyclopedic Knowledge for Automatic Topic Identification“. In : *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*. Sous la dir. de Suzanne STEVENSON et Xavier CARRERAS. ACL, p. 210-218 (cf. p. 47).

CROVITZ, Darren et W Scott SMOOT (2009). „Wikipedia : Friend, not foe“. In : *English Journal* 98.3, p. 91-97.

CUBRIC, Marija et Milorad TOSIC (2011). „Towards automatic generation of e-assessment using semantic web technologies“. In : *International Journal of e-Assessment* 1 (cf. p. 24, 29).

CURTO, Sérgio, Ana Cristina MENDES et Luisa COHEUR (2012). „Question generation based on lexico-syntactic patterns learned from the web“. In : *Dialogue & Discourse* 3.2, p. 147-175 (cf. p. 43).

DE MARNEFFE, Marie-Catherine et Christopher D MANNING (2008). *Stanford typed dependencies manual*. Rapp. tech. Technical report, Stanford University (cf. p. 89, 120).

DEEMTER, Kees van, Mariët THEUNE et Emiel KRAHMER (2005). „Real versus Template-Based Natural Language Generation : A False Opposition ?“ In : *Computational Linguistics* 31.1, p. 15-24 (cf. p. 42).

DEERWESTER, Scott, Susan T DUMAIS, George W FURNAS, Thomas K LANDAUER et Richard HARSHMAN (1990). „Indexing by latent semantic analysis“. In : *Journal of the American society for information science* 41.6, p. 391-407.

DING, Li, Rong PAN, Timothy W. FININ et al. (2005). „Finding and Ranking Knowledge on the Semantic Web“. In : *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*. Sous la dir. d'Yolanda GIL, Enrico MOTTA, V. Richard BENJAMINS et Mark A. MUSEN. T. 3729. Lecture Notes in Computer Science. Springer, p. 156-170 (cf. p. 78).

DOROW, Beate et Dominic WIDDOWS (2003). „Discovering Corpus-Specific Word Senses“. In : *EACL 2003, 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12-17, 2003, Agro Hotel, Budapest, Hungary*. The Association for Computer Linguistics, p. 79-82 (cf. p. 21).

DU, Xinya, Junru SHAO et Claire CARDIE (2017). „Learning to ask : Neural question generation for reading comprehension“. In : *arXiv preprint arXiv :1705.00106*, p. 1342-1352 (cf. p. 41).

DUAN, Nan, Duyu TANG, Peng CHEN et Ming ZHOU (2017). „Question Generation for Question Answering“. In : *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Sous la dir. de Martha PALMER, Rebecca HWA et Sebastian RIEDEL. Association for Computational Linguistics, p. 866-874 (cf. p. 18, 20).

FADER, Anthony, Stephen SODERLAND et Oren ETZIONI (2011). „Identifying Relations for Open Information Extraction“. In : *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, p. 1535-1545 (cf. p. 96).

FENG, Yue, Ebrahim BAGHERI, Faezeh ENSAN et Jelena JOVANOVIC (2017). „The state of the art in semantic relatedness : a framework for comparison“. In : *The Knowledge Engineering Review* 32, e10 (cf. p. 104).

FINKEL, Jenny Rose, Trond GRENAGER et Christopher D. MANNING (2005). „Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling“. In : *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. Sous la dir. de Kevin KNIGHT, Hwee Tou NG et Kemal OFLAZER. The Association for Computer Linguistics, p. 363-370 (cf. p. 14).

FLEISS, Joseph L (1971). „Measuring nominal scale agreement among many raters“. In : *Psychological bulletin* 76, p. 378-382.

FOREHAND, Mary (2010). „Emerging perspectives on learning, teaching, and technology“. In : sous la dir. de M. OREY. Global Text Project, Creative Commons. Chap. Bloom's taxonomy, p. 41-47 (cf. p. 14).

