Méthodes d'apprentissage automatique pour l'analyse de corpus jurisprudentiels

Charles Condevaux

Collège Doctoral Languedoc-Roussillon

THÈSE

Pour obtenir le grade de
Docteur

Délivré par l'**Université de Nîmes**

Préparée au sein de l'école doctorale **Risques & Société**
Et de l'unité de recherche **CHROME**

Spécialité: **Informatique**

Présentée par **Charles Condevaux**

Méthodes d'apprentissage automatique pour l'analyse de corpus jurisprudentiels

Soutenue le 15 décembre 2021 devant le jury composé de

Mohand Boughanem	PR	IRIT	Rapporteur
Massih-Reza Amini	PR	LIG	Rapporteur
Jérôme Azé	PR	LIRMM	Examinateur
Jacky Montmain	PR	IMT Mines Alès	Examinateur
Françoise Seyte	MCF	MRE UM	Examinatrice
Stéphane Mussard	PR	Unîmes	Directeur
Sébastien Harispe	MA	IMT Mines Alès	Encadrant
Guillaume Zambrano	MCF	Unîmes	Encadrant

unîmes UNIVERSITÉ

Remerciements

Je remercie Stéphane, Sébastien et Guillaume pour leur soutien et leur disponibilité durant ces trois années de thèse. Je tiens à souligner la qualité et la constance de l'encadrement dont j'ai pu profiter malgré les contraintes liées à la crise sanitaire.

Je remercie la DSI et Bernard Dardy d'avoir monté et maintenu les serveurs de calcul indispensables à l'apprentissage automatique ; les membres du laboratoire CHROME qui ont contribué ou influencé de près ou de loin ce travail ; Arthur Charpentier qui m'a accueilli plusieurs mois à l'UQAM et la région Occitanie pour le financement de cette thèse.

Je remercie Monsieur Mohand BOUGHANEM, Professeur à l'Université Toulouse III Paul Sabatier (IRIT), et Monsieur Massih-Reza AMINI, Professeur à l'Université Grenoble Alpes (LIG), d'accepter d'être les rapporteurs de ma thèse. Je remercie aussi Monsieur Jérome Azé, Professeur à l'Université de Montpellier (LIRMM), Monsieur Jacky Montmain, Professeur à l'IMT Mines d'Alès, et Madame Françoise Seyte, Maître de Conférences (HDR) à l'Université de Montpellier (MRE), d'accepter d'être examinateurs de ma thèse.

Résumé

Titre : Méthodes d'apprentissage automatique pour l'analyse de corpus jurisprudentiels

Le développement de l'apprentissage profond et l'émergence de modèles Transformers permettent la résolution de tâches variées dans de nombreux domaines d'application. Lorsque les données traitées concernent des décisions de justice, la rareté des bases de données annotées, la longueur des documents et la nécessaire compréhension des prédictions contraignent l'exploitation de telles architectures. Cette thèse propose plusieurs contributions permettant l'estimation de modèles d'apprentissage machine en informatique juridique ; elle traite en particulier de problématiques d'indexation et de classification de demandes notamment dans un contexte de justice prédictive. L'entraînement en faible échantillon imposé par la rareté des bases est abordé à l'aide de techniques d'apprentissage one-shot et d'augmentation de données. La longueur des documents (séquences longues) est traitée par la modification du mécanisme d'attention des Transformers en exploitant un contexte à la fois local, éparse et global. Cette modification permet une réduction du coût en mémoire et de fortes capacités d'adaptation et d'extrapolation à partir de modèles préalablement entraînés. Enfin, deux méthodes d'attribution dérivées de la valeur de Shapley et de complexité linéaire sont proposées, assurant des propriétés souhaitables de la théorie des jeux coopératifs ainsi que l'interprétabilité des prédictions.

Mots clés : TALN, Transformers efficients, IA interprétable, théorie des jeux coopératifs, apprentissage one-shot, justice prédictive.

Abstract

Title : Machine learning methods for the analysis of jurisprudential corpus

The development of deep learning and the emergence of Transformer models make it possible to solve various tasks in many application areas. When the data processed concerns court decisions, the scarcity of labeled datasets, the length of the documents and the required interpretability of predictions limit the use of such architectures. This thesis proposes several contributions to the estimation of machine learning models in legal informatics ; in particular, it addresses the problems of indexing and classifying claims in a predictive justice context. Low-sampling training imposed by datasets scarcity is addressed using one-shot learning and data augmentation techniques. Document length (long sequences) is addressed by modifying the attention mechanism of Transformers and exploiting both local, sparse and global context. This modification is memory efficient and allows strong adaptation and extrapolation capabilities from pre-trained models. Finally, two Shapley-derived attribution methods with linear complexity are proposed, ensuring desirable properties from cooperative game theory as well as explainability of predictions.

Key words : NLP, efficient Transformers, explainable AI, cooperative game theory, one-shot learning, predictive justice.

Table des matières

Liste des figures

Liste des tableaux

Lexique

Batch	Sous-ensemble d'entrées extraites du jeu de données.
BPC	Bits Per Character, métrique dérivée de l'entropie utilisée pour les modèles de langue.
Checkpoint	Ensemble des poids extraits d'un modèle pré-entraîné.
Clustering	Partitionnement non supervisé.
Dropout	Méthode de régularisation supprimant aléatoirement des connexions d'un réseau de neurones durant l'entraînement.
Epoque	Itération sur l'ensemble des éléments d'un jeu de données.
Embedding	Représentation vectorielle d'une entrée ou d'une caractéristique (plongement sémantique).
Fine-tuning	Adaptation des paramètres d'un modèle préalablement entraîné dans le but de résoudre une nouvelle tâche.
GRU	Gated Recurrent Unit, réseau de neurones récurrents dérivé des LSTM pour le traitement de séquences.
LSTM	Long Short-Term Memory, réseau de neurones récurrents avec mémoire pour le traitement de séquences.
One-shot learning	Méthode modifiant une tâche de classification en un problème de similarité.
Optimiseur	Fonction mettant à jour les paramètres d'un modèle à l'aide des gradients calculés lors de la phase d'entraînement.

Pooling	Fonction de compression non paramétrique permettant de réduire la taille d'une séquence ou d'une image.
Seq-to-one	Fonction prenant en entrée une séquence et produisant un seul élément en sortie.
Seq-to-seq	Fonction prenant en entrée une séquence et produisant une séquence en sortie.
TALN	Traitement automatique du langage naturel.
TF-IDF	Term Frequency-Inverse Document Frequency, représentation vectorielle d'une séquence par le biais des fréquences des tokens la constituant.
Token	Plus petite unité composant une phrase qui peut être un mot, une syllabe ou un caractère selon la tâche.
Tokenizer	Fonction découpant une séquence en tokens.
Warmup	Augmentation incrémentale du taux d'apprentissage en début d'entraînement pour améliorer la convergence.

Présentation générale

i Motivations et positionnement

Une décision de justice est un document écrit retraçant le déroulement d'une affaire judiciaire à travers une description d'éléments factuels et les raisons de fait et de droit motivant la solution adoptée par la juridiction saisie.

Pour une question juridique donnée, l'ensemble des décisions rendues et relatives à celle-ci sont regroupées sous le terme de jurisprudence, permettant de retracer un historique des jugements. Le travail d'un juriste, plus spécifiquement d'un avocat, consiste à consulter et analyser ces documents afin d'anticiper l'issue d'un contentieux judiciaire. L'objectif étant d'évaluer la situation et de défendre un client en optimisant les chances pour celui-ci d'obtenir une résolution du litige qui lui est favorable.

L'activité de collecte et d'analyse est alors au centre du travail des juristes qui se retrouvent face à des volumes de données conséquents continuellement enrichis. Les quantités de décisions produites par les tribunaux varient en fonction de la juridiction, les 36 cours d'appel françaises générant à elles seules près de 10.000 décisions par mois en moyenne, rendant l'exploitation manuelle de telles quantités de décisions complexe, chronophage, voire souvent impossible. Afin de diminuer le coût de cette tâche, des outils sont développés pour faciliter la recherche d'informations. Ces derniers prennent généralement la forme de moteurs de recherche [1], permettant par le biais de simples requêtes par mots clés ou méta-données, d'obtenir un sous-ensemble filtré de décisions. Les résultats issus de ces moteurs nécessitent un tri manuel par la suite car ils privilégient

1. Dalloz, Lamyline, LexisNexis...

souvent l'exhaustivité à la pertinence et requièrent l'usage de nombreux termes et critères parfois complexes à exprimer. Un tri par le biais d'un article, d'une loi ou d'une référence est dès lors infiniment plus simple à mettre en oeuvre qu'un tri thématique par demandes pour lesquelles le sens du résultat et le quantum [2] obtenu sont associés. S'ajoute à la présélection des décisions, le recours aux compétences du juriste qui doit lui-même extraire les éléments qu'il juge importants dans des documents souvent longs, techniques et complexes. L'information recherchée pouvant être localisée dans une zone restreinte, distribuée de façon éparse, ou encore revêtir une forme implicite dans certains cas, rendant cette activité d'autant plus chronophage.

La demande de transparence de la justice est par ailleurs de plus en plus forte. Pourtant, les bases de données compilant la jurisprudence ne sont en réalité ni aisément accessibles, ni gratuites. Les principaux moteurs de recherche sont détenus par des acteurs privés qui adoptent des modèles économiques basés sur la vente d'accès par abonnements à des bases elles-mêmes achetées auprès de certaines juridictions, notamment la cour de cassation. Les tarifs proposés par ces entreprises étant souvent prohibitifs, un particulier recherchant une information ou une jurisprudence devra dans la majorité des cas se contenter d'un portail gratuit et partiellement fourni tel que Légifrance [3]. Un professionnel sera lui contraint à souscrire à un service payant. L'autre frein à la transparence et à l'ouverture des données concerne des contraintes liées au respect de certaines règles éthiques, relatives notamment à la confidentialité des parties et des juges. Les parlementaires ont ainsi adopté l'article 33 de la loi du 23 mars 2019 traitant des questions liées à l'open data des décisions en matière judiciaire. Afin d'être publiées, celles-ci doivent préalablement être anonymisées afin de limiter le risque de ré-identification et d'empêcher tout calcul de statistiques descriptives ou prédictives exploitant l'identité des juges ou des greffiers [4], elles-mêmes illégales. Enfin, le régime permettant la mise à disposition des décisions de justice sous forme électronique au public a été précisé par le décret n°2020-797 du 29 juin 2020 et complété le 28 avril 2021 [5] par un arrêté fixant le calendrier de publication des données par juridiction.

2. quantité demandée : somme d'argent, nombre de jours de garde...
3. https://www.legifrance.gouv.fr/
4. Par exemple comparer la sévérité des juges sur un type de demande spécifique.
5. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000043426865

L'ensemble des principes cités, qu'ils soient liés à la recherche ou l'extraction d'informations, au traitement de grandes bases jurisprudentielles, à la compréhension de décisions de justice ou l'anonymisation encouragent l'automatisation afin de limiter le temps consacré à des tâches répétitives et souvent peu productives des professionnels. Le langage judiciaire étant peu accessible et difficilement compréhensible pour les profanes, l'usage d'algorithmes spécifiques permettant d'obtenir des réponses simples à des questions plus complexes a un intérêt certain pour l'individu qui souhaite, de manière autonome, s'informer sur des problèmes juridiques courants sans recourir à un spécialiste.

Enfin, la théorie juridique admet que deux affaires aux contextes semblables et sur un intervalle de temps restreint doivent mener à des décisions identiques, que ce soit sur le sens du résultat ou du quantum : c'est l'hypothèse de stabilité du juge. Cette stabilité permet d'envisager la justice prédictive, définie comme la détermination au moyen de modèles et de techniques algorithmiques « de la probabilité de succès d'une affaire au moyen de l'analyse des décisions antérieures rendues en la même matière » [Buat-Ménard, 2019]. En supposant que le droit est le produit de décisions de juges faisant l'application de normes et qu'un modèle statistique est en mesure de modéliser ce phénomène, il serait possible, à faible coût, d'estimer l'issue d'un litige avec un certain niveau de risque [Tagny Ngompe, 2020]. Ces modèles prédictifs n'auraient pas vocation à remplacer un juge, mais pourraient néanmoins servir de support et filtrer certaines affaires en orientant les parties vers une résolution alternative du conflit. Ces algorithmes pourraient enfin sous certaines conditions, mesurer et réduire le risque d'aléa judiciaire en faisant abstraction de facteurs non-objectivables propres à la personnalité des parties, de leurs avocats et des juges, qu'ils soient cognitifs, idéologiques ou moraux.

De nombreux travaux de recherche s'intéressent ainsi à l'informatique juridique. Ils contribuent aux développements de techniques algorithmiques variées permettant l'automatisation de tâches d'intérêt pour l'analyse de vastes corpus de décisions et la définition de modèles prédictifs dédiés au domaine juridique. Ces travaux font très souvent référence au Traitement Automatique du Langage Naturel, à l'utilisation de représentation de connaissances (modèles de connaissances de type ontologies), et tirent de plus en plus fréquemment parti de techniques d'apprentissage machine.

On note par exemple de nombreux travaux sur l'extraction d'informations [Chalkidis et al., 2018], la classification de normes ou de concepts juridiques [Waltl et al., 2017], en particulier sur la législation européenne [Chalkidis et al., 2019b]. La prédiction de l'issue d'une affaire à partir de données factuelles est notamment un sujet majeur de la littérature de l'informatique juridique.

De nombreux travaux d'informatique juridique reposent sur des formulations de problèmes basées sur des tâches de classification [Gonçalves and Quaresma, 2005] ; il s'agit par exemple d'identifier la nature d'un passage d'un texte juridique, de rattacher une décision à une classe prédéfinie, ou de prédire si une demande sera acceptée ou rejetée sur la base de l'analyse d'un descriptif des faits. La littérature liée à l'informatique juridique est ainsi étroitement liée aux développements des techniques de classification notamment de classification de textes. Nous nous attarderons sur différentes problématiques de classification de textes dans le cadre de cette thèse.

Les approches de classification se divisent traditionnellement en deux groupes, l'un relatif aux approches à base de règles, l'autre se basant sur des techniques d'apprentissage automatique [Waltl et al., 2017]. Les systèmes à base de règles reposent sur l'utilisation de règles expertes prédéfinies et déterministes à partir desquelles les classifications seront effectuées. À titre d'exemple, nous pouvons considérer que l'observation de l'expression « Article 700 » [6] dans une décision fait référence à la classe « Demande de dommages-intérêts ». La large littérature dédiée à ce type d'approches a permis de nombreux raffinements, notamment pour augmenter l'expressivité des règles et faciliter leur identification. Deux avantages de cette approche sont très généralement soulignés : (i) l'interprétabilité des décisions permise par l'analyse des règles qui sous-tendent les classifications produites, et (ii) le fait que cette approche ne nécessite pas la constitution de jeu de données au préalable. Néanmoins, bien qu'effective pour répondre aux problématiques les plus simples, ces systèmes souffrent généralement de la difficulté inhérente à l'identification manuelle des règles pertinentes

6. « Le juge condamne la partie tenue aux dépens ou qui perd son procès à payer : 1° A l'autre partie la somme qu'il détermine, au titre des frais exposés et non compris dans les dépens ; [...]» source : https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000028424729

et à la gestion de leur interaction pour l'obtention et la maintenance d'un système efficace.

L'approche dite à base d'apprentissage machine - ou apprentissage automatique - répond aux limites et freins observés dans l'identification manuelle de règles en permettant, par le biais de modèles prédictifs, d'effectuer la classification. L'approche plébiscitée dans ces situations est généralement une approche supervisée qui, exploitant des jeux de données annotés et composés d'exemples (une décision, des faits, des motifs...), va distinguer des modèles prédictifs ayant comme finalité la classification d'exemples encore méconnus. La sélection des modèles est souvent formulée comme un problème d'optimisation visant à minimiser une expression modélisant l'erreur de classification imputable aux modèles évalués. Il s'agit alors par exemple, dans le cas de modèles définis par un ensemble de paramètres, de distinguer ceux permettant de minimiser l'erreur précitée. Ces modèles encodent implicitement des règles décisionnelles permettant de réaliser la classification. Cette approche a rencontré un grand succès dans l'étude de tâches complexes de classification, et cela dans de nombreux domaines applicatifs ; elle est aujourd'hui souvent privilégiée à l'approche à base de règles introduite ci-avant. Nous concentrerons les travaux exposés dans ce manuscrit sur cette approche à base d'apprentissage machine.

Une vaste littérature en Apprentissage Machine et en Traitement Automatique du Langage Naturel étudie en effet activement le domaine de la classification (de textes). Historiquement, certaines approches sont privilégiées et régulièrement employées, parmi lesquelles la régression logistique [Berkson, 1944], les modèles à base d'arbres tels que les forêts aléatoires [Breiman, 2001], ou encore les machines à vecteur de support (SVM) [Drucker et al., 1997]. L'estimation de ces classifieurs requiert la transformation des données textuelles en entrée en une représentation vectorielle exploitable par le système. Pour cela, des caractéristiques spécifiques sont extraites ou construites sous une forme booléenne pour décrire la présence d'une expression particulière comme un article de loi. Cependant, cette définition manuelle n'est que rarement exhaustive et des statistiques supplémentaires peuvent être employées que ce soit à travers des approches par sac de mots (n-grammes) ou des schémas de pondération exploitant des fréquences relatives (TF-IDF) afin de donner de l'importance à

certains termes dans la résolution de tâches de classification dans de nombreux contextes applicatifs [Joachims, 1998].

À titre d'exemple, les SVM ont été utilisés pour effectuer de la classification de normes avec une précision de plus de 90% [Waltl et al., 2017]. L'utilisation d'un modèle semblable basé sur une représentation vectorielle de type TF-IDF a aussi permis l'obtention de performances allant jusqu'à 94% de précision sur une tâche de classification de phrases du domaine légal [de Maat and Winkels, 2009]. Cependant, certaines problématiques restent hors de portée de ces méthodes, particulièrement dans des tâches requérant un niveau de langage sophistiqué et la prise en compte d'une syntaxe complexe.