FOULONNEAU, Muriel (2011). „Generating Educational Assessment Items from Linked Open Data : The Case of DBpedia“. In : *The Semantic Web : ESWC 2011 Workshops - ESWC 2011 Workshops, Heraklion, Greece, May 29-30, 2011, Revised Selected Papers*. Sous la dir. de Raul GARCIA-CASTRO, Dieter FENSEL et Grigoris ANTONIOU. T. 7117. Lecture Notes in Computer Science. Springer, p. 16-27 (cf. p. 26).

GATT, Albert et Emiel KRAHMER (2018). „Survey of the State of the Art in Natural Language Generation : Core tasks, applications and evaluation“. In : *Journal of Artificial Intelligence Research* 61, p. 65-170 (cf. p. 40).

GATT, Albert et Ehud REITER (2009). „SimpleNLG : A Realisation Engine for Practical Applications“. In : *ENLG 2009 - Proceedings of the 12th European Workshop on Natural Language Generation, March 30-31, 2009, Athens, Greece*. Sous la dir. d'Emiel KRAHMER et Mariët THEUNE. The Association for Computer Linguistics, p. 90-93 (cf. p. 41).

GRAESSER, Arthur C et Robert A WISHER (2001). *Question generation as a learning multiplier in distributed learning environments*. US Army Research Institute for the Behavioral et Social Sciences, Alexandria VA (cf. p. 1, 21, 23, 85, 122).

GRONLUND, Norman Edward (1982). *Constructing achievement tests*. Prentice Hall (cf. p. 29, 35).

GUO, Qi, Chinmay KULKARNI, Aniket KITTUR, Jeffrey P. BIGHAM et Emma BRUNSKILL (2016). „Questimator : Generating Knowledge Assessments for Arbitrary Topics“. In : *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. Sous la dir. de Subbarao KAMBHAMPATI. IJCAI/AAAI Press, p. 3726-3732 (cf. p. 13, 26, 41).

HU, Zhiting, Zichao YANG, Xiaodan LIANG, Ruslan SALAKHUTDINOV et Eric P. XING (2017). „Toward Controlled Generation of Text“. In : *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Sous la dir. de Doina PRECUP et Yee Whye TEH. T. 70. Proceedings of Machine Learning Research. PMLR, p. 1587-1596 (cf. p. 41).

HUBERT, Lawrence (1977). „Kappa revisited“. In : *Psychological Bulletin* 84.2, p. 289 (cf. p. 139).

HULPUS, Ioana, Conor HAYES, Marcel KARNSTEDT et Derek GREENE (2013). „Unsupervised graph-based topic labelling using dbpedia“. In : *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. Sous la dir. de Stefano LEONARDI, Alessandro PANCONESI, Paolo FERRAGINA et Aristides GIONIS. ACM, p. 465-474 (cf. p. 46).

JÄRVELIN, Kalervo et Jaana KEKÄLÄINEN (2002). „Cumulated gain-based evaluation of IR techniques“. In : *ACM Transactions on Information Systems (TOIS)* 20.4, p. 422-446 (cf. p. 73).

KAPTEIN, Rianne, Pavel SERDYUKOV, Arjen P. de VRIES et Jaap KAMPS (2010). „Entity ranking using Wikipedia as a pivot“. In : *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*. Sous la dir. de Jimmy HUANG, Nick KOUDAS, Gareth J. F. JONES et al. ACM, p. 69-78 (cf. p. 47).

KILGARRIFF, Adam (1995). *BNC database and word frequency lists* (cf. p. 23).

KIROS, Ryan, Yukun ZHU, Ruslan SALAKHUTDINOV et al. (2015). „Skip-Thought Vectors“. In : *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Sous la dir. de Corinna CORTES, Neil D. LAWRENCE, Daniel D. LEE, Masashi SUGIYAMA et Roman GARNETT, p. 3294-3302 (cf. p. 26).

KLEIN, Dan et Christopher D. MANNING (2003). „Accurate Unlexicalized Parsing“. In : *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan*. Sous la dir. d'Erhard W. HINRICHS et Dan ROTH. ACL, p. 423-430 (cf. p. 14, 18).