Les récents développements de l'apprentissage profond, notamment en TALN, impactent tout naturellement l'informatique juridique. Ils sont par exemple largement étudiés pour tenter d'adresser la tâche consistant à prédire l'issue d'un litige. Dans des situations spécifiques, ces modèles à base d'apprentissage profond permettent l'obtention de performances inégalées par des approches dites traditionnelles (régression logistique, les modèles basés sur des arbres ou les SVM [Wei et al., 2019]). Des précisions supérieures à 80% ont par exemple été obtenues sur des décisions pénales chinoises [Zhong et al., 2018]. Diverses performances intéressantes ont plus généralement été établies autour de plusieurs algorithmes de classification [Aletras et al., 2016, Medvedeva et al., 2020] ou d'apprentissage profond par le biais de modèles de type Transformers [O'Sullivan and Beel, 2019, Chalkidis et al., 2019a]. Il existe de plus aujourd'hui des variantes de modèles standards du TALN comme LEGAL-BERT [Chalkidis et al., 2020], adaptés au traitement de textes juridiques anglais et permettant des gains de performances sur un grand nombre de tâches liées à la justice prédictive. La littérature se concentre cependant généralement la plupart du temps sur la langue anglaise. Nous proposons dans ce contexte d'étudier l'apprentissage automatique, dans certains cas profond, pour traiter différentes problématiques d'informatique juridique appliquées à des corpus de langue française. Celles-ci sont précisées par la suite.

ii Thèmes abordés et objectifs

Cette thèse propose d'étudier des problématiques relatives au traitement automatique de la langue dans le cadre d'un exercice de justice prédictive sur des arrêts de cour d'appel et de Cour Européenne des Droits de l'Homme (CrEDH).

Les principales tâches consistent dans un premier temps à reconnaître et catégoriser des types de demandes afin de trier thématiquement et regrouper des affaires pour lesquelles les prétentions des parties sont similaires. Il est ainsi possible d'exploiter ces groupes de décisions et de se focaliser sur la prédiction du sens du résultat dans un second temps. Puisque les demandes sont identiques, les critères discriminants permettant de déduire l'issue se situent nécessairement dans les éléments factuels et dans les éléments juridiques ayant motivé la décision du juge.

Ces différentes tâches peuvent être résolues par le biais de modèles de classification suffisamment sophistiqués et en mesure de traiter un langage juridique complexe et requérant des approches capables de comprendre la synonymie et un niveau de polysémie supplémentaire par rapport au langage naturel ordinaire, connu sous le nom de « texture ouverte » [Hart, 1961]. Par souci de transparence, il est nécessaire, au-delà de l'aspect prédictif, que l'approche utilisée soit en mesure de donner les raisons à l'origine du résultat obtenu [Denis and Varenne, 2019] afin de comprendre la relation entre des éléments factuels et juridiques et la solution apportée par le juge.

La résolution de ces tâches fait appel à l'utilisation de réseaux de neurones exploitant des plongements lexicaux préentrainés et capables de désambiguïsassion, qu'ils soient à base de réseaux récurrents [Elman, 1990] comme les modèles ELMo [Peters et al., 2018] et Flair [Akbik et al., 2019] ou à base d'une architecture de type Transformer [Vaswani et al., 2017] telles que BERT [Devlin et al., 2019] et RoBERTa [Liu et al., 2019]. Cependant, l'exploitation de ces architectures se heurte à des limites liées à la structure des données utilisées en entrée. Puisque le traitement de décisions de justice requiert une expertise métier et que les tâches d'annotation manuelle sont en pratique complexes et chronophages, la quantité d'observations disponibles est fortement restreinte.

Un autre aspect problématique concerne le format et l'hétérogénéité des séquences traitées. Puisque l'objectif est d'exploiter des éléments factuels ou des éléments motivant la décision du juge, les séquences peuvent prendre des tailles variables allant de quelques dizaines à plusieurs milliers de mots. Ce contexte rend l'application des architectures citées difficile car celles-ci présupposent, en particulier pour les Transformers, une longueur limitée.

La prise en compte des contraintes relatives aux échantillons de faible taille et aux séquences longues dans la résolution de tâches de classification sont prioritaires dans cette thèse. Pour cela, plusieurs approches sont présentées, la première faisant appel à l'apprentissage *one-shot* [Fei-fei et al., 2006] capable de généraliser une tâche moyennant très peu d'exemples et en exploitant des critères de similarité par paires d'observations. La seconde se base sur des approches de type Transformers efficients [Tay et al., 2020] pour lesquelles les architectures sont modifiées dans le but de limiter la complexité générale du mécanisme d'attention [Bahdanau et al., 2014], que ce soit par le biais de compression, d'approximation ou de *kernelisation*. Le traitement de séquences longues pour un coût calculatoire relativement limité est dès lors possible.

Le dernier thème traité concerne l'interprétabilité des modèles. Puisque la transparence est un critère recherché chez les juristes, être en mesure de justifier la prédiction d'une boite noire à un humain est nécessaire. Pour cela, est présenté un modèle mêlant une approche par occlusion à un critère dérivé de la valeur de Shapley, permettant de déterminer les éléments (variables, mots, n-grammes) contribuant le plus fortement aux choix émis par le réseau de neurones. Cette approche peut en outre être appliquée aux problématiques relatives aux Transformers puisqu'elle fonctionne indépendamment du modèle.

Les objectifs des travaux exposés dans cette thèse sont les suivants :

1. Identifier des catégories de demandes dans un large corpus de décisions afin de regrouper des décisions similaires et permettre par la suite de prédire un résultat.
2. Prédire un résultat à partir de motifs, c'est-à-dire des éléments juridiques ayant permis au juge de statuer et d'accepter ou refuser une demande spécifique. Ces données étant extrêmement limitées et difficiles à annoter, le cadre de l'apprentissage en petit échantillon est étudié.

3. Prédire un résultat à partir d'éléments factuels. En pratique, les faits sont très hétérogènes même pour des affaires jugées similaires. Le récit pouvant être réduit à quelques phrases dans les cas les plus simples et jusqu'à plusieurs pages dans des cas plus complexes. Les prédictions sont ici effectuées grâce à l'exploitation de modèles capables de compresser l'information et d'exploiter efficacement des séquences longues.
4. Proposer un cadre offrant un certain niveau d'interprétabilité des prédictions afin de limiter l'aspect boite noire de nombreuses approches modernes.

iii Structure de la thèse

Cette thèse se structure autour de 5 chapitres. Le premier d'entre eux contextualise et positionne les travaux proposés par rapport à la littérature existante sur l'analyse de corpus et de décisions juridiques ainsi que sur les outils modernes de traitement automatisé de la langue, notamment à base de plongements lexicaux et de Transformers. Le second chapitre traite principalement du format des données, de leur structure et de l'exploitation de caractéristiques simples permettant le partitionnement thématique de décisions. Le chapitre suivant aborde principalement la résolution de tâches d'apprentissage en très petit échantillon à l'aide de variantes de *one-shot learning* afin d'estimer des modèles en mesure d'offrir des prédictions fiables sous contrainte d'un nombre d'exemples limité. Le quatrième chapitre est lié à la résolution de tâches dont les données en entrée se caractérisent par de longues séquences, non exploitables par les architectures standards de Transformers. Les modèles proposés se basant sur la compression de séquences et l'exploitation d'un contexte à plusieurs niveaux afin d'obtenir des prédictions pour un coût relativement limité. Enfin, un dernier chapitre est consacré à l'interprétabilité des résultats issus des modèles précédents afin de donner un cadre d'analyse compatible avec certaines préoccupations éthiques et de transparence liées à l'automatisation de tâches juridiques.

Chapitre 1

Analyse et traitement de corpus juridiques

Présentation du chapitre

Nous abordons dans ce chapitre l'état de l'art technique sur l'analyse et le traitement de corpus juridiques d'intérêt au regard du positionnement de nos travaux exposés en introduction générale. Nous introduisons pour cela au préalable différentes notions importantes pour la compréhension des outils modernes du Traitement Automatique du Langage Naturel, et plus généralement de l'Apprentissage Automatique. Ces dernières nous permettront par la suite une introduction plus fine des approches relatives à nos contributions qui ont été proposées dans l'état de l'art de l'informatique juridique.

1.1 Outils modernes en traitement automatisé du langage naturel

1.1.1 Auto-encodeurs et plongements sémantiques

Un auto-encodeur est un réseau de neurones conçu pour apprendre, de façon auto-supervisée, la représentation d'un ensemble de données [Kramer, 1991, Hinton and Salakhutdinov, 2006]. Cette représentation est généralement de dimension réduite et a pour objectif de compresser les entrées fournies au modèle. L'absence de supervision permet de traiter de larges volumes de données sans recourir à un étiquetage manuel, ce qui rend ce type d'architecture attrayant dans le traitement automatisé de la langue.

Un auto-encodeur est composé de deux éléments : un encodeur permettant de projeter les données dans un espace latent à l'aide d'une fonction $g_\phi(.)$ paramétrée par ϕ et un décodeur effectuant l'opération inverse à l'aide d'une fonction $f_\theta(.)$ paramétrée par θ. Pour une entrée $\boldsymbol{x} \in \mathbb{R}^d$, un auto-encodeur recherche les fonctions d'encodage et de décodage $g_\phi(.)$ et $f_\theta(.)$ de sorte que $\boldsymbol{x} \approx f_\theta(g_\phi(\boldsymbol{x}))$. C'est donc un réseau de neurones conçu pour apprendre la fonction identité via $f_\theta \circ g_\phi$. L'optimisation des paramètres θ et ϕ est obtenue en construisant une tâche de régression pour laquelle est minimisée la fonction de coût $\mathcal{L} = ||\boldsymbol{x} - f_\theta(g_\phi(\boldsymbol{x}))||^2$. En pratique, cet objectif est difficile à réaliser

puisque la représentation latente $\boldsymbol{z} = g_\phi(\boldsymbol{x}) \in \mathbb{R}^{d'}$ est généralement de dimensionalité inférieure à celle de $\boldsymbol{x}$ ($d' < d$). L'architecture d'un auto-encodeur est présentée en Figure 1.1.

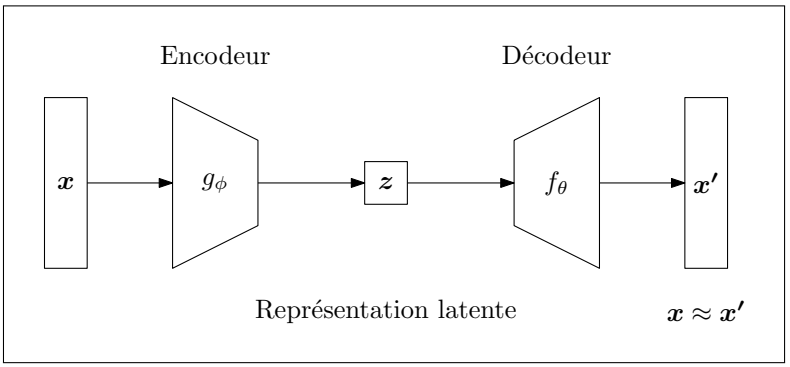


FIGURE 1.1 – Architecture d'un auto-encodeur.

Il existe de nombreuses variantes de ce modèle jouant sur le choix des fonctions f et g ou sur la procédure d'entraînement afin de répondre à des problématiques spécifiques. Les auto-encodeurs de débruitage prennent par exemple en entrée des données modifiées ou bruitées dans le but de reconstruire leur forme initiale [Vincent et al., 2008]. Les auto-encodeurs variationnels se basent quant à eux sur une approche bayésienne : plutôt que de transformer l'entrée en un vecteur, ces derniers les transforment en une distribution [Kingma and Welling, 2014].

Le concept d'auto-encodeur est finalement repris dans le domaine du traitement de la langue avec l'apparition du modèle Word2Vec [Mikolov et al., 2013]. Exploiter des données textuelles requiert la construction d'un vocabulaire $\mathcal{V}$ composé de mots, de syllabes, de caractères (tokens). La manière la plus simple et intuitive de représenter une phrase est sous la forme d'un encodage binaire où les tokens présents prennent la valeur 1. Cependant, puisque les volumes de données textuelles sont généralement immenses et que le vocabulaire peut être composé de plusieurs millions d'éléments, cette approche est peu efficiente. Le but d'un auto-encodeur dans cette situation est de transformer une matrice initialement creuse et très majoritairement composée de 0 en une matrice dense de plus faible dimension. Pour cela Word2Vec pose l'hypothèse qu'un mot peut être représenté par son contexte puis modifie la tâche de régression en une tâche de classification qui ne nécessite pas d'étiquetage manuel.

Le modèle se base sur la distinction entre un contexte et une cible. Pour un mot w à une position t, l'objectif de Word2Vec est de déterminer une probabilité conditionnelle notée y_t :

$$y_t = p(w_t | w_{t-k}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}; \phi, \theta)$$

où k est un hyperparamètre représentant la distance maximale entre la cible et un élément de son contexte. Pour résoudre cette problématique, avec $v = |\mathcal{V}|$ la taille du vocabulaire considéré, l'auto-encodeur prend en entrée la représentation binaire $\boldsymbol{x} \in \mathbb{R}^v$ du contexte[1] et prédit une distribution de probabilités $\hat{y} \in \mathbb{R}^v$ avec pour objectif de prédire le mot manquant. Les fonctions f et g consistent en un simple produit matriciel. En notant $d' < v$ la dimension de la représentation latente et $\boldsymbol{W}_\phi, \boldsymbol{W}_\theta \in \mathbb{R}^{v \times d'}$ des matrices de paramètres :

$$\hat{y} = \sigma\left(\boldsymbol{x}\boldsymbol{W}_\phi \boldsymbol{W}_\theta^\top\right)$$

où $\sigma(.)$ est la fonction Softmax telle que :

$$\sigma_c(\boldsymbol{z}) = \frac{e^{z_c}}{\sum_{i=1}^{i=v} e^{z_i}}, \quad \forall c \in [1, \dots, v]$$

La fonction de coût minimisée est l'entropie croisée entre la prédiction du modèle et la cible. Grâce à sa faible profondeur, Word2Vec peut être estimé sur de grandes quantités de données pour un coût restreint.

Ce modèle a cependant quelques défauts. Premièrement, il est linéaire et n'est donc pas en mesure de modéliser des relations complexes entre une cible et son contexte. Word2Vec ne prend de plus pas en compte les positions des mots dans le contexte sélectionné, il est donc invariant aux permutations alors que l'ordre des éléments est naturellement important pour encoder l'information portée par l'entrée traitée. De plus, la représentation est unique, c'est-à-dire que chaque mot est représenté par un vecteur qui ne change pas selon les phrases. Il est donc impossible de désambiguïser des homonymes. Enfin, le modèle est incapable de traiter des éléments qui n'appartiennent pas à son vocabulaire d'où la nécessité que ce dernier soit suffisamment large.

1. $\boldsymbol{x}_i = 1$ si l'élément de $\mathcal{V}$ associé à l'indice $i \in [1, v]$ est présent dans le contexte ; $\boldsymbol{x}_i = 0$ dans le cas contraire.

D'autres modèles ont été conçus sur des idées proches de Word2Vec. Glove par exemple [Pennington et al., 2014] se base sur des matrices de co-occurrences pour calculer les vecteurs. FastText [Bojanowski et al., 2017] quant à lui se base sur la concatenation de syllabes, cela permet au modèle de fonctionner lorsqu'un mot est inconnu puisque celui-ci peut être décomposé en caractères et syllabes.

1.1.2 Contextualisation des représentations

L'incapacité de désambiguïser les mots a focalisé l'attention sur des modèles capables de contextualiser les représentations sémantiques en fonction de la phrase ou de la position des tokens dans la phrase. La façon la plus naturelle pour cela consiste à exploiter des couches de neurones récurrents qui sont en mesure de conserver l'information sur des séquences plus longues. Les réseaux de neurones de type LSTM [Choromanski et al., 2017] et GRU [Chung et al., 2014] permettent aussi d'estimer des modèles de langue causaux et génératifs dans lesquels un mot doit être prédit à partir de ceux qui le précèdent.

Le modèle ELMo [Peters et al., 2018] se base sur l'architecture BIG-CNN-LSTM [Jozefowicz et al., 2016], c'est-à-dire sur l'exploitation de réseaux convolutifs (CNN - Convolutional Neural Network) et de couches de LSTM. Cela lui permet de produire des plongements sémantiques contextualisés en mesure de traiter des mots hors vocabulaire.
Pour cela, le modèle se base sur plusieurs éléments :

- un traitement au niveau des caractères à l'aide de convolutions afin de reconstituer n'importe quel mot ;
- deux très grandes couches de LSTM bidirectionnels (4096 unités, dimension de 512) ;
- des connexions résiduelles [He et al., 2015] ;
- des prédictions sur les mots et non sur les caractères.

Ce modèle propose aussi l'ajout de paramètres spécifiques permettant de calculer une somme pondérée des différentes sorties des couches pour faciliter le *fine-tuning*. À noter que dans l'usage, il était commun de précalculer les vecteurs puis de les utiliser pour la résolution d'une nouvelle tâche. ELMo change cette approche en proposant un entraînement des poids du réseau directement.

Cette idée permet à l'architecture d'améliorer significativement l'état de l'art sur de nombreuses tâches : classification, reconnaissance d'entités nommées, problématiques de coréférence, étiquetage de rôles sémantiques...

Le modèle Flair [Kitaev et al., 2020] se passe des convolutions en traitant les entrées au niveau des caractères. Cela a pour but de réduire très significativement la taille du vocabulaire et de permettre des gains significatifs sur les tâches de reconnaissance d'entités nommées. Le modèle contrairement au précédent se base sur une approche bidirectionnelle disjointe. Pour cela, deux modèles dotés de deux grandes couches de LSTM sont entrainés séparément sur la base d'une approche causale. Le premier modèle exploite les séquences non modifiées, le second les traite en les inversant. Ainsi, il est possible de procéder à la génération dans les deux sens. Pour le *fine-tuning*, les sorties des deux modèles sont concaténées. Bien que performant, Flair souffre des limites des couches de neurones récurrents pour le traitement des caractères puisque les séquences deviennent mécaniquement longues.

Les LSTM bien que performants et efficients en terme d'utilisation de mémoire, sont en pratique lents et difficiles à entraîner pour plusieurs raisons. La première concerne l'aspect récursif qui empêche la parallélisation. La seconde est relative aux problèmes de perte d'informations sur les séquences longues.

Le mécanisme d'attention [Graves et al., 2014, Bahdanau et al., 2014, Luong et al., 2015] qui consiste à représenter un élément d'une séquence comme la somme pondérée de tous les éléments de la même ou d'une autre séquence est une réponse aux limites des approches dites *seq-to-seq* [Sutskever et al., 2014]. Pour la résolution de ce type de tâches, les auteurs utilisaient traditionnellement des couches de LSTM couplées à une architecture encodeur-decodeur qui nécessitait des vecteurs de taille fixe (*seq-to-one*) pour faire la jonction. Cela avait pour effet de fortement compresser l'information ou d'en perdre une partie significative du fait de la longueur des séquences. L'attention permet deux effets notoires : la parallélisation et des connexions directes entre des éléments éloignés.

Le modèle Transformer [Vaswani et al., 2017] généralise le concept d'attention et permet une avancée significative dans la résolution de toutes les tâches traitant de problématiques *seq-to-seq*. Pour des séquences de longueur

t_q, t_k, t_v représentées par des vecteurs de taille d, l'attention entre des requêtes $\boldsymbol{Q} \in \mathbb{R}^{t_q \times d}$, des clés $\boldsymbol{K} \in \mathbb{R}^{t_k \times d}$ et des valeurs $\boldsymbol{V} \in \mathbb{R}^{t_v \times d}$ est donnée par :

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \sigma\big(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\big)\boldsymbol{V} \tag{1.1}$$

D'après l'équation 1.1, les requêtes ont une connexion directe vers chaque clé. Pour les tâches de traduction, cela permet de connecter les mots d'une langue vers les mots de l'autre langue sans passer par un vecteur de taille fixe. Cette équation s'adapte quelle que soit les longueurs dès lors que $t_k = t_v$. En plus de ce mécanisme, les Transformers sont basés sur plusieurs composants additionnels. L'architecture est illustrée Figure 1.2.

En pratique, plusieurs couches de Transformers sont connectées de manière séquentielle. Chacune d'entre elles possède des connexions résiduelles et des normalisations de couches [Ba et al., 2016] permettant d'assurer une meilleure stabilité durant l'entraînement. S'ajoute une couche de projection composée de deux grandes matrices de poids $\boldsymbol{W}_p, \boldsymbol{W}_q \in \mathbb{R}^{d \times d'}$ où d' est généralement choisi 3 à 4 fois plus grand que d :

$$Projection(\boldsymbol{x}) = \text{GELU}\Big(\boldsymbol{x}\boldsymbol{W}_p\Big)\boldsymbol{W}_q$$

où la fonction GELU [Hendrycks and Gimpel, 2020] est définie par :

$$\text{GELU}(x) = 0.5x(1 + \tanh[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)])$$

Afin de contextualiser et de désambiguïser les mots, les positions des éléments dans la séquence sont ajoutées aux entrées puisque l'attention est invariante aux positions sans cette information. Cela peut être effectué de deux façons : soit en ajoutant avant la première couche une fonction périodique prédéfinie soit en intégrant une couche d'embedding où les entrées sont les positions dans la séquence.

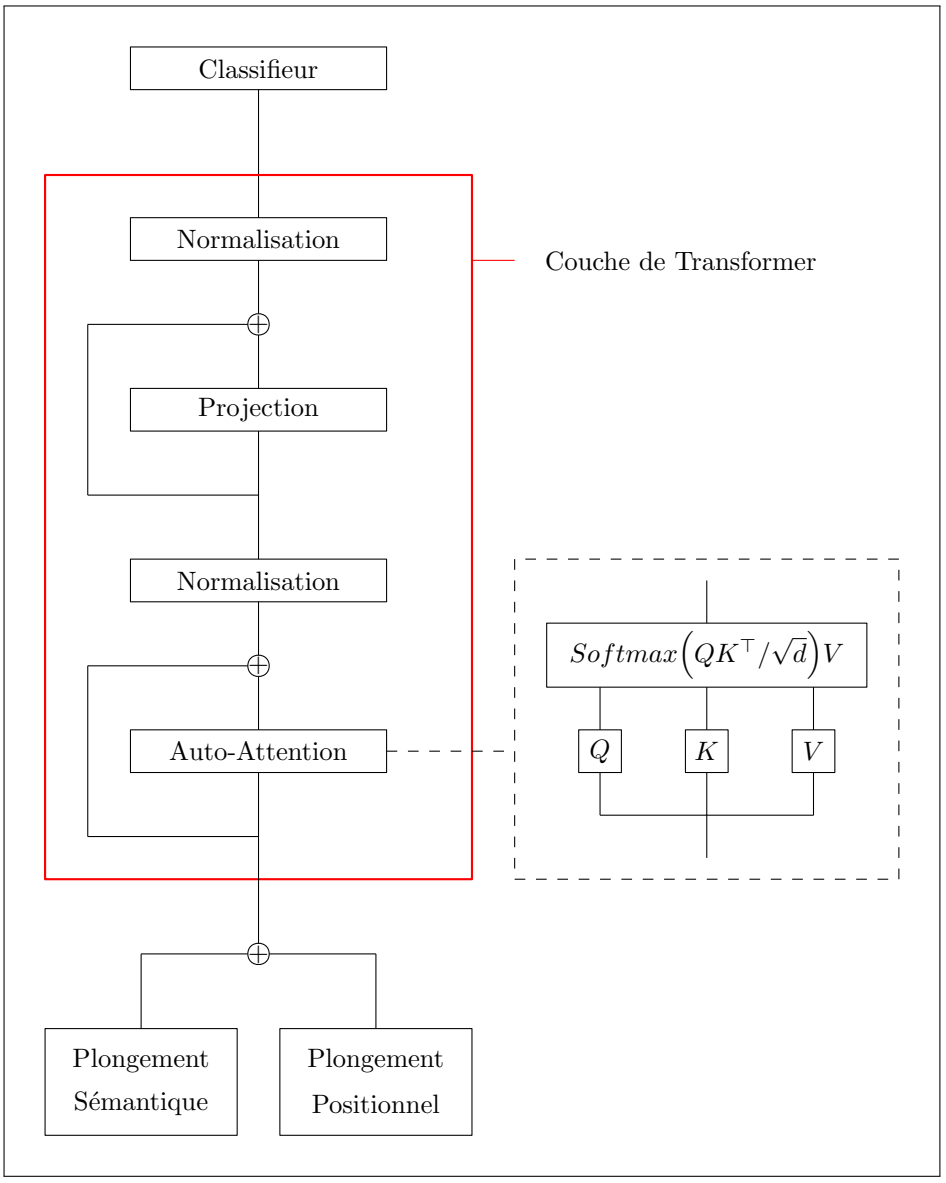


FIGURE 1.2 – Architecture des Transformers.

Le modèle BERT [Devlin et al., 2019] adapte l'architecture Transformer pour entraîner des embeddings contextualisés. Pour cela, une nouvelle tâche est créée consistant à masquer de façon aléatoire certains éléments des séquences (15%) en le remplaçant par un token générique prédéfini. Afin de complexifier la tâche, les auteurs ajoutent des permutations aléatoires et une seconde tâche de prédiction visant pour le modèle à déterminer si deux séquences se suivent. L'objectif

étant de pousser le modèle à chercher de l'information au-delà de la séquence traitée.

Grâce à l'accès à du matériel plus performant, aux capacités des Transformers à paralléliser les opérations et à la disponibilité de grandes quantités de données, BERT a largement influencé l'état de l'art contemporain en montrant des performances largement supérieures aux approches existantes sur de nombreuses tâches (classification, traduction) et dans plusieurs jeux de données de référence comme GLUE [Wang et al., 2019]. Ce modèle a été amélioré de nombreuses fois par la suite afin de couvrir certains défauts relatifs aux Transformers, notamment en ce qui concerne le traitement de séquences longues.

1.2 Contributions à l'informatique juridique

1.2.1 Contributions de référence

Peu de contributions traitent directement de problèmes de catégorisation de demandes abordés dans la suite de cette thèse. Les travaux se focalisent généralement sur des discriminations relatives au droit applicable ou sur la modélisation de thèmes. Ces problématiques peuvent être traitées de façon non supervisée par une LDA (Latent Dirichlet Allocation) [Blei et al., 2003] afin de regrouper les décisions partageant des thèmes proches [Remmits, 2017]. L'évaluation est effectuée par des juristes sur des décisions de la cour suprême américaine et pour laquelle une précision supérieure à 85% est observée. Certains auteurs se focalisent sur le droit applicable dans les décisions en comparant plusieurs approches dans des tâches multi-classes et multi-labels suivant différentes tailles de jeux de données, de 588 à 5599 exemples [Soh et al., 2019]. Les modèles testés sont à base de SVM et de matrices de fréquences mais aussi de plongements sémantiques tels que GloVe et BERT entraînés sur des corpus anglais généraux mais non spécialisés sur les textes juridiques. Les auteurs montrent que ces derniers sont d'autant plus performants comparativement aux autres approches que le nombre d'exemples est faible. Les performances sont quant à elles comprises entre des F-mesures de 55% à 65%, les SVM (noyaux

linéaires) ayant les scores les plus élevés sur des décisions issues de la cour suprême singapourienne. Une tâche similaire a été proposée sur les arrêts de cour de cassation pour laquelle la chambre de la cour doit être prédite selon une tâche de classification multi-classes [Sulea et al., 2017]. Les auteurs atteignent une F-mesure de 90% à l'aide d'un modèle SVM.

De nombreuses contributions traitent de la prédiction du sens du résultat. Une majorité d'entre elles se focalise sur l'utilisation de modèles linéaires, de SVM avec des matrices de fréquences et des n-grammes en entrées. Les performances maximales atteintes dans ces conditions varient en fonction de la langue étudiée : F-mesure de 90% pour des décisions de la cour suprême turque [Sert et al., 2021], entre 90% et 95% pour des décisions de justice brésiliennes [Bertalan and Ruiz, 2020] et 69% pour des décisions britanniques [Strickson and De La Iglesia, 2020]. D'autres modèles atteignent des F-mesures de 65% en exploitant des décisions issues de la cour suprême thaïlandaise à l'aide de GRU bi-directionnel et d'un mécanisme d'attention [Kowsrihawat et al., 2018].

La majeure partie de la littérature relative à la prédiction du sens du résultat concerne l'analyse de décisions de la Cour Européenne des Droits de l'Homme (CrEDH). Les premiers à s'y intéresser exploitent une représentation des séquences à base de n-grammes et utilisent un modèle SVM comme prédicteur [Aletras et al., 2016]. Les décisions traitées sont en anglais et se basent sur l'exploitation de faits extraits. Les auteurs parviennent à obtenir des précisions entre 75% et 80% selon l'article concerné. Cette expérimentation est reprise en étendant l'étude sur les autres articles de la convention. Des modèles similaires sont utilisés pour des précisions variant de 65% et 85% [Medvedeva et al., 2020]. Des comparaisons sont aussi effectuées entre des SVM, des régressions logistiques et des forêts aléatoires menant à des observations similaires aux contributions précédentes, les SVM étant toujours plus performants que les autres approches [Liu and Chen, 2017]. L'utilisation de modèles d'apprentissage profond à base de réseaux de neurones récurrents puis de BERT ont permis par la suite d'obtenir des gains de précision entre 5 et 10 points en fonction de l'article traité [Chalkidis et al., 2019a, Medvedeva et al., 2021]. A noter qu'il n'existe pas à notre connaissance de publication traitant les décisions françaises de la CrEDH.

Du fait des considérations éthiques et morales intrinsèquement liées au traitement de corpus juridiques, certaines contributions s'orientent vers des approches en mesure d'expliquer les prédictions. Certains auteurs observent que l'approche consistant à extraire les scores dans l'attention pour expliquer un prédicteur ne corrobore pas les décisions des experts [Branting et al., 2019, 2021]. Ils proposent pour cela des méthodes alternatives à base de clustering et d'extraction de termes clés à l'aide de critères de similarité sémantique entre ces derniers et d'autres définis préalablement. Une autre approche se base sur des modélisations *seq-to-seq* afin de générer les motifs d'une décision et donc une sortie interprétable [Ye et al., 2018]. Pour cela les auteurs utilisent les faits en entrée ainsi que les réquisitions de décisions pénales chinoises dans un modèle composé de couches récurrentes et d'attention. L'évaluation est effectuée par des juristes avec une échelle de notation prédéfinie qui consiste à noter la qualité des motifs générés et à évaluer à quel point ces derniers répondent aux faits rapportés. D'autres auteurs se basent sur un modèle à deux composants, l'un servant à extraire les éléments discriminants issus des faits, l'autre exploitant ces extractions pour prédire le sens du résultat [Chao et al., 2019]. Les architectures utilisées sont à base de GRU et d'attention, les performances quant à elles se situent autour de 85% de précision pour des affaires pénales chinoises. Des contributions se basent aussi sur des modèles d'interprétation existants [Górski and Ramakrishna, 2021] tels que LIME [Ribeiro et al., 2016], SHAP [Lundberg and Lee, 2017] ou Grad-CAM [Selvaraju et al., 2017] afin de comparer leur pertinence dans des tâches de dénomination juridique de phrases. Les auteurs concluent qu'il est difficile de déterminer qu'une méthode est meilleure qu'une autre, leur pertinence étant situationnelle. Le Chapitre 5 de cette thèse propose deux nouvelles méthodes d'interprétation adaptées au traitement de la langue.

1.2.2 Contributions récentes

Les performances observées avec l'utilisation des modèles de plongement sémantique puis de BERT sur des tâches générales de TALN ont poussé des auteurs à adapter ces approches à de nouvelles données pour résoudre des problématiques dans des domaines spécifiques.

Dans le cadre du traitement de textes juridiques, plusieurs Transformers spécialisés sur le langage juridique ont été entraînés suivant des quantités variables de données. Le modèle LEGAL-BERT est entraîné suivant différents paramétrages de tailles et de profondeurs en se basant sur 12 Go de documents juridiques anglais [Chalkidis et al., 2020]. Les auteurs montrent notamment que les performances finales du modèle sont en réalité largement conditionnées aux choix de l'hyperparamètrage lors des phases de fine-tuning, l'évaluation étant effectuée sur des tâches de classification sur des décisions de CrEDH. D'autres auteurs montrent qu'une adaptation par MLM [2] sur quelques documents seulement permet d'obtenir des gains de performances [Elwany et al., 2019, Zheng et al., 2021]. Cependant, ces gains dépendent fortement de la proximité de la distribution des données entre la tâche de MLM et celle pour laquelle le modèle est adapté par la suite.

Les contributions concernant des données en langue française sont plus rares. CriminelBART [Garneau et al., 2021] se focalise sur les modèles BART et BARThez [Lewis et al., 2019, Eddine et al., 2021], variantes de BERT orientées vers la génération. Le modèle est entraîné sur des corpus de décisions pénales canadiennes et testé sur des tâches de génération et de prédiction de charges pénales en masquant des passages spécifiques dans les documents. Concernant le droit français, seul JuriBERT est entraîné à notre connaisance sur des décisions de justice française [Douka et al., 2021]. Le corpus de 6.3Go est extrait de *Légifrance* et le modèle est évalué sur deux tâches de classification. L'une relative à la prédiction de la chambre et de la section de la cour de Cassation (84% de précision), l'autre consistant à classer les plaidoiries du demandeur selon 151 catégories préétablies (71% de précision). Les auteurs testent leur modèle dans plusieurs configurations de tailles et de profondeurs et concluent qu'une version réduite améliore les performances sur les tâches annexes. À noter qu'un modèle similaire est entraîné dans la suite de cette thèse sur un corpus juridique de 30Go.

Certaines contributions soulèvent la nécessité de recourir à des modèles capables de traiter des séquences longues. La raison vient du fait que la taille

2. *Masked Language Modeling*, consiste à remplacer aléatoirement des tokens puis d'entraîner un modèle à les prédire.

moyenne des documents juridiques surpasse largement la longueur de 512 éléments sur laquelle la majorité des modèles sont entraînés. Dans ces conditions contraintes, il est d'usage de recourir à des modélisations à base de réseaux récurrents. Des contributions proposent des alternatives en découpant les documents longs en sous-segments pour les traiter indépendamment [Wan et al., 2019]. Ces derniers sont ensuite fusionnés par le biais d'un LSTM pour résoudre des tâches de catégorisation de documents. D'autres auteurs [Bambroo and Awasthi, 2021] proposent de fusionner plusieurs architectures de Transformers telles que DistillBERT [Sanh et al., 2020] et Longformer [Beltagy et al., 2020] pour permettre le traitement de documents composés de plusieurs milliers de tokens. Ils observent notamment des gains de performances importants (+10% de précision) par rapport à l'utilisation d'un modèle BERT sur des tâches de classification de documents. Plusieurs modèles en mesure de traiter des séquences longues sont proposés dans cette thèse.

Chapitre 2

Analyse préliminaire : indexation et prédiction

La justice prédictive requiert la construction de jeux de données annotées et la modélisation d'un problème pour permettre l'entraînement de modèles. Dans une décision, le juge répond à des prétentions formulées par les parties en y associant, pour chacune d'entre elles, une règle de droit (un article, une norme), un résultat (acceptation ou rejet) et un quantum (une somme d'argent ou une peine). La prise en compte des faits et l'identification des demandes sont les premières étapes effectuées par le praticien pour comprendre le document qu'il consulte et souhaite exploiter. Or, cette recherche manuelle peut s'avérer chronophage en fonction de la nature du litige et de la façon dont les éléments sont exprimés. Bien que certains mots et expressions permettent généralement de caractériser les types de demandes rencontrés, ce seul critère peut s'avérer insuffisant lors de l'utilisation d'un moteur de recherche juridique favorisant la quantité des résultats retournés à leur pertinence. La construction de classes de prétentions, définies par leur objet et leur fondement, peut dès lors servir de point de départ pour l'automatisation de cette recherche en permettant de regrouper des décisions selon un critère de similarité.

Ce chapitre décrit dans un premier temps la structure interne des décisions de justice et la répartition de l'information exploitable afin de résoudre des problématiques de classifications liées à des types de prétentions spécifiques.

Dans un second temps sont présentées des méthodes permettant de filtrer les décisions afin de déterminer les catégories de demandes présentes et d'inférer leur résultat.

2.1 Format des décisions et données

Bien que la rédaction et la forme d'une décision de justice peuvent varier, la structure générale reste sensiblement la même d'une juridiction à l'autre, que ce soit dans un cadre local ou plus large (CJUE, CrEDH [1]).

2.1.1 Structure d'une décision

La rédaction d'une décision est effectuée par un greffier sous le contrôle d'un juge et doit respecter des règles principales dont certaines sont rassemblées dans le code de procédure civile (CPC).

2.1.1.1 Structure générale

Le jugement civil est traditionnellement composé de quatre parties présentées en Figure 2.1 dans lesquelles se retrouvent :

— l'*en-tête* dont les éléments la constituant sont prévus dans l'article 454 du CPC ;
— le *litige* permettant d'exposer les faits, les prétentions et les moyens des parties afin de délimiter la matière du procès ;
— les *motifs* permettant d'exposer les éléments de fait et de droit motivant la décision du juge ;
— le *dispositif* présentant les conclusions associées à chaque prétention.