KRÄENBRING, Jona, Tika Monzon PENZA, Joanna GUTMANN et al. (2014). „Accuracy and completeness of drug information in Wikipedia : a comparison with standard textbooks of pharmacology“. In : *PloS one* 9.9, e106930 (cf. p. 61).

KRIPPENDORFF, Klaus (1970). „Estimating the reliability, systematic error and random error of interval data“. In : *Educational and Psychological Measurement* 30.1, p. 61-70.

KRISHNA, Shenai (1992). *Introduction to database and knowledge-base systems*. T. 28. World Scientific (cf. p. 53).

LANGVILLE, Amy N et Carl D MEYER (2011). *Google's PageRank and beyond : The science of search engine rankings*. Princeton University Press (cf. p. 47).

LE, Nguyen-Thinh, Tomoko KOJIRI et Niels PINKWART (2014). „Automatic Question Generation for Educational Applications - The State of Art“. In : *Advanced Computational Methods for Knowledge Engineering - Proceedings of the 2nd International Conference on Computer Science, Applied Mathematics and Applications, ICCSAMA 2014, 8-9 May, 2014, Budapest, Hungary*. Sous la dir. de Tien Van DO, Hoai An Le THI et Ngoc Thanh NGUYEN. T. 282. Advances in Intelligent Systems and Computing. Springer, p. 325-338 (cf. p. 12).

LEVENSHTEIN, Vladimir I (1966). „Binary codes capable of correcting deletions, insertions, and reversals“. In : *Soviet physics doklady*. T. 10. 8, p. 707-710 (cf. p. 22).

LIN, Lawrence I-Kuei (1989). „A Concordance Correlation Coefficient to Evaluate Reproducibility“. In : *Biometrics* 45.1, p. 255-268 (cf. p. 139).

LIU, Ming, Rafael A. CALVO et Vasile RUS (2012). „G-Asks : An Intelligent Automatic Question Generation System for Academic Writing Support“. In : *Dialogue & Discourse* 3.2, p. 101-124 (cf. p. 14, 15, 42).

LIU, Ming, Vasile RUS et Li LIU (2018). „Automatic Chinese Multiple Choice Question Generation Using Mixed Similarity Strategy“. In : *IEEE Transactions on Learning Technologies* 11.2, p. 193-202 (cf. p. 35).

MANNING, Christopher D., Prabhakar RAGHAVAN et Hinrich SCHÜTZE (2008). *Introduction to information retrieval*. Cambridge University Press.

MARCUS, Mitchell P., Grace KIM, Mary Ann MARCINKIEWICZ et al. (1994). „The Penn Treebank : Annotating Predicate Argument Structure“. In : *Human Language Technology, Proceedings of a Workshop held at Plainsboro, New Jerey, USA, March 8-11, 1994*. Morgan Kaufmann, p. 114-119 (cf. p. 22, 23, 89, 120).

MEDELYAN, Olena, Ian H WITTEN et David MILNE (2008). „Topic indexing with Wikipedia“. In : *Proceedings of the AAAI WikiAI workshop*. T. 1, p. 19-24 (cf. p. 46).

MIHALCEA, Rada et Andras CSOMAI (2007). „Wikify ! : linking documents to encyclopedic knowledge“. In : *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*. Sous la dir. de Mário J. SILVA, Alberto H. F. LAENDER, Ricardo A. BAEZA-YATES et al. ACM, p. 233-242 (cf. p. 47).

MIHALCEA, Rada et Paul TARAU (2004). „TextRank : Bringing Order into Text“. In : *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*. ACL, p. 404-411 (cf. p. 48).

MITKOV, Ruslan et Le An HA (2003). „Computer-aided generation of multiple-choice tests“. In : *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*. Association for Computational Linguistics, p. 17-22 (cf. p. 22).