L'*en-tête*, rédigée par le greffier, se compose principalement de méta-données permettant d'identifier la décision que ce soit par le biais du numéro de référence, de la juridiction, de la date, de la ville, du nom de l'appelant et de l'intimé

1. Cour de Justice de l'Union Européenne et Cour Européenne des Droits de l'Homme.

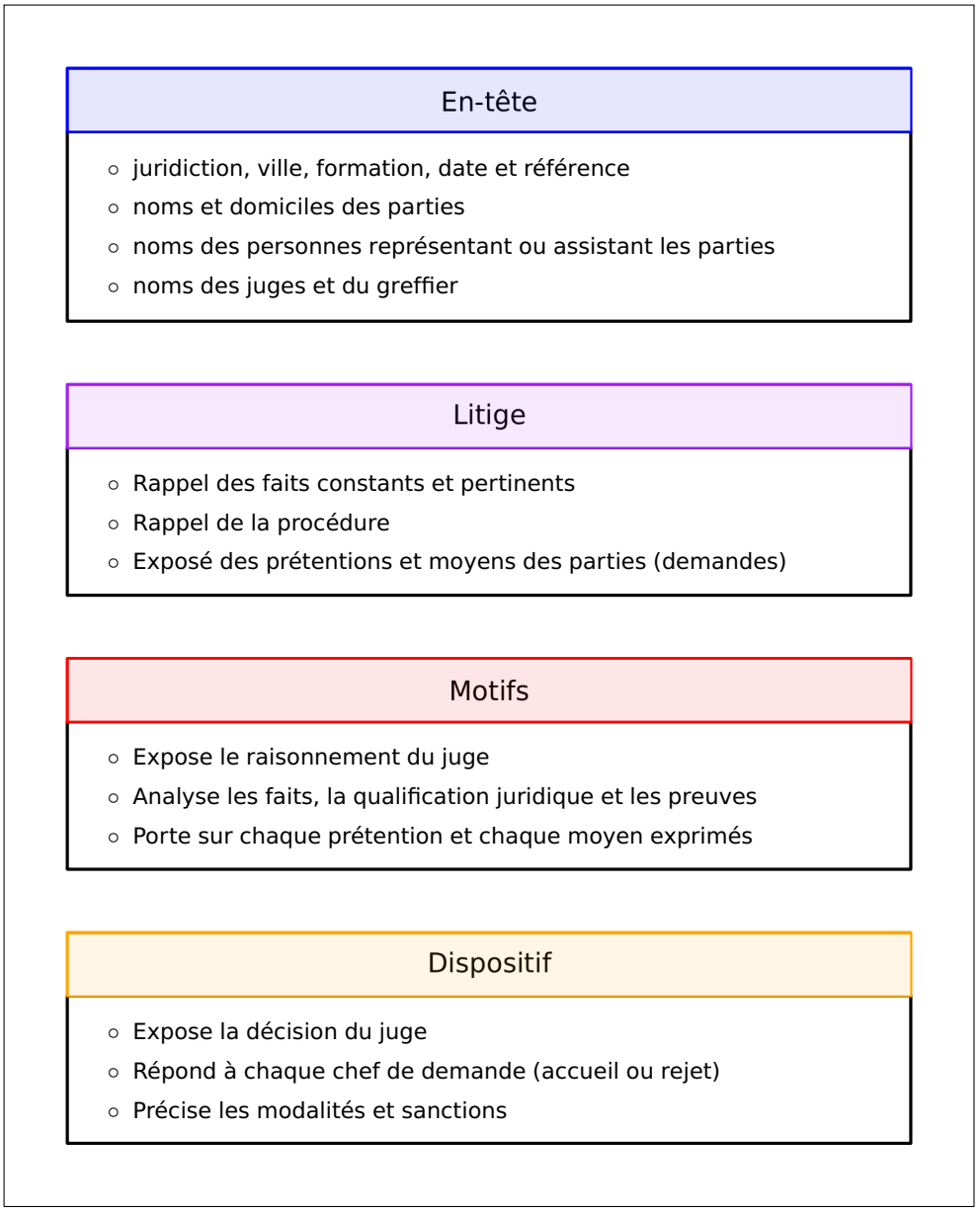


FIGURE 2.1 – Structure d'une décision.

(anonymisés), des avocats ou encore du greffier et du juge. En théorie, les différentes cours doivent respecter un format spécifique afin de simplifier la lecture et l'obtention de telles informations. En pratique, il existe une forte variabilité aussi bien entre les juridictions que géographiquement puisque les juges et greffiers tendent à suivre un modèle qui leur est propre. Ce processus d'extraction et d'anonymisation peut cependant s'automatiser en entraînant un modèle de

reconnaissance d'entités nommées [Tagny Ngompe, 2020] pouvant aussi être exploité dans la mise en place d'un processus d'anonymisation, étape obligatoire pour la publication des décisions.

L'exposé du *litige* s'apparente à un travail de synthèse des éléments afin de clarifier les positions et les prétentions. Le litige s'articule autour de trois composants : les faits, la procédure et les prétentions et moyens des parties. Le rappel des faits peut prendre différentes formes et varier en fonction de la nature du contentieux. On y distingue les faits constants, c'est-à-dire non constestés et permettant de soutenir les demandes des parties et les faits pertinents dont la connaissance est utile à la compréhension du résultat. Leur distinction n'étant généralement pas explicite, il est d'autant plus difficile de les extraire dans un but de prédiction. Le rappel de procédure permet de synthétiser et d'énumérer les étapes de la procédure. Il est généralement introduit par la formule « *Par acte d'huissier du [...]* » et suivi, pour les juridictions de second degré ou pour la cour de cassation d'un rappel des conclusions précédentes. Ainsi, dans le cadre d'une décision de cour d'appel sont énumérés les prétentions et les résultats associés de la première instance en suivant l'organisation juridictionnelle décrite en Figure 2.2.

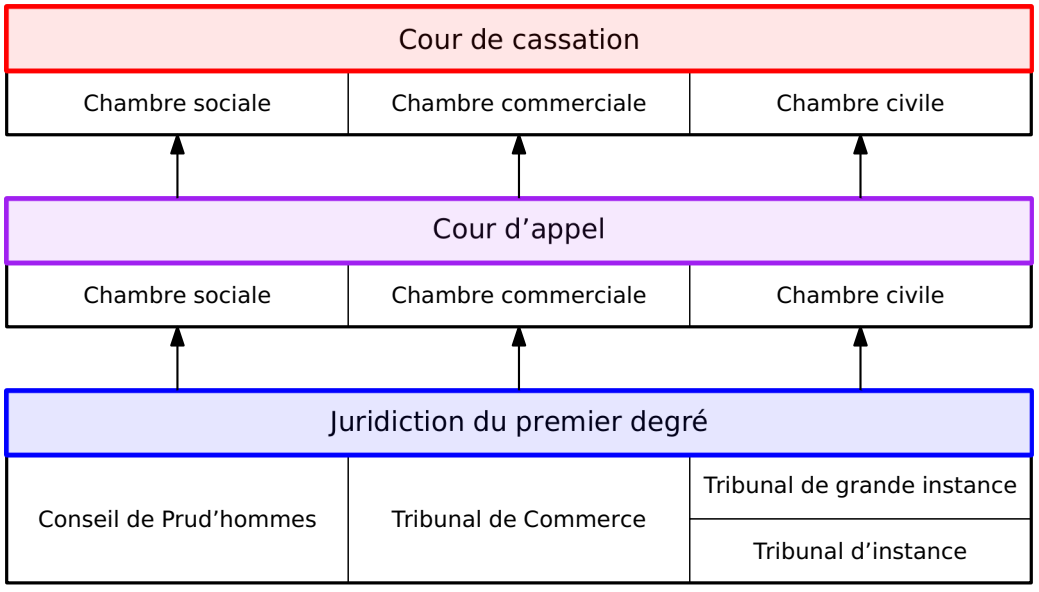


FIGURE 2.2 – Organisation juridictionnelle en matière civile.

Enfin le *litige* se conclut par l'exposé des prétentions et des moyens. Les parties fixent l'objet du litige en énonçant leurs demandes. Cet exposé permet

d'assurer une certaine clarté et d'éviter que le juge altère, déforme ou omette des éléments dans le reste de la décision.

Le juge a l'obligation de motiver sa décision afin de répondre à plusieurs finalités : assurer que les moyens et prétentions des parties sont traités et examinés, éviter une forme de partialité et permettre le contrôle par la cour de cassation de l'application correcte du droit. Les *motifs* permettent de suivre le raisonnement du juge, celui-ci va traiter tous les éléments exprimés par les parties dans leurs conclusions de façon pertinente, précise et catégorique afin de limiter les risques d'ambiguïté et de rendre intelligibles les motivations pour le justiciable. Les motifs sont généralement ordonnés pour faciliter la lecture, privilégiant dans un premier temps les prétentions principales puis celles subsidiaires des demandeurs et dans un second temps les prétentions des défendeurs. Le raisonnement du juge s'articule enfin en trois temps : il énonce une règle, vérifie les conditions d'application de celle-ci compte tenu de la situation et conclut en conséquence.

Souvent introduit par la formule « *Par ces motifs, [...]* », le *dispositif* donne la solution du litige. Par souci de clarté, celui-ci est généralement concis et impératif pour faciliter la compréhension et l'exécution. Le *dispositif* doit répondre à toutes les prétentions, sans exception, en conservant l'ordre d'examen de ces dernières dans les motifs. Il doit de surcroît être intelligible, c'est-à-dire ne pas contredire les motivations pour éliminer toute ambiguïté. Enfin, les résultats énoncés dans le *dispositif* sont exécutoires et donnent donc lieu à l'exécution. Une décision de cour d'appel est présentée en Figure 2.3.

RÉPUBLIQUE FRANCAISE AU NOM DU PEUPLE FRANCAIS
6ème Chambre A
ORDONNANCE No 283
R. G : 17/ 04448
Mme Céline X...
C/
M. Jean-Marie Michel Lucien Y...
Déclare l'acte de saisine caduc
Copie exécutoire délivrée le :
à :
RÉPUBLIQUE FRANÇAISE AU NOM DU PEUPLE FRANÇAIS
COUR D'APPEL DE RENNES ORDONNANCE DE MISE EN ETAT DU 04 DECEMBRE 2017
Le quatre Décembre deux mille dix sept, par mise à disposition au Greffe,
Monsieur Yves LE NOAN, Magistrat de la mise en état de la 6ème Chambre A, assisté de Xavier LE COLLEN, faisant fonction de Greffier,
Statuant dans la procédure opposant :
Madame Céline X... née le 29 Avril 1974 à SAINT RENAN (29290) ... Représentée par Me Françoise NAUDY-ORTAIS de la SCP KERDILES-KAYA & NAUDY-ORTAIS, Plaidant/ Postulant, avocat au barreau de BREST (bénéficie d'une aide juridictionnelle Totale numéro 2017/ 004995 du 09/ 06/ 2017 accordée par le bureau d'aide juridictionnelle de RENNES)
APPELANTE
à
Monsieur Jean-Marie Michel Lucien Y... né le 22 Avril 1964 à PALAISEAU (91120) ... Représenté par Me Luc BOURGES de la SELARL LUC BOURGES, Plaidant/ Postulant, avocat au barreau de RENNES
INTIME
A rendu l'ordonnance suivante :
Vu la demande d'observations sur la caducité de la déclaration d'appel adressée aux parties le 2 novembre 2017 ;
Vu l'absence d'observation des parties ;
Vu les dispositions des articles 908, 911 et 911-1 du code de procédure civile, dans leur rédaction, applicable au présent incident, antérieure au décret no 2017-891 du 6 mai 2017 ;
Selon l'article 908 du code de procédure civile, à peine de caducité de la déclaration d'appel, relevée d'office, l'appelant dispose d'un délai de trois mois à compter de la déclaration d'appel pour conclure ;
Selon l'article 911 du code de procédure civile, sous les sanctions prévues aux articles 908 à 910, les conclusions sont notifiées aux avocats des parties dans le délai de leur remise au greffe de la cour. Sous les mêmes sanctions, elles sont signifiées dans le mois suivant l'expiration de ce délai aux parties qui n'ont pas constitué avocat ; cependant, si, entre-temps, celles-ci ont constitué avocat avant la signification des conclusions, il est procédé par voie de notification à leur avocat ;
En l'espèce, la déclaration d'appel de madame Céline X...a été effectuée le 20 juin 2017. L'appelante a déposé ses conclusions au greffe le 7 septembre 2017, soit dans le délai prévu à l'article 908. A cette date, monsieur Jean-Marie Y..., intimé, n'avait pas constitué avocat, ce qu'il n'a fait que 19 septembre 2017. Il appartenait en conséquence à l'appelante, en application de l'article 911, de notifier ses conclusions à l'avocat de l'intimé dans le mois suivant l'expiration du délai prévu à l'article 908, soit au plus tard le 20 octobre 2017, à défaut d'avoir signifié ces conclusions à l'intimé avant sa constitution d'avocat ;
Si l'appelante a signifié sa déclaration d'appel à l'intimé le 28 septembre 2017, conformément à l'article 902, elle ne lui a en revanche pas signifié ses conclusions, ni notifié celles-ci à son conseil avant le 20 octobre 2017 ;
PAR CES MOTIFS
Prononce la caducité de la déclaration d'appel,
Condamne l'appelante aux dépens.
Le Greffier, Le Conseiller de la mise en état,

FIGURE 2.3 – Cour d'appel de Rennes, 4 décembre 2017, 17/04448.

2.1.1.2 Similarité et prétentions

Lorsqu'un juriste tente d'anticiper l'issue d'un procès, il va dans son travail rechercher des affaires similaires afin d'évaluer les chances pour un client d'obtenir un verdict qui lui est favorable. Cette tâche de recherche peut s'effectuer à plusieurs niveaux : en recherchant une situation où les faits sont semblables, en sélectionnant des décisions invoquant un ou plusieurs articles jugés essentiels, en privilégiant des prétentions des parties. Le principe même de similarité peut être perçu différemment par les juristes.

En traitement automatisé de la langue, la capacité de regrouper des textes similaires par le biais de métriques spécifiques est essentielle puisqu'elle permet de réduire significativement une phase d'annotation manuelle parfois coûteuse ou impossible sur de grands volumes. En pratique, les approches traditionnelles de calcul de similarité fonctionnent mal dans le cadre de documents juridiques pour différentes raisons. La première concerne la taille des textes pouvant sensiblement varier et rendant la plupart des métriques très approximatives. La seconde concerne la forme et le vocabulaire employé, des décisions relatives à des troubles de voisinage peuvent être similaires pour un juriste alors qu'elles traitent de faits qui n'ont que peu de ressemblances. Dans ces conditions, les métriques telles que le cosinus, la *Word Mover's Distance* (WMD) [Kusner et al., 2015] ou la distance de Jaccard ont un intérêt limité, plus particulièrement si les documents sont longs et complexes [Moodley et al., 2019]. L'usage de multiples critères comme des ontologies ou des graphes est en pratique plus cohérent et performant [Wagh and Anand, 2020].

Dans les pays où le *Common Law* est la norme, le principe de similarité est au cœur du système juridique puisque la jurisprudence est la principale source de droit. Ainsi un jugement se réfère systématiquement à un autre antérieur et similaire si l'affaire ou la situation n'est pas inédite. Il est donc possible en théorie, de construire un graphe modélisant toutes les décisions et leur proximité. Dans le système français, le droit est écrit dans des codes, il est par conséquent bien plus difficile de relier des affaires similaires entre elles.

Lorsque des parties s'engagent dans une procédure judiciaire, ces dernières constituent des demandes avec pour objet une prétention qu'elles soumettent

au magistrat. Dans une procédure écrite, le juge va traiter chaque prétention en y répondant dans ses conclusions (accueil ou rejet). Puisque ces dernières sont des affirmations de fait et de droit, elles peuvent être considérées comme des éléments clés et discriminants et ainsi permettre de regrouper les décisions de justice dans des catégories prédéfinies par des juristes. Dès lors, la tâche peut être ramenée à un problème de classification si des groupes de documents similaires sont créés. Cela suppose cependant la constitution d'un grand nombre de classes définies manuellement et cette liste peut en théorie évoluer dans le temps. Le droit évoluant en permanence, les juges peuvent se retrouver face à des affaires et des problèmes juridiques inédits.

2.1.2 Contexte et collecte des données

La construction de bases dans le but de résoudre des tâches de détection du type de demande et du sens du résultat est une étape particulièrement chronophage qui nécessite notamment de sélectionner des ensembles de documents similaires et de les étiqueter par la suite.

2.1.2.1 Contexte de la tâche

Dans la pratique, les demandes sont nommées selon leur objet. Ainsi lorsque l'on se réfère à la réparation d'un dommage ou d'une prestation, l'on parle de demande de compensation. Une catégorie de demandes regroupe un sous-ensemble de prétentions partageant à la fois leur objet et leur fondement. Il est par exemple possible d'espérer le paiement de dommages et intérêts au titre d'une norme spécifique, et ainsi créer une catégorie ayant ces deux caractéristiques. Si celle-ci est correctement définie et qu'il est possible de regrouper un nombre suffisant de décisions dans lesquelles ce type de demande apparaît, alors une tâche de classification peut être construite en parallèle afin d'en automatiser la détection et l'indexation.

En pratique, les juristes savent associer par expérience un certain nombre de termes à une catégorie sans avoir au préalable strictement défini celle-ci. Ils vont par exemple exploiter, dans un moteur de recherche, des mots clés ayant une

forte probabilité d'apparition dans la décision puis vont filtrer manuellement les résultats afin de ne conserver seulement les documents jugés intéressants. En pratique cette méthode est sous-efficiente et approximative. Il parait plus naturel de filtrer directement par type de prétention sans au préalable connaître tous les termes et expressions clés.

2.1.2.2 Construction des bases

Pour résoudre cette tâche, des juristes construisent des bases composées de décisions semblables selon des critères juridiques prédéfinis afin de constituer des catégories de demande. Si ces sélections sont suffisamment représentatives, un ensemble précis de mots-clés peut être déterminé pour filtrer automatiquement chaque ensemble, en procédant par des comparaisons de fréquences.

Neuf bases sont construites selon le paiement de dommages et intérêts au titre : d'une action en résolution, d'une action estimatoire, d'une action rédhibitoire, de la responsabilité de l'avocat, de l'article 1384 du code civil, de l'article 1792 du code civil, de troubles de voisinage, d'une exception d'inexécution et d'un licenciement abusif. Les tailles, les résultats et les longueurs moyennes des décisions sont présentés en Table 2.1.

	Taille	Accueil (+)	Rejet (−)	Longueur
Act. en résolution	100	50	50	2598
Act. estimatoire	92	46	46	2980
Act. rédhibitoire	98	49	49	2754
Resp. avocat	144	61	83	3795
Dm. intérêts (1384)	98	48	50	3898
Dm. intérêts (1792)	68	34	34	4909
Trouble de voisinage	45	23	22	2831
Except. d'inexécut.	106	48	58	2172
Licenciement	472	229	243	4022

TABLE 2.1 – Caractéristiques des bases.

Le résultat (accueil ou rejet) correspond uniquement à celui de la demande analysée, les autres résultats ne sont pas exploités [2]. Compte tenu du nombre de tokens des documents (séparateur espace), il est nécessaire de filtrer l'information. Pour cela, nous proposons une approche dans laquelle chaque décision est découpée en phrases, auxquelles on associe un score construit selon la présence de termes discriminants. Nous détaillerons dans la section suivante la procédure d'extraction puis celle du calcul des scores.

2.2 Extraction des demandes

L'extraction automatique des demandes et de leurs résultats ouvre l'accès à des informations d'importance permettant de regrouper des décisions similaires. Pour cela, il est nécessaire de connaître des caractéristiques discriminantes pour classer les décisions par type de prétention. Celles-ci permettant d'inférer les zones d'importance dans le document pour extraire les éléments permettant la prédiction du résultat.

2.2.1 Eléments discriminants et sélection

La résolution de la tâche de classification par type de demande nécessite la construction de bases annotées et la sélection d'un ensemble de métriques permettant de discriminer les termes jugés importants.

2.2.1.1 Extraction du vocabulaire discriminant

Puisque des bases relatives à des catégories de demandes ont été créées et que des corpus de décisions comme CAPP sont accessibles, il est possible de définir des métriques permettant de filtrer le vocabulaire discriminant propre à ces documents. Soit $\mathcal{C}$ l'ensemble des catégories et $\mathcal{W}$ le vocabulaire, un terme $w \in \mathcal{W}$ peut être considéré comme discriminant si sa fréquence $f_w^c \in [0, 1]$ dans les documents de la catégorie $c \in \mathcal{C}$ excède fortement sa fréquence dans l'ensemble

2. Plusieurs demandes et donc résultats peuvent être exprimés dans une décision.

des autres catégories $\bar{c}$. Sont sélectionnées plusieurs métriques reportées dans la Table 2.2 pour discriminer le vocabulaire.

Métrique	Formule
Différence	$f_w^c - f_w^{\bar{c}}$
Entropie croisée	$-f_w^c \log f_w^{\bar{c}}$
Entropie croisée à seuil*	$-H_\alpha(f_w^c) \log f_w^{\bar{c}}$
Divergence KL	$f_w^c \log \frac{f_w^c}{f_w^{\bar{c}}}$
Co-occurrence à seuil**	$\frac{1}{\lvert\mathcal{W}\rvert} \sum_j y_{w,j}^s$

* Seuil $\alpha \in [0,1]$, $\quad H_\alpha(x) = 1$ si $x > \alpha$ sinon 0

** $y_{i,j}^s = H_0(x_{i,j} - q_s(x_{.,j}))$, $\quad q_s$ quantile s du vecteur

$x_{i,j}^c \in [0,1]$, co-occurrence des termes $i, j \in \mathcal{W}$ dans c, $\quad x_{i,j} = x_{i,j}^c - x_{i,j}^{\bar{c}}$

TABLE 2.2 – Métriques de sélection.

Les métriques à base de seuils sont testées avec des valeurs comprises dans $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ afin de déterminer si le seuil impacte la sélection des termes. Le choix du vocabulaire discriminant a pour but d'associer à chaque phrase dans une décision, un score permettant de déterminer si celle-ci est dans le thème du type de demande analysée. Puisque le juge tend à traiter l'information de façon assez structurée, connaître la localisation de ces éléments peut permettre l'extraction et l'inférence du résultat associé. Seuls les termes représentant la catégorie ont un intérêt dans l'élaboration des scores, les métriques sont calculées sans exploiter les fréquences complémentaires, notamment pour l'entropie et la divergence KL qui nécessitent normalement un calcul d'espérance. Ainsi, des termes $w, w' \in \mathcal{W}$ auront des scores différents même si $f_w^c = f_{w'}^{\bar{c}}$. Les termes discriminants sont reportés en Figure 2.4.

En pratique, les métriques donnent des listes de mots très similaires mais l'ordre peut légèrement changer. Un autre effet observé est relatif à l'amplitude des scores. Puisqu'ils sont normalisés et réutilisés dans la phase de vectorisation des séquences, des différences peuvent être observées sur la qualité des prédictions dans les expérimentations menées par la suite.

Action en résolution	vendeur, vices, cachés, conformité, 1604, prix, acquéreur, vendeur, restituer, véhicule, manquement, litigieux, expertise, garantie, usage
Action estimatoire	estimatoire, vices, cachés, 1641, réduction, vendeur, garantie, impropre, usage, acquéreur, expertise, connaissance, désordres, 1644, travaux, réparation
Action rédhibitoire	vices, cachés, 1641, vendeur, impropre, restitution, usage, acheteur, expertise, acquéreur, vendue, véhicule, moteur, destination, défauts, rédhibitoire
Resp. avocat	chance, perte, responsabilité, manquement, faute, réparation, professionnelle, commis, client, assigner, honoraires, mission, mandat, indemnisation, preuve, préjudices
Dm. et intérêts (1384)	endurées, souffrances, 1384, accident, temporaire, dommage, victime, responsable, esthétique, expertise, responsabilité, préjudices, indemnisation, chute, cpam, assureur
Dm. et intérêts (1792)	impropre, destination, constructeurs, solidité, affectant, assureur, construction, réception, contractuelle, reprise, solidum, devis, désordre, garantir, réparation
Trouble de voisinage	voisin, travaux, troubles, habitation, propriété, huissier, propriétaire, astreinte, réparation, inconvénients, maison, constat, terrain, anormaux, construction
Exception d'inexécution	travaux, obligations, loyers, clause, paiement, résiliation, bailleur, constat, désordre, facture, huissier, demeure, exécution, délivrance, résolutoire
Licenciement	sérieuse, réelle, salarié, préavis, congés, salaire, rupture, payés, emploi, afférents, indéterminée, compensatrice, ancienneté, poste, durée, indemnité, rémunération

FIGURE 2.4 – Termes ayant les scores moyens les plus élevés (différence).

2.2.1.2 Sélection des segments exploitables

Déterminer le sens du résultat relatif à une demande spécifique suppose de connaître, au moins approximativement, les zones de la décision contenant l'information recherchée. Généralement, le juge fait référence à la demande dans les trois parties principales : le litige, les motifs et le dispositif. Cependant,

ces références peuvent être plus ou moins exploitables. Le rappel des faits est notoirement complexe à traiter puisque les faits peuvent revêtir des aspects très hétérogènes, que ce soit par le vocabulaire utilisé ou dans la forme plus ou moins détaillée de l'exposé. Ainsi, un trouble de voisinage peut être compté d'une infinité de façons et dépendre de situations parfois inédites. Lorsque le juge motive sa décision, certains traits peuvent être communs (article invoqué) entre des décisions munies de demandes similaires. L'exploitation du dispositif parait suffire pour inférer le résultat mais nécessite en réalité une contextualisation puisque la façon d'exposer le résultat peut largement varier en fonction des situations. Certaines expressions sont difficilement exploitables car elles font une référence implicite, d'autres répondent à la mauvaise prétention.

La tâche de sélection de passages peut être abordée de différentes façons : soit en inférant un résultat sur la décision brute qui peut être de grande taille, soit en sélectionnant automatiquement des segments dans lesquels l'information recherchée a le plus de chance d'être présente. La résolution de la tâche présente plusieurs contraintes. Premièrement, le nombre d'exemples pour chaque catégorie est très limité (quelques dizaines ou quelques centaines), rendant la tâche difficile et limitant l'usage d'un nombre élevé de paramètres. Deuxièmement, l'information est distribuée dans les trois parties de la décision. Dès lors sont testées dans les expérimentations suivantes, des approches où l'extraction se fait sur le document complet puis dans une variante où le litige, les motifs et le dispositif sont traités indépendamment puis concaténés pour obtenir une sélection des phrases. En pratique cela nécessite d'utiliser un modèle de sectionnement capable de segmenter la décision puisque les parties ne sont pas connues *a priori*. Les expérimentations se basent sur un modèle pré-entrainé de découpe de documents juridiques spécialisé dans cette tâche [Tagny Ngompé et al., 2017].

La sélection se fait dans un premier temps en associant un score à chaque phrase en fonction des métriques présentées en Table 2.2 puis en préservant un sous-ensemble de segments ayant les valeurs les plus élevées. Deux critères sont utilisés, un premier conservant uniquement les éléments ayant un score supérieur à la moyenne (15-25% du document) et un autre à seuil fixe sélectionnant

uniquement 10% des passages. À noter que l'utilisation de seuils élevés, supérieurs à 30%, génère des pertes de performances significatives du fait de l'ajout d'éléments inutiles à la prévision du résultat de la demande traitée.

L'autre critère important est le calcul de la pondération par segment. Puisque seuls les mots issus de la liste des termes discriminants ont un score non nul, il existe diverses façons d'associer un poids à une phrase. L'utilisation de la moyenne par exemple peut avantager des segments courts par rapport à des segments plus longs alors que les deux possèdent un même nombre d'éléments jugés discriminants. En notant t la longueur de la séquence, nous proposons d'étudier :

- une moyenne des termes (division par t) ;
- une moyenne favorisant les séquences plus longues (division par $\sqrt{t}$) ;
- une moyenne des scores non nuls ;
- le score maximal dans la séquence.

La Figure 2.5 donne un exemple d'extraction des phrases ayant un score supérieur à la moyenne.

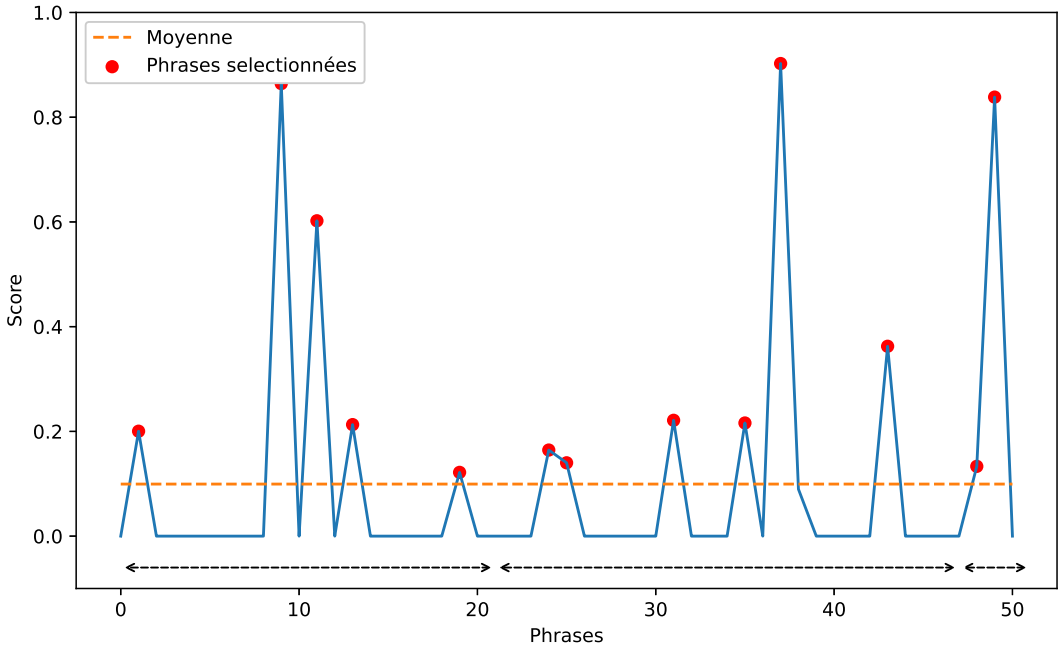


FIGURE 2.5 – Extraction des phrases importantes dans une décision.

En pratique, les segments ont généralement des scores très hétérogènes avec des pics marqués à certains endroits. La grande majorité des phrases ont un

score nul du fait de l'absence de mots relatifs à la demande. Les chances qu'elles informent sur le sens du résultat sont donc très faibles. À noter que les décisions sont composées de transitions, en particulier dans l'annonce de l'une des trois parties (« *Par ces motifs [...]* ») ou avant une énumération, ce qui rend la représentation parfois erratique.

À titre d'exemple, la Figure 2.6 expose les phrases maximisant le score dans chaque partie de la décision sur le thème d'un trouble de voisinage.

Litige	Par assignation en date du 20 février 2012, monsieur et madame M. ont fait assigner leurs voisins devant le tribunal de grande instance de Périgueux afin que le tribunal les condamne principalement à faire cesser le trouble anormal de voisinage subi par la présence de batraciens introduits dans une mare créée au pied de leur immeuble, ordonne d'une part la construction d'un mur pour avoir coupé une haie mitoyenne et d'autre part l'arrachage de bambous surplombant leurs panneaux photovoltaïques.
Motif	Il est donc établi en tout état de cause que les époux P. ont créé une mare sur leur propriété à moins de 10 m de la maison des époux M. alors qu'ils sont propriétaires d'un terrain d'une superficie de plus de 15 hectares.
Dispositif	Ordonne aux époux P. de combler leur mare située à moins de 10 mètres de l'habitation M. sous un délai de 4 mois après le prononcé du présent arrêt, ce sous astreinte provisoire de 150 euros par jour de retard et pendant un délai de 2 mois.

FIGURE 2.6 – Extraction de la phrase la plus fortement pondérée en fonction de la partie (troubles de voisinage, CA Bordeaux 14/02584).

Les phrases peuvent être particulièrement longues dans certaines conditions puisque le juge tend à énoncer tous les éléments les uns à la suite des autres. Cela entraîne, dans les motifs notamment, des segments parfois composés d'une centaine de mots.

2.2.2 Résultats

Cette section reporte les résultats des tâches de détection des catégories de demandes puis ceux relatifs à la détermination du sens du résultat en fonction de zones extraites de la décision par le biais d'un sectionnement et de l'exploitation des métriques présentées.

2.2.2.1 Les catégories de demandes

La classification des catégories de demandes a des contraintes. Premièrement, le nombre de classes peut varier avec le temps [3] puisqu'en pratique de nouvelles catégories peuvent être ajoutées un ré-entrainement est alors nécessaire pour mettre à jour le modèle. Deuxièmement, une décision de justice peut contenir plusieurs demandes de thèmes différents et donc la tâche peut être vue comme un problème multi-classes et multi-labels. En pratique, il est plus intéressant et plus simple d'entraîner indépendamment les catégories une à une. Puisqu'un accès à plusieurs milliers de décisions aux thèmes très variables existe, la construction de jeux composés de documents possédant la catégorie recherchée et d'autres ne la possédant pas est possible.

Pour chaque classe de demande, une base déséquilibrée de 1.000 décisions est créée dans laquelle sont présentes les décisions annotées par les juristes complétées par des décisions hors du thème traité (*One vs All*). Le pré-traitement est limité à la suppression de la ponctuation, à la suppression de la casse, à la substitution des sommes d'argent par un token spécial et à la tokenisation par le séparateur espace. Puisqu'une liste de termes discriminants est connue grâce à l'approche définie précédemment, cette dernière est fusionnée au vocabulaire extrait de la base composée de 1.000 décisions pour chaque catégorie de demande. La représentation des séquences est obtenue par le biais d'une matrice d'occurrences binaire.

Une validation croisée est effectuée en 10 étapes et les performances de deux classifieurs simples sont comparées selon la précision, le rappel et la F-mesure moyennés. Ces dernières sont reportées dans la Table 2.3.

Les résultats montrent que la détection des catégories prises indépendamment est triviale et qu'un simple modèle linéaire comme la régression logistique, basé sur des unigrammes et une représentation binaire, est en mesure de résoudre presque parfaitement la tâche dans la majorité des cas. En pratique, un juriste est en capacité de déterminer une classe de prétentions en détectant certaines expressions et certaines lois associées. Ces éléments se retrouvent notamment dans la liste des termes précédemment extraits (Figure 2.4), par

3. A terme, plusieurs centaines de catégories peuvent être créées.

	Logistique			Forêt aléatoire		
	P	R	F	P	R	F
Act. en résolution	1.000	1.000	1.000	1.000	1.000	1.000
Act. estimatoire	1.000	1.000	1.000	1.000	1.000	1.000
Act. rédhibitoire	1.000	1.000	1.000	0.992	0.994	0.993
Resp. avocat	0.998	0.986	0.991	0.996	0.980	0.988
Dm. intérêts (1384)	1.000	1.000	1.000	1.000	1.000	1.000
Dm. intérêts (1792)	1.000	1.000	1.000	1.000	1.000	1.000
Trouble de voisinage	1.000	1.000	1.000	1.000	1.000	1.000
Except. d'inexécut.	0.999	0.993	0.996	0.998	0.982	0.990
Licenciement	0.999	0.999	0.999	0.997	0.997	0.997

TABLE 2.3 – Performances de classification en fonction de la demande.

exemple l'expression « vices cachés » est systématiquement présente dans les décisions relatives aux trois premières catégories, celle-ci étant couplée à un article (1604, 1641 ou 1644 du code civil). À noter que l'entraînement d'un modèle sur une version multi-classes du problème, où les 9 types de demandes sont considérés, aboutit à des résultats légèrement inférieurs (F-mesure de 0.97) et nécessite en cas d'ajout d'une nouvelle catégorie un nouvel entraînement.

2.2.2.2 Détermination du résultat

La tâche de prédiction du résultat prend en compte un ensemble de critères pouvant fortement jouer sur les performances. Le premier critère est relatif au schéma de pondération et sélection du vocabulaire discriminant (Table 2.2), le second concerne le calcul du score par phrase et de la prise en compte de la longueur des segments, le troisième concerne le nombre de séquences extraites qu'il soit en fonction de la moyenne des scores ou en sélectionnant les 10% les plus élevés. Enfin le dernier critère dépend de la partie extraite soit par un traitement de la décision entière, soit en découpant le document pour traiter indépendamment les différentes parties et les concaténer ensuite.

La représentation des séquences se base sur des matrices de fréquences et par le biais d'un modèle FastText [Bojanowski et al., 2017] entrainé sur un corpus extrait de Légifrance regroupant des décisions de cour d'appel, de cour de cassation et du conseil d'état (soit 1 million de décisions). Afin d'obtenir une représentation des phrases dans une matrice de taille fixe, les mots sont pondérés en fonction de leur score obtenu par les métriques de sélection de vocabulaire (Table 2.2) après normalisation par une fonction *softmax*. Pour une phrase composée de t mots dont le mot i est représenté par un couple score-embedding ($s_i \in \mathbb{R}$, $\boldsymbol{r}_i \in \mathbb{R}^d$), sa représentation $\tilde{\boldsymbol{r}} \in \mathbb{R}^d$ est donnée par :

$$\tilde{\boldsymbol{r}} = \frac{1}{\sum_{j=1}^{j=t} e^{s_j}} \sum_{i=1}^{i=t} e^{s_i} \boldsymbol{r}_i$$

Pour chaque décision, seule l'étiquette d'accueil ou de rejet de la demande est connue, les résultats des demandes hors thème ne sont pas prédits. La tâche est traitée en plusieurs temps, d'abord des scores sont calculés par phrase, une segmentation de la décision est effectuée ensuite, puis les segments sélectionnés sont vectorisés et utilisés comme entrées dans un réseau de neurones simple à deux couches et activation Elu [Clevert et al., 2016a]. La taille du réseau de neurones est limitée afin d'éviter un sur-apprentissage compte tenu du faible nombre d'exemples et l'embedding est de dimension $d = 128$. Chaque modèle est testé suivant une stratégie de validation croisée à 10 étapes avec comme mesures de performance la précision, le rappel et la F-mesure. Le nombre total de modèles à tester est grand pour chaque catégorie de demande puisqu'il est nécessaire de définir 4 critères à chaque fois : la métrique de pondération des mots, la métrique de pondération des phrases, la méthode d'extraction et les parties à considérer. Les meilleurs performances sont reportées dans la Table 2.4.