MITKOV, Ruslan, Le An HA, Andrea VARGA et Luz RELLO (2009). „Semantic similarity of distractors in multiple-choice tests : extrinsic evaluation“. In : *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. Association for Computational Linguistics, p. 49-56 (cf. p. 22, 29, 35, 37).

MORENO, Rafael, Rafael J MARTÍNEZ et José MUÑIZ (2006). „New guidelines for developing multiple-choice items“. In : *Methodology* 2.2, p. 65-72 (cf. p. 29).

NARENDRA, Annamaneni, Manish AGARWAL et Rakshit SHAH (2013). „Automatic Cloze-Questions Generation“. In : *Recent Advances in Natural Language Processing, RANLP 2013, 9-11 September, 2013, Hissar, Bulgaria*. Sous la dir. de Galia ANGELOVA, Kalina BONTCHEVA et Ruslan MITKOV. RANLP 2013 Organising Committee / ACL, p. 511-515 (cf. p. 14, 34, 36, 38, 41).

OLNEY, Andrew McGregor, Arthur C. GRAESSER et Natalie K. PERSON (2012). „Question Generation from Concept Maps“. In : *Dialogue & Discourse* 3.2, p. 75-99 (cf. p. 42).

OREN, Eyal, Renaud DELBRU et Stefan DECKER (2006). „Extending Faceted Navigation for RDF Data“. In : *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*. Sous la dir. d'Isabel F. CRUZ, Stefan DECKER, Dean ALLEMANG et al. T. 4273. Lecture Notes in Computer Science. Springer, p. 559-572 (cf. p. 78).

PAGE, Lawrence, Sergey BRIN, Rajeev MOTWANI et Terry WINOGRAD (1999). *The PageRank Citation Ranking : Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab.

PANDEY, Shivank et KC RAJESWARI (2013). „Automatic Question Generation Using Software Agents for Technical Institutions“. In : *International Journal of Advanced Computer Research* 3.4, p. 307-311 (cf. p. 14, 31).

PAPASALOUROS, Andreas, Konstantinos KANARIS et Konstantinos KOTIS (2008). „Automatic Generation Of Multiple Choice Questions From Domain Ontologies“. In : *IADIS International Conference e-Learning 2008, Amsterdam, The Netherlands, July 22-25, 2008. Proceedings*. Sous la dir. de Miguel Baptista NUNES et Maggie MCPHERSON. IADIS, p. 427-434 (cf. p. 17, 24, 25, 37, 38).

PEASE, Adam, Ian NILES et John LI (2002). „The suggested upper merged ontology : A large ontology for the semantic web and its applications“. In : *Working notes of the AAAI-2002 workshop on ontologies and the semantic web*. T. 28, p. 7-10 (cf. p. 54).

PHO, Van-Minh, Anne-Laure LIGOZAT et Brigitte GRAU (2015). „Distractor Quality Evaluation in Multiple Choice Questions“. In : *Artificial Intelligence in Education - 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015. Proceedings*. Sous la dir. de Cristina CONATI, Neil T. HEFFERNAN, Antonija MITROVIC et M. Felisa VERDEJO. T. 9112. Lecture Notes in Computer Science. Springer, p. 377-386 (cf. p. 34).

PINO, Juan et Maxine ESKÉNAZI (2009). „Semi-automatic generation of cloze question distractors effect of students' L1“. In : *ISCA International Workshop on Speech and Language Technology in Education, SLaTE 2009, Warwickshire, England, UK, September 3-5, 2009*. ISCA, p. 65-68 (cf. p. 23, 24).

PINO, Juan, Michael HEILMAN et Maxine ESKENAZI (2008). „A selection strategy to improve cloze question quality“. In : *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, p. 22-32 (cf. p. 36, 38).

PIWEK, Paul et Kristy Elizabeth BOYER (2012). „Varieties of Question Generation : Introduction to this Special Issue“. In : *Dialogue & Discourse* 3.2, p. 1-9 (cf. p. 12, 41).