Le choix des paramètres dépend de la demande traitée. En pratique, sélectionner 10% des phrases de la décision a un effet significatif sur les performances pour les demandes relatives à l'article 1792 du code civil et aux exceptions d'inexécution qui se retrouvent respectivement dans les décisions les plus longues et les plus courtes (Table 2.1). Le choix de la pondération des phrases montre que favoriser les séquences longues aux séquences courtes en divisant par $\sqrt{t}$ a peu d'effet sur les performances puisque les modèles tendent à sélectionner des

phrases similaires dans la majorité des cas. En pratique, seules les demandes sur l'article 1384 sont exploitables en réalité avec des F-mesures supérieures à 95%. Les autres types de prétention nécessitent le recours à des techniques plus avancées afin de les intégrer dans le cadre d'un entraînement sur une faible volumétrie et de désambiguïser des séquences complexes à interpréter. Les critères optimaux pour chaque catégorie sont présentés dans la Table 2.5.

En moyenne, la segmentation puis l'extraction de segments issus des motifs et du dispositif permettent de maximiser les performances tandis que l'utilisation du litige reste assez anecdotique pour des modèles peu sophistiqués et incapables d'exploiter pleinement des séquences nécessitant un raisonnement complexe. Le choix de la métrique et de la pondération des phrases a en réalité un effet limité comparé à la sélection des zones adéquates. Le fait de filtrer préalablement la décision a en effet des conséquences importantes sur le niveau de performances des modèles. La Table 2.6 permet de mesurer celles-ci.

	Poids	> Moyenne			Top 10%		
		P	R	F	P	R	F
Act. en résolution	t	0.870	0.852	0.850	0.852	0.829	0.820
	$\sqrt{t}$	0.834	0.839	0.823	0.8342	0.844	0.825
	$\tilde{t}$	0.866	0.872	**0.862**	0.814	0.812	0.796
	max	0.851	0.849	0.840	0.859	0.838	0.836
Act. estimatoire	t	0.779	0.783	0.750	0.741	0.730	0.693
	$\sqrt{t}$	0.803	0.790	**0.759**	0.771	0.774	0.720
	$\tilde{t}$	0.785	0.781	0.741	0.769	0.779	0.728
	max	0.781	0.771	0.750	0.786	0.794	0.746
Act. rédhibitoire	t	0.843	0.829	0.822	0.865	0.858	**0.845**
	$\sqrt{t}$	0.839	0.844	0.832	0.872	0.839	0.835
	$\tilde{t}$	0.846	0.839	0.838	0.851	0.828	0.820
	max	0.843	0.845	0.832	0.868	0.849	0.832
Resp. avocat	t	0.742	0.725	0.722	0.766	0.763	**0.745**
	$\sqrt{t}$	0.735	0.723	0.722	0.737	0.729	0.723
	$\tilde{t}$	0.746	0.719	0.720	0.746	0.737	0.732
	max	0.725	0.714	0.711	0.739	0.721	0.718
Dm. intérêts (1384)	t	0.932	0.944	0.930	0.951	0.967	0.954
	$\sqrt{t}$	0.941	0.961	0.943	0.948	0.969	0.953
	$\tilde{t}$	0.957	0.966	**0.957**	0.928	0.951	0.932
	max	0.948	0.967	0.953	0.942	0.959	0.942
Dm. intérêts (1792)	t	0.749	0.733	0.705	0.810	0.766	0.749
	$\sqrt{t}$	0.713	0.713	0.691	0.773	0.780	0.752
	$\tilde{t}$	0.766	0.758	0.725	0.769	0.759	0.724
	max	0.748	0.774	0.709	0.796	0.814	**0.773**
Trouble de voisinage	t	0.867	0.858	0.814	0.838	0.846	0.822
	$\sqrt{t}$	0.821	0.833	0.785	0.829	0.842	0.801
	$\tilde{t}$	0.879	0.879	0.832	0.871	0.850	0.822
	max	0.871	0.883	**0.855**	0.825	0.817	0.806
Except. d'inexécut.	t	0.677	0.668	0.638	0.678	0.677	0.666
	$\sqrt{t}$	0.667	0.665	0.631	0.684	0.705	0.676
	$\tilde{t}$	0.695	0.679	0.631	0.691	0.702	0.664
	max	0.676	0.690	0.648	0.722	0.737	**0.713**
Licenciement	t	0.793	0.797	0.791	0.768	0.769	0.762
	$\sqrt{t}$	0.789	0.792	0.788	0.768	0.771	0.764
	$\tilde{t}$	0.799	0.804	**0.798**	0.774	0.776	0.770
	max	0.800	0.807	0.797	0.767	0.770	0.763

TABLE 2.4 – Performances en fonction des demandes et des pondérations.

	Vecteur*	Zone**	Métrique	Poids
Act. en résolution	TF + FT	M, D	Co-oc $q_{80\%}$	$\tilde{t}$
Act. estimatoire	TF + FT	M, D	Entropie	$\sqrt{t}$
Act. rédhibitoire	TF + FT	L, M, D	Différence	t
Resp. avocat	FT	M, D	Co-oc $q_{80\%}$	t
Dm. intérêts (1384)	TF + FT	M, D	Co-oc $q_{80\%}$	$\sqrt{t}$
Dm. intérêts (1792)	TF + FT	M, D	Entropie	max
Trouble de voisinage	FT	L, M, D	Co-oc $q_{80\%}$	max
Except. d'inexécut.	FT	L, M	Co-oc $q_{80\%}$	max
Licenciement	TF + FT	M, D	Différence	$\tilde{t}$
Top perf. moyenne	TF + FT	M, D	Entropie	max

* TF = matrice de fréquences, FT = FastText

** L = litige, M = motifs, D = dispositif, C = document complet

TABLE 2.5 – Critères les plus performants par classe de prétention.

	Décision complète			Décision filtrée		
	P	R	F	P	R	F
Act. en résolution	0.695	0.695	0.678	0.866	0.872	0.862
Act. estimatoire	0.553	0.569	0.545	0.803	0.790	0.759
Act. rédhibitoire	0.809	0.800	0.784	0.865	0.858	0.845
Resp. avocat	0.631	0.638	0.620	0.766	0.763	0.745
Dm. intérêts (1384)	0.921	0.935	0.920	0.957	0.966	0.957
Dm. intérêts (1792)	0.644	0.649	0.589	0.796	0.814	0.773
Trouble de voisinage	0.746	0.754	0.701	0.871	0.883	0.855
Except. d'inexécut.	0.567	0.568	0.509	0.722	0.737	0.713
Licenciement	0.748	0.748	0.745	0.799	0.804	0.798

TABLE 2.6 – Performances sur les décisions brutes et filtrées.

Des gains significatifs sont observés avec des différences avoisinant les 0.20 point de F-mesure pour quatre des neufs classes de prétention. Les décisions non filtrées sont généralement trop longues et mêlent informations utiles et bruit car elles répondent dans la majorité des cas à plusieurs demandes simultanément. La sélection des segments permet donc de séparer les demandes dans le corps du document, ce que fait aussi le juge lorsqu'il répond de façon organisée et méthodique aux prétentions des parties.

Conclusion

Les modèles présentés montrent qu'il est aisé de déterminer la présence d'une demande spécifique dans le corps d'une décision mais que la prédiction du résultat de celle-ci est beaucoup plus complexe car plusieurs éléments sont susceptibles de limiter les performances. Le premier concerne le nombre d'exemples disponibles, souvent limité par la difficulté de l'annotation de documents juridiques nécessitant du temps et une expertise puisque ce type de tâche requiert une lecture attentive de jugements souvent longs et peu accessibles. Le second problème se situe au niveau de la capacité des modèles à traiter des formes grammaticales et syntaxiques complexes qui nécessitent des architectures profondes. En pratique, il est difficile d'augmenter la profondeur d'un réseau lorsqu'il y a trop peu d'exemples disponibles. Le recours à des techniques spécifiques est donc un pré-requis à l'obtention de meilleures performances. Enfin, malgré la réduction de la taille des décisions par la sélection des segments les plus informatifs, les modèles employés doivent être en mesure de traiter des séquences longues de plusieurs centaines à plusieurs milliers d'éléments dans certains cas de figure. Les modèles à base de LSTM [Hochreiter and Schmidhuber, 1997] sont en mesure de répondre à ce besoin au prix d'un temps de calcul élevé tandis que les approches de l'état de l'art à base d'attention et de Transformers [Vaswani et al., 2017] ne sont pas en mesure de résoudre des tâches dont les entrées excèdent quelques centaines de mots. Les chapitres suivants répondent en partie à ces problématiques.

Chapitre 3

Petits échantillons et apprentissage one-shot

Déterminer l'issue d'un litige en exploitant les arguments juridiques décrits par le juge constitue une tâche d'importance dans l'analyse prédictive du sens du résultat. La complexité du langage juridique requiert des modèles capables de traiter des structures syntaxiques et grammaticales sophistiquées, par conséquent, l'utilisation de techniques d'apprentissage modernes à base de réseaux de neurones s'impose naturellement. L'application de telles techniques s'avère toutefois difficile lorsque les jeux de données disponibles et étiquetés sont de petite taille, car l'annotation manuelle est coûteuse et nécessite un expert. Ce chapitre propose un modèle permettant de meilleures performances dans ce contexte de rareté en combinant une approche *one-shot* avec un mécanisme d'attention et l'extraction de termes clés en parallèle. Les résultats obtenus sont relatifs à la prédiction du sens du résultat de décisions issues de différentes catégories de demandes sur des décisions judiciaires de cour d'appel par l'exploitation de portions de motifs exposés par le juge.

Ce chapitre présente en premier lieu les caractéristiques des données exploitées ainsi que le modèle proposé. Les performances sur chaque catégorie de demande sont reportées dans un second temps.

3.1 Apprentissage en petit échantillon

La pratique de l'apprentissage automatique sur des jeux de données connus et partagés tend à ne pas considérer certaines problématiques rencontrées en situation réelle. De nombreux domaines souffrent de l'absence de bases labélisées, c'est le cas dans le traitement de données juridiques où le coût de l'annotation est souvent prohibitif. Cette section présente les obstacles rencontrés, la construction des bases et les alternatives permettant l'estimation effective de modèles sophistiqués avec peu d'exemples.

3.1.1 Données exploitées

3.1.1.1 Étiquetage des données

La détermination du résultat d'une demande formulée par une partie nécessite la construction de jeux de données et l'annotation de décisions de justice. Cette tâche est effectuée en deux temps et repose sur la compétence de juristes. En premier lieu, des arrêts de cour d'appel sont sélectionnés en fonction de la présence d'un certain type de demande afin de compiler une centaine de documents. A partir de cette base, chaque décision est analysée par un expert qui va extraire des motifs, les motivations de fait et de droit permettant de justifier l'acceptation ou le rejet de la demande concernée. Ces extraits prennent la forme de passages issus de la décision, leur taille variant de quelques mots à quelques phrases contiguës seulement. Le résultat est quant à lui exprimé de manière binaire puisque le juge est obligé de statuer pour chaque demande. La Figure 3.1 donne plusieurs exemples relatifs à des demandes sur un changement de prénom et sur des remboursements de créances.

Ces exemples montrent l'expertise nécessaire et la maîtrise d'un niveau de langue supérieure pour pleinement extraire l'information. Le premier exemple, qui est vraisemblablement le plus simple, est déjà ambigu puisqu'il mêle les termes « confirmé » et « rejeté » que l'on peut qualifier de discriminants dans une zone très restreinte. Ainsi le juge confirme un rejet de la demande, c'est-à-dire qu'il s'inscrit dans la continuité du jugement de première instance. En absence

Que la cour n'étant pas mise en mesure d'apprécier l'intérêt légitime invoqué, le jugement déféré ne pourra qu'être confirmé en ce qu'il a rejeté la demande de modification du prénom de Charlotte ;	Rejet
L'ordonnance critiquée sera en conséquence infirmée et la créance admise conformément à la demande en l'absence de contestation sur ce point, sauf à en rectifier la ventilation dans la mesure où les conclusions comportent, comme la déclaration de créance elle-même, une anomalie puisqu'elles portent sur une créance globale de 618 129,55 euros constituée de deux créances dont la somme s'élève à 668 527,55 euros.	Acceptation
Le caractère vicié des menuiseries, parties privatives, n'est donc pas établi et les deux rapports d'expertise tendent davantage à pointer un problème d'entretien, que ce soit un défaut de ravalement, qui concerne les parties communes, ou une nécessité de changer des joints.	Rejet

FIGURE 3.1 – Exemples de motifs associés à leur résultat.

de contexte, la résolution de la tâche par un profane est difficile et montre la difficulté pour un utilisateur quelconque de s'informer et de se familiariser avec les outils juridiques de façon autonome.

3.1.1.2 Construction de la base

Ainsi cinq jeux de données sont construits pour couvrir différents types de demande. Ces derniers sont relatifs à des demandes de changement de nom (600.NOM), à des remboursements de créances (600.DEC), à la responsabilité d'avocats (500.RES et 400.RES) et à des demandes de dommages et intérêts pour un déficit fonctionnel permanent (300.DOM). Les tailles de chaque jeu sont reportées en Table 3.1.

Le faible nombre d'observations par catégorie est la conséquence de plusieurs facteurs. Le premier est relatif à la difficulté de l'annotation manuelle qui requiert plusieurs heures de travail par base. Le second concerne la rareté de

Données	600.NOM	600.DEC	500.RES	400.RES	300.DOM
Taille	74	96	100	100	98

TABLE 3.1 – Taille des différents jeux de données.

certains types de demandes puisque l'accès à la jurisprudence des cours judiciaires d'appel n'est pas libre et que seul un sous-ensemble (environ 5%) est disponible en ligne[1]. Enfin, lorsque la motivation du juge est trop implicite dans les motifs et qu'il n'est pas possible d'extraire un passage suffisamment informatif, la décision n'est alors pas prise en compte.

3.1.2 Représentation et estimation

3.1.2.1 Représentation et modélisation one-shot

Le développement de l'apprentissage profond a produit de nouvelles approches se basant exclusivement sur l'utilisation de réseaux de neurones, en particulier pour la résolution de tâches de classification. Les réseaux de neurones récurrents de type LSTM permettent par exemple d'intégrer l'aspect temporel dans le traitement de données à caractère séquentiel (textes, séquences de mots) grâce à l'exploitation dynamique d'une cellule de mémoire. L'apprentissage profond permet de représenter et d'encoder de l'information portée par des entités sémantiques.

Traditionnellement, les modèles servant à définir les représentations se basent sur la construction de tableaux binaires ou fréquentiels (TF-IDF), dans lesquels les caractéristiques représentent la présence ou la fréquence d'un terme, mot ou n-grammes dans un document donné. Ces approches naïves trouvent rapidement leurs limites puisqu'elles nécessitent la manipulation de matrices dont la taille dépend du vocabulaire et dont le contenu est souvent très clairsemé. Les modèles de plongement sémantique exploitent le principe des auto-encoder [Schmidhuber, 2014] en se basant sur une tâche auto-supervisée dans laquelle, le réseau cherche à prédire un mot connaissant son contexte, permettant ainsi

1. Base CAPP : https://www.data.gouv.fr/fr/datasets/capp/

de définir une représentation sur des espaces de taille réduite. S'ajoute à cela, l'utilisation de techniques permettant de traiter des termes n'existant pas dans le vocabulaire initial. Ainsi, les modèles FastText [Bojanowski et al., 2017] et BERT [Devlin et al., 2019] exploitent des n-grammes afin de reconstruire des mots inconnus tandis qu'ELMo [Peters et al., 2018] et Flair [Akbik et al., 2019] travaillent au niveau des caractères qu'ils couplent à des couches de LSTM pour cela. Ces représentations vectorielles peuvent par la suite être utilisées pour répondre à des objectifs de classification ou intervenir en tant que composante dans une chaîne de traitements plus riche [Angelidis et al., 2018, Glaser et al., 2018]. Contrairement aux approches traditionnelles de construction de représentation (TF-IDF, n-grammes...), la construction de ces représentations ne requiert aucune définition préalable de caractéristiques qui seront utilisées pour les construire.

Le dernier mécanisme permettant pleinement l'exploitation de tous les éléments des séquences est l'attention [Bahdanau et al., 2014, Luong et al., 2015]. Dans sa forme la plus simple et exploitée dans ce chapitre, elle permet de pondérer chaque mot d'une phrase afin d'obtenir une représentation indépendante du nombre d'éléments la composant, cette transformation est communément appelée *many-to-one* ou *seq-to-one*. Généralement couplée à une couche de LSTM, elle permet de limiter la perte d'informations entre éléments distants à l'intérieur de la séquence grâce à des connexions directes. L'attention se substitue ainsi à l'utilisation de la cellule de mémoire du réseau récurrent pour représenter la séquence entière.

Pour une séquence $\boldsymbol{S} = [\boldsymbol{s_0}, \boldsymbol{s_1}, \ldots, \boldsymbol{s_t}] \in \mathbb{R}^{t \times d}$, de longueur t et dont les éléments sont représentés par des vecteurs de taille d, l'attention $\boldsymbol{y} \in \mathbb{R}^{1 \times d}$ est obtenue en effectuant un produit entre un vecteur de scores et la séquence :

$$\boldsymbol{y} = \sigma(\boldsymbol{c}\boldsymbol{W}\boldsymbol{S}^\intercal)\boldsymbol{S}$$

où $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ est une matrice de paramètres apprise par rétropropagation et $\boldsymbol{c} \in \mathbb{R}^{1 \times d}$ la mémoire terminale de la couche récurrente précédant l'attention. En cas d'absence de cellule de mémoire $\boldsymbol{c}$, celle-ci peut être remplacée par un élément $\boldsymbol{s_i}$ de $\boldsymbol{S}$ ou par une fonction capable de réduire la taille de la séquence à

un seul élément (moyenne, pooling...). La fonction de score peut elle aussi être modifiée mais l'utilisation de la softmax a un avantage majeur puisque les poids sont positifs et leur somme vaut 1, facilitant grandement leur interprétation. Ainsi un mot considéré comme important aura une pondération plus élevée qu'un autre jugé moins utile par le modèle.

3.1.2.2 Estimation en petit échantillon

Du fait de leurs propriétés intrinsèques (e.g. nombre important de paramètres), les modèles à base d'apprentissage profond nécessitent de larges quantités de données (labélisées) pour être entraînés [Slingerland et al., 2018]. Cet écueil est central et limite sévèrement leur potentielle utilisation dans le domaine légal ; domaine dans lequel il est le plus souvent difficile de mobiliser des experts – ce qui souvent induit des contextes d'étude dans lesquels seuls peuvent être considérés des jeux de données coûteux et de taille réduite. Cette limitation contribue à expliquer le faible nombre de travaux s'intéressant à l'apprentissage profond pour la classification dans le domaine légal.

Des recherches actives en apprentissage automatique travaillent sur la réduction de la dépendance aux volumétries importantes de données labélisées. Nous distinguons en particulier les axes de recherches connectés visant (i) la réutilisation de modèles entraînés pour des contextes applicatifs ou des tâches proches comme l'apprentissage par transfert, l'apprentissage multi-tâches, les techniques de fine-tuning, (ii) à exploiter des données non-labelisées (via l'apprentissage non-supervisé ou auto-supervisé de représentations), ou (iii) à tirer parti tant que possible de l'information véhiculée par des annotations disponibles.

L'apprentissage *one-shot* [Fei-fei et al., 2006] consiste à transformer une tâche de classification en une autre de type binaire, dans laquelle une mesure de similarité est calculée entre les représentations latentes d'une paire d'observations. En pratique, cette mesure revêt la forme d'une fonction distance (distance euclidienne, distance de Manhattan...) ou d'une fonction de similarité (produit scalaire, cosinus, divergence KL...) nécessairement dérivable pour assurer la rétropropagation à travers le réseau de neurones. L'architecture la plus commune est présentée dans la Figure 3.2.

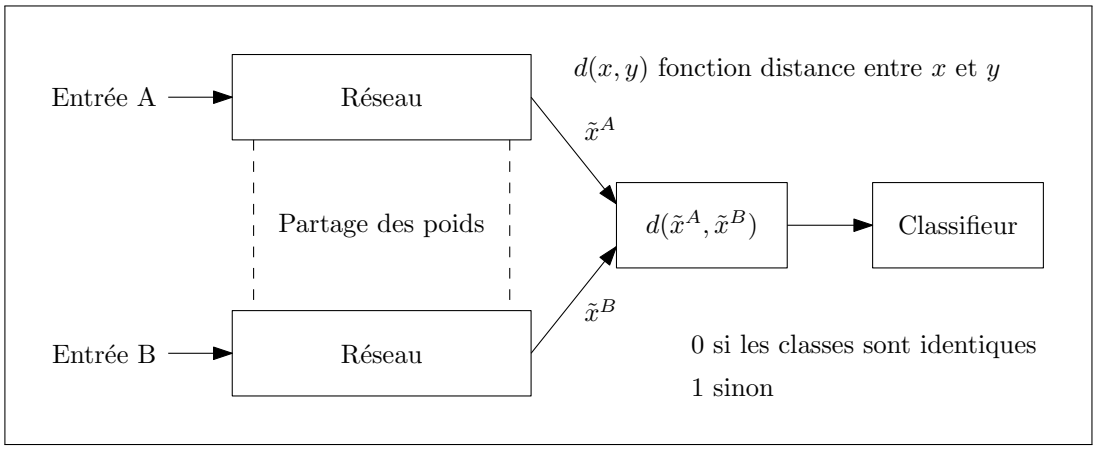


FIGURE 3.2 – Modèle *one-shot* simple.

La fonction distance est appliquée indépendamment sur chaque caractéristique et une couche finale de classification est ajoutée en fin de réseau pour effectuer la prédiction. Puisque le modèle exploite des paires d'observations, il est ainsi possible d'augmenter artificiellement le nombre d'exemples. Pour un nombre initial n d'observations, $\frac{n(n-1)}{2}$ paires peuvent être construites, réduisant ainsi les risques de sur-apprentissage pour un nombre de paramètres identique.

L'autre approche permettant d'estimer des modèles sous contrainte d'un faible nombre d'observations consiste à augmenter la taille de la base en générant des données artificiellement. Dans la littérature de traitement de l'image, cette tâche peut être effectuée par de simples manipulations comme des rotations ou des translations [Shorten and Khoshgoftaar, 2019]. Celles-ci sont plus complexes à mettre en œuvre dans le cadre du traitement du langage puisqu'il est nécessaire de conserver une certaine structure grammaticale et syntaxique pour conserver le sens du texte, une simple négation pouvant affecter l'étiquette d'un exemple dans des problématiques de classification.

Il existe quatre outils principaux permettant la création de nouvelles données [Wei and Zou, 2019] : l'adjonction de bruit, l'exploitation de synonymes, la génération par un modèle de langue et la traduction arrière. Dans le traitement de la langue, l'ajout de bruit peut se faire de plusieurs manières : en remplaçant, ajoutant, supprimant aléatoirement certains mots et en effectuant des permutations aléatoires. Cette pratique, très simple à mettre en œuvre, est limitante puisqu'il est impossible d'assurer que le sens du texte soit conservé,

en particulier dans le domaine juridique où le placement d'un mot peut considérablement affecter la prédiction et l'étiquette. L'exploitation de synonymes a des implications différentes grâce à la conservation de la structure des données initiales (hors conjugaison en français). Cette méthode est efficace dans le cadre de tâches générales dans lesquelles la substitution a peu d'influence sur le sens du texte et où le niveau de langage étudié est commun. Le jargon juridique ne respectant malheureusement pas cette règle, de nombreuses expressions n'ont pas de reformulation simple et l'existence d'un synonyme n'est que rarement assurée. La méthode par génération quant à elle permet, sous condition d'un modèle suffisamment entrainé et performant, d'assurer une cohérence syntaxique sans dénaturer la phrase initiale. Puisqu'un modèle de langue émet la probabilité d'apparition d'un terme en fonction des mots ou caractères précédents, il est possible de substituer des portions aléatoires par d'autres jugées tout aussi probables. En pratique, le modèle Flair est utilisé pour effectuer cette tâche au niveau des caractères, l'espace servant de séparateur. Enfin, la dernière approche consiste à exploiter des modèles de traduction en partant de la langue d'origine puis en passant par une ou plusieurs traductions intermédiaires pour revenir à la langue initiale. Grâce au développement d'outils de plus en plus performants (DeepL, Google traduction...), les résultats obtenus permettent de conserver le sens du texte tout en paraphrasant et reformulant certains aspects. La forme peut ainsi être différente tandis que le fond reste similaire. La Figure 3.3 propose plusieurs exemples passant par deux langues intermédiaires.

3.2 Architecture et expérimentations

Afin d'entraîner des modèles sur des volumes restreints, l'approche que nous proposons exploite plusieurs éléments. Le modèle entraîne une représentation vectorielle des phrases en entrée à l'aide de plongements lexicaux pré-entrainés et de réseaux siamois munis d'un mécanisme d'attention. Cette représentation est injectée en parallèle dans un second réseau qui va la coupler avec des termes réputés très discriminants et préalablement sélectionnés afin de limiter au mieux les risques de sur-apprentissage.

Attendu que dans ces conditions, les appelants qui ne rapportent pas la preuve de l'intérêt légitime devant conduire à la suppression du second prénom de leur fille mineure doivent être déboutés de leur demande ;	Texte initial
Considérant que, dans ces conditions, les requérants, qui ne justifient pas d'un intérêt légitime devant conduire à la suppression du second prénom de leur fille mineure, doivent être déboutés de leur demande ;	Anglais → Portugais
Au vu de ces circonstances, l'absence de démonstration par la requérante de l'existence d'un intérêt légitime devant conduire à la suppression du second prénom de sa fille mineure doit être rejetée.	Chinois → Suédois
Dans les circonstances, toutefois, la recourante, qui n'a pas apporté la preuve d'un intérêt légitime qui aurait dû conduire à la suppression du second prénom de sa fille mineure, doit être rejetée.	Japonais → Bulgare

FIGURE 3.3 – Exemples d'augmentation par traduction (DeepL).

Cette section présente dans un premier temps l'architecture générale puis compare les performances issues de différents scenarii, notamment les gains à l'utilisation d'une approche *one-shot* et de l'augmentation des données.

3.2.1 Architecture générale

3.2.1.1 Entraînement des plongements sémantiques

La représentation des mots et des phrases par le biais d'un plongement sémantique (embedding) est une étape pouvant impacter significativement les performances d'un modèle de prédiction. Plusieurs méthodes de représentation allant de la simple approche fréquentielle TF-IDF aux plongements sémantiques tels que FastText, ELMo, Flair et CamemBERT [Martin et al., 2019] (dérivé de BERT pour la langue française) sont comparées.

Puisque le langage juridique est complexe, muni d'un vocabulaire spécifique et d'une syntaxe s'éloignant d'un corpus général comme Wikipédia, il est nécessaire d'effectuer un entraînement sur une base spécialement construite pour résoudre des tâches relatives au droit. Pour cela, plusieurs bases libres d'accès sont compilées du portail DILA [2], notamment les bases CAPP, CASS, JURI et LEGI composées d'arrêts de cours d'appel, de cassation et de textes de lois. La taille totale s'estimant à environ 700 millions de mots après suppression des phrases doublons. Compte tenu de la sensibilité des textes juridiques français à la casse [3] et de la ponctuation particulière, notamment sur l'utilisation systématique du point-virgule comme séparateur, le pré-traitement se limite à une tokenization simple au niveau des mots. FastText, ELMo et Flair sont entraînés de zéro et plusieurs modèles sont estimés avec des tailles de représentations allant de 32 à 256 dimensions. En pratique, la similarité syntaxique des textes juridiques, le vocabulaire et les expressions redondantes rendent la tâche de pré-entraînement simple avec une faible perplexité dès l'utilisation de tailles 64 ou plus. Cela montre entre autres que le langage utilisé par les juristes a une structure à la fois prévisible et régulière malgré un vocabulaire et des expressions rarement employés dans un cadre plus général. Notons que compte tenu des temps de calcul induits par CamemBERT, les représentations utilisées pour ce modèle proviennent du *checkpoint* publié par ses auteurs.

3.2.1.2 Modèle proposé

Le modèle proposé est présenté dans la Figure 3.4 et peut être décomposé en deux zones. La partie haute sur laquelle nous nous concentrons d'abord est composée du réseau siamois qui exploite des plongements pré-entrainés, un mécanisme d'attention et une fonction de similarité afin de discriminer des paires de phrases appartenant ou non à la même classe. L'estimation de cette partie du réseau permet entre autre, par le biais du mécanisme d'attention, de repérer les termes jugés les plus discriminants, sachant que le modèle opère au niveau des mots directement. Elle permet aussi d'obtenir un plongement sémantique

2. https://www.dila.premier-ministre.gouv.fr/repertoire-des-informations-publiques/les-donnees-juridiques

3. La séparation des différentes parties des décisions est souvent faite par le biais d'expressions en majuscules.

de chaque phrase, plus précis qu'une simple moyenne des termes. Cette représentation est réutilisée en parallèle par la partie basse de l'architecture, servant à la classification finale des observations.

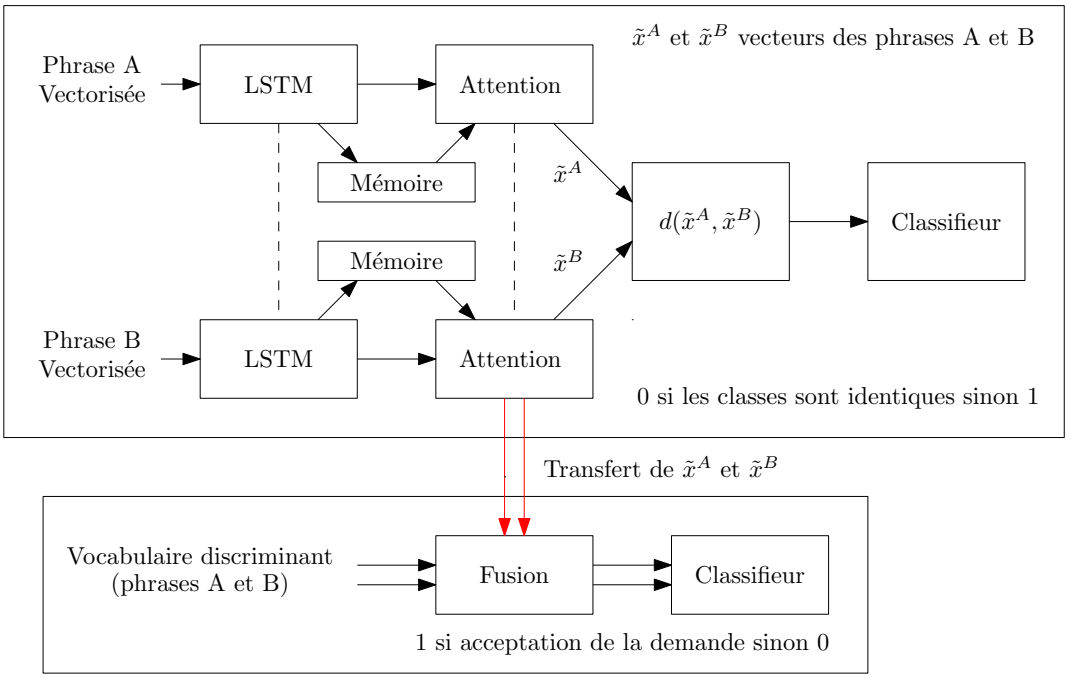


FIGURE 3.4 – Architecture générale du modèle.

Le réseau siamois (partie haute) se décompose en deux réseaux symétriques partageant leurs poids mais recevant cependant des données différentes. Ces deux modèles reposent sur l'application d'une couche LSTM sur des entrées pré-vectorisées [4] afin d'obtenir une première représentation de la phrase. Celle-ci est issue de la cellule de mémoire du réseau récurrent, que l'attention repondère par la suite. En effet, l'attention utilisée est de type *seq-to-one*, c'est-à-dire qu'elle reçoit une séquence et estime en sortie une représentation compacte et indépendante de la longueur de chaque séquence, la taille finale étant la même pour chaque phrase. Cette projection est réexploitée en parallèle par la partie basse de l'architecture.

L'apprentissage *one-shot* dépend du calcul d'une fonction de similarité permettant de discriminer les paires d'observations passant par le réseau siamois.

4. Les plongements ont des paramètres fixés afin de ne pas augmenter significativement leur nombre puisque dans ce type de configuration, le sur-apprentissage est quasi systématique et instantané.

Le choix de cette fonction dépend fortement de la tâche et du type de données. En pratique, nous observons qu'une différence en valeur absolue permet d'obtenir une meilleure convergence et globalement de meilleurs résultats dans nos expérimentations. Cette fonction est appliquée indépendamment sur chaque caractéristique issue de la représentation vectorielle des phrases. Ainsi pour la caractéristique i, $d_i(\tilde{x}^A, \tilde{x}^B) = |\tilde{x}_i^A - \tilde{x}_i^B|$.

Enfin, la sortie de la couche d'attention permet d'obtenir une représentation vectorielle de taille fixe, indépendante de la longueur de chaque séquence. Dans la partie basse du modèle, ces représentations sont concaténées avec des mots et expressions jugés discriminants et propagés dans un dernier classifieur (couche linéaire) effectuant la prédiction. Puisque la tâche est binaire (la demande étant acceptée ou rejetée), un élément est discriminant s'il a une forte présence dans une classe et une faible présence dans l'autre. Si f_j^i est la fréquence du n-grammes j dans la classe $i \in \{0, 1\}$ alors les 50 n-grammes maximisant la différence $|f_j^1 - f_j^0|$ sont sélectionnés.

3.2.2 Résultats

Cette section présente les résultats afin de mettre en lumière le rôle de l'apprentissage *one-shot* sur la capacité prédictive d'un modèle par rapport à des approches plus classiques de la littérature (régression logistique, forêts aléatoires, SVM). Enfin, les conséquences de l'augmentation de données et la capacité interprétative de l'attention sont discutées en dernier lieu pour justifier le choix de l'architecture.

3.2.2.1 Protocole et performances

Le protocole se base sur une validation croisée en 10 étapes pour chaque modèle. Afin de garantir l'indépendance lors de la génération de données à chaque évaluation, la base est d'abord divisée puis seule la base d'entraînement est augmentée à l'aide de la technique présentée dans la section précédente. Ainsi, la prévision se fait uniquement sur de vraies observations. Les tâches à base de réseaux de neurones minimisent deux entropies croisées en même

temps (celle de la tâche *one-shot* et celle de la classification finale), utilisent un *dropout* de 0.25 [Srivastava et al., 2014], un taux d'apprentissage de 0.001 et un optimiseur de type Adam [Kingma and Ba, 2015]. Tous les modèles sont comparés selon la précision et la F-mesure, pondérées relativement à la taille du jeu de test car la dernière itération de validation croisée peut être déséquilibrée.

Dans un premier temps sont discutées les performances des algorithmes communément utilisés dans la littérature, plus précisément la régression logistique, la forêt aléatoire et le SVM. Les données fournies aux modèles sont de deux natures. Dans un premier temps, les phrases sont encodées sous forme de matrice TF-IDF, puis sont fusionnés des plongements sémantiques provenant du modèle ELMo pré-entrainé avec des n-grammes jugés discriminants et représentés par une matrice binaire. Les résultats sont décrits dans la Table 3.2.

	SVM		Forêt aléatoire		Logistique	
	P	F	P	F	P	F
600.NOM*	0.798	0.748	0.782	**0.786**	0.743	0.728
600.NOM**	0.820	0.781	0.867	**0.813**	0.752	0.739
600.DEC*	0.669	0.788	0.747	**0.795**	0.658	0.767
600.DEC**	1.000	0.908	0.931	**0.959**	0.889	0.902
500.RES*	0.591	0.568	0.651	0.619	0.674	**0.645**
500.RES**	0.716	**0.659**	0.623	0.636	0.639	0.570
400.RES*	0.943	0.795	0.826	**0.837**	0.810	0.828
400.RES**	0.783	0.789	0.924	**0.893**	0.709	0.736
300.DOM*	0.827	0.834	0.847	**0.854**	0.847	0.825
300.DOM**	0.963	0.931	1.000	**0.952**	1.000	0.910

* TF-IDF — Précision (P) et F-mesure (F)

** Plongement sémantique moyen + sélections de n-grammes

TABLE 3.2 – Comparaisons des performances de classification avec différentes entrées.