POLK, Tracy, Melissa P. JOHNSTON et Stephanie EVERS (2015). „Wikipedia Use in Research : Perceptions in Secondary Schools“. In : *TechTrends* 59.3, p. 92-102.

POWLEY, Brett et Robert DALE (2007). „Evidence-Based Information Extraction for High Accuracy Citation and Author Name Identification“. In : *Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 2007, 8th International Conference, Carnegie Mellon University, Pittsburgh, PA, USA, May 30 - June 1, 2007. Proceedings, CD-ROM*. Sous la dir. de David A. EVANS, Sadaoki FURUI et Chantal SOULÉ-DUPUY. CID (cf. p. 14).

RAKANGOR, Sheetal et Dr YR GHODASARA (2015). „Literature review of automatic question generation systems“. In : *International journal of scientific and research publications* 5 (cf. p. 13).

RANDOLPH, Justus J (2010). „Free-Marginal Multirater Kappa (multirater K [free]) : An Alternative to Fleiss' Fixed-Marginal Multirater Kappa.“ In : *Advances in Data Analysis Classification* 4.

RATINOV, Lev-Arie et Dan ROTH (2009). „Design Challenges and Misconceptions in Named Entity Recognition“. In : *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*. Sous la dir. de Suzanne STEVENSON et Xavier CARRERAS. ACL, p. 147-155 (cf. p. 14).

RAYNAUD, Tanguy, Julien SUBERCAZE et Frédérique LAFOREST (2018a). „Fouilla : Navigating DBpedia by Topic“. In : *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, p. 1907-1910 (cf. p. 8, 77, 81).

RAYNAUD, Tanguy, Julien SUBERCAZE et Frederique LAFOREST (2018b). „Thematic Question Generation over Knowledge Bases“. In : *Proceedings of the 2018 International Conference on Web Intelligence*. IEEE (cf. p. 9).

REITER, Ehud et Robert DALE (2000). *Building Natural Language Generation Systems*. Cambridge University Press (cf. p. 40).

ROCHA, Oscar Rodríguez et Catherine FARON-ZUCKER (2018). „Automatic Generation of Quizzes from DBpedia According to Educational Standards“. In : *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*. Sous la dir. de Pierre-Antoine CHAMPIN, Fabien L. GANDON, Mounia LALMAS et Panagiotis G. IPEIROTIS. ACM, p. 1035-1041 (cf. p. 19).

ROCHA, Oscar Rodríguez, Catherine FARON-ZUCKER et Alain GIBOIN (2018). „Extraction of Relevant Resources and Questions from DBpedia to Automatically Generate Quizzes on Specific Domains“. In : *Intelligent Tutoring Systems - 14th International Conference, ITS 2018, Montreal, QC, Canada, June 11-15, 2018, Proceedings*. Sous la dir. de Roger NKAMBOU, Roger AZEVEDO et Julita VASSILEVA. T. 10858. Lecture Notes in Computer Science. Springer, p. 380-385 (cf. p. 19).

ROSS, Kenneth A., Angel JANEVSKI et Julia STOYANOVICH (2005). „A Faceted Query Engine Applied to Archaeology“. In : *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*. Sous la dir. de Klemens BÖHM, Christian S. JENSEN, Laura M. HAAS et al. ACM, p. 1334-1337 (cf. p. 78).

RUS, Vasile, Zhiqiang CAI et Art GRAESSER (2008). „Question generation : Example of a multi-year evaluation campaign“. In : *Proc WS on the Question Generation Shared Task Evaluation Challenge* (cf. p. 12).

SEYLER, Dominic, Mohamed YAHYA et Klaus BERBERICH (2017). „Knowledge Questions from Knowledge Graphs“. In : *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2017, Amsterdam, The Netherlands, October 1-4, 2017*. Sous la dir. de Jaap KAMPS, Evangelos KANOULAS, Maarten de RIJKE, Hui FANG et Emine YILMAZ. ACM, p. 11-18 (cf. p. 18, 25, 31, 37, 42, 122, 147).

SHEN, Yikang, Zhouhan LIN, Chin-Wei HUANG et Aaron C. COURVILLE (2017). „Neural Language Modeling by Jointly Learning Syntax and Lexicon“. In : *CoRR* abs/1711.02013. arXiv : 1711.02013 (cf. p. 41).

SINGHAL, Amit (2001). „Modern Information Retrieval : A Brief Overview“. In : *IEEE Data Eng. Bull.* 24.4, p. 35-43 (cf. p. 104).

STASASKI, Katherine et Marti A. HEARST (2017). „Multiple Choice Question Generation Utilizing An Ontology“. In : *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*. Sous la dir. de Joel R. TETREAULT, Jill BURSTEIN, Claudia LEACOCK et Helen YANNAKOUDAKIS. Association for Computational Linguistics, p. 303-312 (cf. p. 26, 27).

SUCHANEK, Fabian M., Gjergji KASNECI et Gerhard WEIKUM (2007). „Yago : a core of semantic knowledge“. In : *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. Sous la dir. de Carey L. WILLIAMSON, Mary Ellen ZURKO, Peter F. PATEL-SCHNEIDER et Prashant J. SHENOY. ACM, p. 697-706 (cf. p. 53).

TOUTANOVA, Kristina, Dan KLEIN, Christopher D. MANNING et Yoram SINGER (2003). „Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network“. In : *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. Sous la dir. de Marti A. HEARST et Mari OSTENDORF. The Association for Computational Linguistics (cf. p. 14).

TVAROZEK, Michal et Mária BIELIKOVÁ (2007). „Personalized Faceted Navigation in the Semantic Web“. In : *Web Engineering, 7th International Conference, ICWE 2007, Como, Italy, July 16-20, 2007, Proceedings*. Sous la dir. de Luciano BARESI, Piero FRATERNALI et Geert-Jan HOUBEN. T. 4607. Lecture Notes in Computer Science. Springer, p. 511-515 (cf. p. 78).

VIBHANDIK, Seepshree S. et Rucha SAMANT (2014). „An Overview of Automatic Question Generation Systems“. In : *Int. J. Science, Engineering and Technology Research* 3.10, p. 2612-16 (cf. p. 13).

VIERA, Anthony J. et Joanne M. GARRETT (2005). „Understanding interobserver agreement : the kappa statistic“. In : *Family Medicine* 37.5, p. 360-363 (cf. p. 140, 141).

VRANDECIC, Denny et Markus KRÖTZSCH (2014). „Wikidata : a free collaborative knowledgebase“. In : *Commun. ACM* 57.10, p. 78-85 (cf. p. 53).

WARRENS, Matthijs J. (2010). „Inequalities between multi-rater kappas“. In : *Adv. Data Analysis and Classification* 4.4, p. 271-286 (cf. p. 140).

WIEMER-HASTINGS, Peter, K WIEMER-HASTINGS et A GRAESSER (2004). „Latent semantic analysis“. In : *Proceedings of the 16th international joint conference on Artificial intelligence*. Citeseer, p. 1-14 (cf. p. 71, 104).

WISEMAN, Sam, Stuart M. SHIEBER et Alexander M. RUSH (2018). „Learning Neural Templates for Text Generation“. In : *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Sous la dir. d'Ellen RILOFF, David CHIANG, Julia HOCKENMAIER et Jun'ichi TSUJII. Association for Computational Linguistics, p. 3174-3187 (cf. p. 41).

XING, Wenpu et Ali A. GHORBANI (2004). „Weighted PageRank Algorithm“. In : *2nd Annual Conference on Communication Networks and Services Research (CNSR 2004), 19-21 May 2004, Fredericton, N.B., Canada*. IEEE Computer Society, p. 305-314.

YAO, Xuchen, Gosse BOUMA et Yi ZHANG (2012). „Semantics-based Question Generation and Implementation“. In : *Dialogue & Discourse* 3.2, p. 11-42 (cf. p. 41).

ZHIROV, A. O., O. V. ZHIROV et D. L. SHEPELYANSKY (2010). „Two-dimensional ranking of Wikipedia articles“. In : *CoRR* abs/1006.4270 (cf. p. 47).

ZHOU, Qingyu, Nan YANG, Furu WEI et al. (2017). „Neural Question Generation from Text : A Preliminary Study“. In : *Natural Language Processing and Chinese Computing - 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8-12, 2017, Proceedings*, p. 662-671 (cf. p. 41).

A

Liste des thèmes triés alphabétiquement

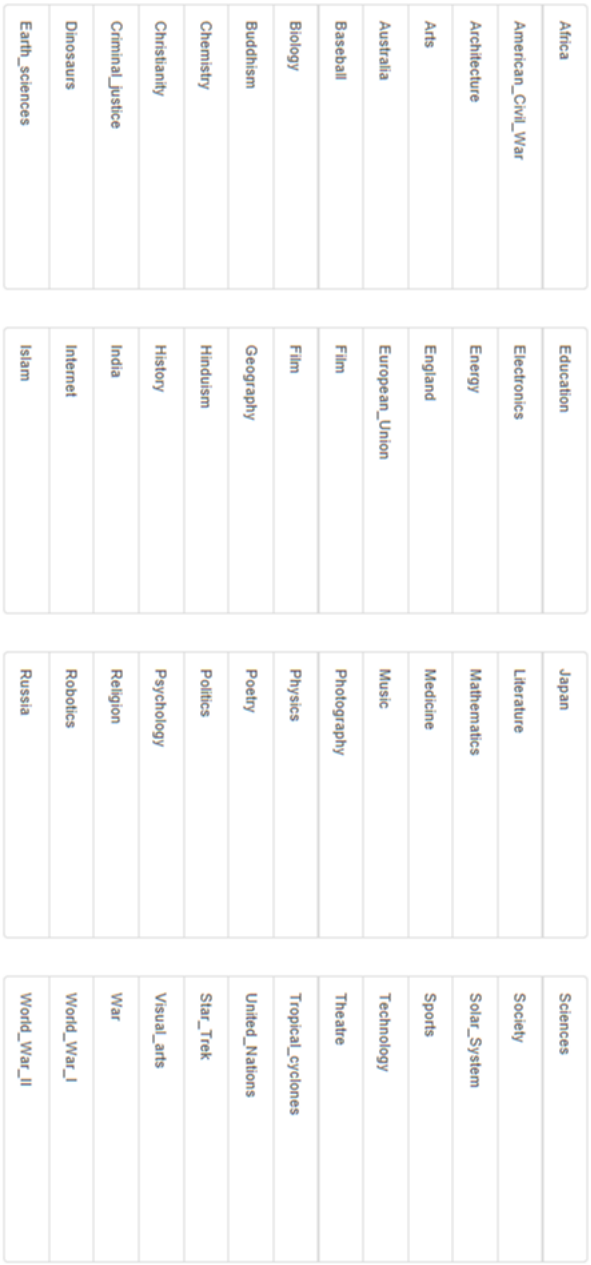


Fig. A.1.: Liste exhaustive des 52 thèmes extraits automatiquement, triés alphabétiquement

B

Liste des thèmes avec mention de hiérarchie

Fig. B.1.: Liste exhaustive des 52 thèmes extraits automatiquement, regroupés au sein d'une hiérarchie à 2 niveaux

Table des figures

Liste des tableaux

Liste des publications

Tanguy Raynaud, Julien Subercaze, Frederique Laforest : **Thematic Question Generation over Knowledge Bases**. In *Proceedings of the International Conference on Web Intelligence*, Chile, December 03-06, 2018. IEEE 2018.

Tanguy Raynaud, Julien Subercaze, Frederique Laforest : **Fouilla : Navigating DBpedia by Topic**. In *Proceedings of the 2018 ACM on Conference on Information and Knowledge Management*, CIKM 2018, Italy, October 22-26, 2018. ACM 2018.