Il parait évident à la vue de ces résultats que l'utilisation de matrices TF-IDF baisse systématiquement les performances, occasionnant des pertes pouvant atteindre plus de 0.15 point de F-mesure sur les données relatives aux créances

(600.DEC) et 0.10 point sur celles relatives au paiement de dommages et intérêts (300.DOM). Ces deux types de demande tendent à énumérer de nombreuses sommes d'argent difficilement exploitables dans une représentation fréquentielle puisque chaque montant à une forte probabilité d'être unique et donc d'être trop peu fréquent pour être exploité. L'utilisation du plongement sémantique permet de partiellement prendre en compte ce type d'information car les modèles savent contextualiser les nombres.

La deuxième expérimentation consiste à pleinement exploiter l'architecture présentée dans la section précédente en utilisant l'apprentissage *one-shot* couplé au mécanisme d'attention et à l'extraction de termes et d'expressions discriminantes. Dans cette première version, les données ne sont pas augmentées et donc l'apprentissage reste cantonné aux séquences initiales. Dans la Table 3.3 sont présentées les performances en fonction du type de plongement sémantique. À noter ici que la taille des représentations est fixée à 64 dimensions (hormis pour CamemBERT) car l'augmenter entraîne en pratique une perte de performance liée au surapprentissage. Afin que CamemBERT converge convenablement, le taux d'apprentissage est réduit à 0.0001 pour ce modèle uniquement.

	FastText		ELMo		Flair		CamemBERT	
	P	F	P	F	P	F	P	F
600.NOM	0.842	0.818	0.867	**0.846**	0.842	0.843	0.822	0.820
600.DEC	0.945	0.967	0.975	0.986	0.986	**0.992**	0.976	0.985
500.RES	0.756	0.701	0.774	**0.734**	0.805	0.709	0.742	0.718
400.RES	0.868	0.882	0.907	**0.908**	0.828	0.854	0.904	0.898
300.DOM	1.000	**1.000**	1.000	0.990	0.983	0.982	0.983	0.982

TABLE 3.3 – Précision et F-mesure en fonction des plongements lexicaux (sans augmentation).

Sur cette configuration, ELMo et Flair tendent à légèrement être plus performants dans la majorité des cas. Les raisons à cela sont multiples. Premièrement, FastText est un modèle bien plus simple car linéaire, avec un nombre restreint de paramètres, dans l'incapacité de modéliser des relations complexes ou de désambiguïser certains mots. Enfin, il est incapable de contextualiser les mots

et expressions malgré la présence de l'attention qui tend à répondre partiellement à ce problème. CamemBERT souffre de son côté d'une représentation de grande dimension et d'une absence de *fine-tuning* sur des données juridiques, cela joue évidemment sur la qualité globale du modèle. Comparativement aux approches précédentes, les modèles sont tous plus performants au prix d'une complexité accrue et d'une vitesse de convergence bien plus élevées.

3.2.2.2 Augmentation et interprétabilité

La génération de données comme décrite précédemment peut permettre d'améliorer les performances de certaines tâches si la génération conserve la cohérence grammaticale et syntaxique et permet de ne pas trop s'éloigner de la distribution des données initiales. Les trois méthodes évaluées, la traduction, la génération de mots manquants et l'usage de synonymes possèdent chacune des limites propres. Les deux premières dépendent fortement de la qualité du modèle utilisé tandis que la dernière peut occasionner l'apparition de termes mal adaptés et hors contexte, plus particulièrement lorsque la tâche concerne un domaine spécifique avec son propre vocabulaire et ses propres expressions. Les résultats des différentes approches sont présentés dans la Table 3.4 et exploitent les modèles ELMo ou Flair pour des raisons de performances.

	Traduction		Synonymes		Génération	
	P	F	P	F	P	F
600.NOM	0.870	**0.880**	0.852	0.844	0.868	0.878
600.DEC	0.986	**0.992**	0.978	0.988	0.982	0.990
500.RES	0.794	**0.817**	0.751	0.761	0.785	0.801
400.RES	0.918	**0.940**	0.888	0.908	0.910	0.932
300.DOM	1.000	**1.000**	0.987	0.985	1.000	0.990

TABLE 3.4 – Meilleures performances des modèles en fonction de la méthode d'augmentation.

Le modèle exploitant des synonymes est moins performant dans toutes les configurations car les phrases générées perdent leur sens, les termes juridiques ne se substituant pas correctement avec un niveau de langue plus commun.

La génération quant à elle reste cohérente mais conserve la structure initiale de la phrase contrairement à la traduction qui occasionne des reformulations permettant une plus grande variété de phrases. Les gains liés à l'augmentation des données sont compilés en Table 3.5.

	Augmentation		Base		Différence	
	P	F	P	F	ΔF*	ΔF**
600.NOM	0.870	**0.880**	0.867	0.846	+0.034	+0.067
600.DEC	0.986	**0.992**	0.986	**0.992**	+0.000	+0.033
500.RES	0.794	**0.817**	0.774	0.734	+0.083	+0.158
400.RES	0.918	**0.940**	0.907	0.908	+0.032	+0.047
300.DOM	1.000	**1.000**	1.000	**1.000**	+0.000	+0.048
Moyenne					+0.030	+0.076

* Différence de F-mesure avec et sans augmentation

** Différence de F-mesure entre les modèles de base les plus performants et le *one-shot* avec augmentation

TABLE 3.5 – Récapitulatif des meilleures performances.

Les gains restent cantonnés à seulement trois des cinq jeux de données car deux des tâches sont presque parfaitement résolues avec l'approche *one-shot* seule. Les gains sont particulièrement élevés pour les demandes les plus difficiles à traiter initialement, relatives à la responsabilité des avocats (500.RES).

L'ajout du mécanisme d'attention a pour but de faciliter la désambiguïsassion mais aussi de permettre un certain niveau d'interprétabilité après extraction des matrices de pondération. Grâce à la nature de la fonction *softmax*, les poids permettent directement de hiérarchiser les termes que le modèle considère comme importants. Cette pratique est d'autant plus intéressante lorsque la tokenization est faite au niveau des mots et non au niveau de syllabes ou de n-grammes comme cela est le cas avec les modèles dérivés de BERT. La Figure 3.5 montre quelques exemples de pondération où les mots les plus importants sont encadrés.

Le dernier exemple permet d'observer que le modèle considère qu'une négation en début de phrase est discriminante. Dans les autres cas, le modèle

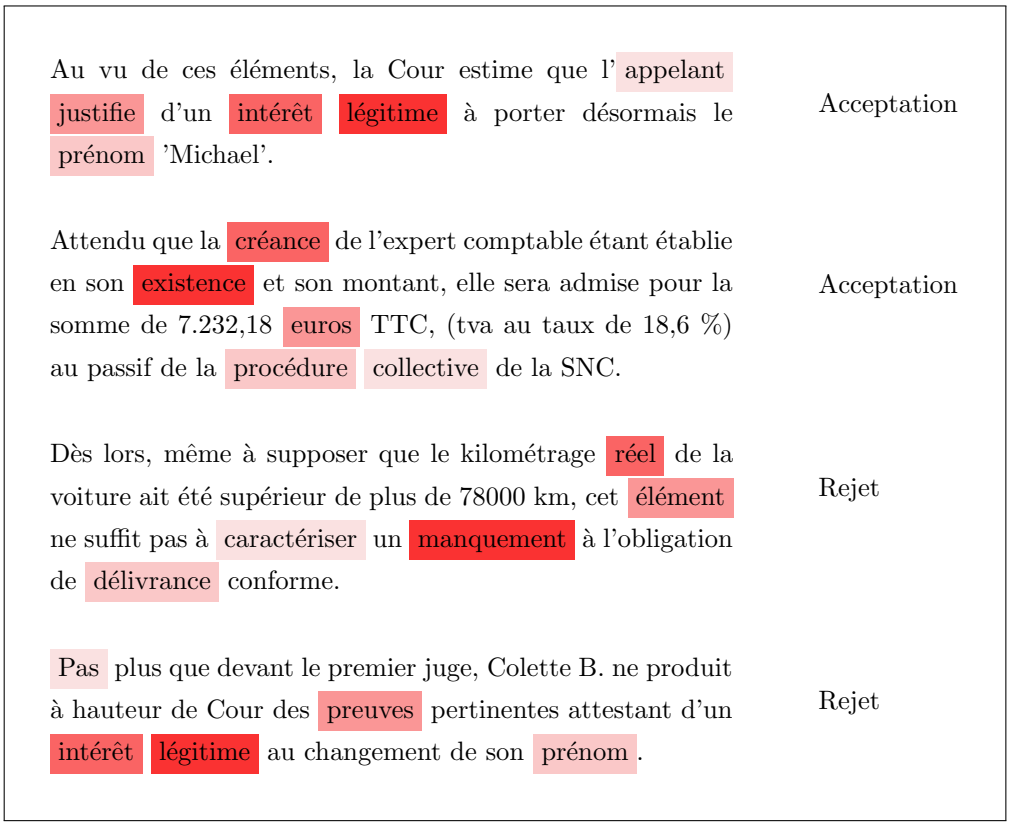


FIGURE 3.5 – Termes sur-pondérés par l'attention.

souligne des termes qui paraissent à première vue pertinents. Notons qu'une *softmax* pondère tous les éléments sans exception, l'analyse des mots pris indépendamment peut donc être trompeuse puisqu'elle ne prend pas en compte les interactions et les compensations que certains mots ont sur d'autres. Par exemple une double négation revient à une expression positive, dans ce cas, un modèle bien entraîné en fera abstraction.

Conclusion

La résolution de tâches de classification sur de petits ensembles de données peut être abordée à l'aide de plusieurs outils. L'apprentissage par transfert est essentiel lorsque peu d'exemples sont étiquetés tandis que des tâches auto-supervisées peuvent être définies en parallèle sur de grands corpus non annotés.

Dans ces conditions, les modèles de langue sont adaptés sur des tâches spécifiques en modifiant les paramètres déjà entraînés. Cependant, le risque de surrapprentissage nécessite tout de même l'ajout de contraintes consistant à figer une partie des poids du modèle et à surveiller étroitement le choix des hyperparamètres. L'utilisation d'architectures spécifiques comme les réseaux siamois ou de techniques d'augmentation de données permettent de partiellement relâcher ces contraintes et d'offrir un plus grand degré de liberté dans la résolution de tâches annexes. Ces outils permettent ainsi de gagner en stabilité et en qualité des prédictions tout en assurant de meilleures capacités de généralisation sur de nouvelles données.

Chapitre 4

Séquences longues et Transformers efficients

Une décision de justice est composée de trois éléments, un ensemble de faits permettant de retracer l'histoire et les différents évènements, un ensemble de motifs ayant pour but de justifier la décision du juge et le résultat lui-même. Lorsque l'objectif est de prédire ce dernier, seuls les éléments relatant les faits peuvent être, en situation réelle, connus *a priori*. Contrairement aux arguments juridiques exposés par le juge et parfois redondants, l'énumération des faits peut revêtir une forme très hétérogène en fonction des litiges de même nature. C'est notamment le cas pour la Cour Européenne des Droits de l'Homme (CrEDH) devant laquelle s'opposent des personnes physiques ou morales face à des États. L'analyse de ce type de décisions permet, par le biais de quelques règles, d'extraire automatiquement des ensembles de faits et les résultats associés. Cependant, l'exploitation de ces données requiert l'utilisation d'outils capables de traiter de très longues séquences, ce que les modèles de l'état de l'art de type Transformers ne sont pas en mesure de faire par défaut.

La première section de ce chapitre présente un état de l'art approfondi des architectures *efficientes* de Transformers en proposant par la suite des modèles alternatifs et performants. La seconde partie se focalise quant à elle sur l'exploitation de faits afin d'inférer le sens du résultat de décisions issues de la CrEDH.

4.1 Apprentissage en contexte long

Les modèles sémantiques à base de blocs de Transformers reposent sur le principe de l'attention. En se basant sur le calcul d'un score d'importance entre chaque paire de mots, de syllabes ou de caractères, l'attention permet de représenter chaque élément d'une séquence comme une somme pondérée de tous les éléments de celle-ci. Cependant, cette opération requiert le stockage d'une matrice de scores dont la taille augmente quadratiquement avec la longueur du segment. Ainsi, la grande majorité des modèles sémantiques (BERT, RoBERTa...) [Devlin et al., 2019, Liu et al., 2019] se limitent à des séquences de 512 éléments afin d'éviter des coûts en temps et en mémoire prohibitifs. En pratique, il n'est pas rare de rencontrer de longs documents dont la taille dépend des entités traitées, travailler au niveau des caractères s'avère plus difficile que de travailler au niveau des mots. Des modèles à même de traiter ce problème ont été développés par le biais de diverses méthodes d'approximations à base de noyaux, de factorisations, de variantes éparses ou de mécanismes de récurrence.

Les sections suivantes présentent des architectures en mesure de résoudre des problématiques en contexte long pour lesquelles des connexions entre des éléments spatialement ou temporellement éloignés sont facilitées.

4.1.1 Traitement des séquences longues

Les versions *efficientes* de l'*auto-attention* permettent de résoudre des tâches inabordables avec l'attention vanille [1] [Vaswani et al., 2017], que cela soit dans le cadre du traitement du langage ou du traitement d'images. Sans ces outils capables de traiter un contexte élargi, il est nécessaire d'effectuer directement des modifications sur les données en tronquant des phrases ou en diminuant la résolution d'images. Cependant, ces transformations peuvent affecter les résultats et les performances puisque les modèles entraînés n'ont dans ce cas qu'un accès partiel à l'information.

La littérature relative à l'attention efficiente s'est fortement élargie [Beltagy et al., 2020, Katharopoulos et al., 2020, Choromanski et al., 2021] grâce à la

1. Se réfère à l'attention traditionnelle utilisée dans BERT et RoBERTa.

diffusion des bibliothèques d'*HuggingFace* [Wolf et al., 2020] mais tend parfois à négliger certaines caractéristiques souhaitables au profit d'approches souvent complexes où l'efficience n'est réelle que sous certaines conditions (taille minimale, hyperparamétrage...).

Dans les paragraphes suivants, la taille du batch, le nombre de têtes d'attention, la longueur de la séquence et la taille de l'embedding seront représentés par les lettres $n, h, t, d \in \mathbb{N}^+$.

4.1.1.1 Les caractéristiques souhaitables

L'attention permet de pondérer chaque élément d'une séquence en fonction de tous les autres éléments de celle-ci. Dans l'entraînement d'un modèle de langue masqué (MLM) tel que présenté par BERT, où les phrases en entrée sont partiellement masquées par un token générique [MASK], l'attention permet d'obtenir une représentation bidirectionnelle directe du contexte. Cela diffère des approches à base de réseaux récurrents pour lesquels deux branches sont nécessaires, l'une travaillant sur le texte à l'endroit, l'autre à l'envers.

Cette simple caractéristique permet d'obtenir des modèles facilement parallélisables puisqu'il n'est ainsi plus nécessaire de traiter les entrées de manière séquentielle. Cela permet aux Transformers d'être des modèles performants, pouvant être entraînés efficacement sur de grandes quantités de données. Ils sont donc en capacité d'intégrer de larges connaissances par la suite réexploitables dans la résolution de tâches multiples [Radford et al., 2019]. L'attention exploite trois éléments, les requêtes (*queries* notées $\boldsymbol{Q}$), les clés (*keys* notées $\boldsymbol{K}$) et les valeurs (*values* notées $\boldsymbol{V}$). Dans la grande majorité des modèles, ces trois éléments sont issus de la projection d'une séquence notée $\boldsymbol{X} \in \mathbb{R}^{t \times d}$ de longueur t et représentée par des vecteurs de taille d. En pratique, l'attention est subdivisée en plusieurs têtes afin d'être calculée plusieurs fois en parallèle. Ainsi, la sortie d'une tête d'attention $i \in [1, \ldots, h]$ notée $\boldsymbol{Y}_i \in \mathbb{R}^{t \times d_i}$ avec $d_i = \frac{d}{h}$ est calculée suivant :

$$\boldsymbol{Y}_i = Softmax\Big(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^\top}{\sqrt{d_i}}\Big)\boldsymbol{V}_i \tag{4.1}$$

Avec $\boldsymbol{Q}_i = \boldsymbol{X}\boldsymbol{W}_i^q$, $\boldsymbol{K}_i = \boldsymbol{X}\boldsymbol{W}_i^k$, $\boldsymbol{V}_i = \boldsymbol{X}\boldsymbol{W}_i^v$ et $\boldsymbol{W}_i^q, \boldsymbol{W}_i^k, \boldsymbol{W}_i^v \in \mathbb{R}^{d \times d_i}$.

Les versions efficientes proposent de substituer cette formule par une alternative permettant de passer outre le stockage et le calcul complet de la *softmax* ou en approximant directement celle-ci. Les modèles répondant à cette problématique sont généralement regroupés en quatre catégories :

— les modèles à base de récurrence comme *Transformers-XL* [Dai et al., 2019] et *Compressive Transformers* [Rae et al., 2019] ;
— les modèles à base de factorisation ou de kernelisation tels que *Linformer* [Wang et al., 2020b] ou *Performer* [Choromanski et al., 2021] ;
— les modèles à base de clustering tels que *Reformer* [Kitaev et al., 2020] ;
— les modèles éparses avec un schéma fixe ou partiellement aléatoire comme *Longformer* [Beltagy et al., 2020] ou *Big Bird* [Zaheer et al., 2020].

Prise en compte des positions Les Transformers dépendent d'un second élément, l'embedding de position. Afin de discriminer les éléments identiques mais présents à des positions différentes dans le segment, ces modèles intègrent des paramètres supplémentaires ajoutés à l'embedding initial. L'objectif étant de désambiguïser chaque terme car un Transformer n'est pas en mesure de le faire naturellement : chaque mot ou token n'étant qu'une représentation invariante à la permutation. Puisque ces paramètres dépendent de la longueur de la séquence et que ces modèles limitent traditionnellement la taille à 512 éléments, il n'est pas possible sans modification, d'entraîner un Transformer sur des phrases plus longues. En pratique, le mécanisme d'attention tend à surpondérer en moyenne les éléments proches entre eux [Clark et al., 2019], en particulier dans les tâches de MLM, de traduction et plus généralement dans les tâches *seq-to-seq*. Puisque cette tendance à la localité affecte la qualité des prévisions, il est commun de simplement dupliquer la représentation de la position pour conserver les caractéristiques locales [Beltagy et al., 2020]. Pour des segments composés de 512 éléments, il y aura 8 copies successives pour traiter des entrées de longueur 4096, la Figure 4.1 illustre le mécanisme.

Encoder la position absolue est critiquable et dépend nécessairement de la longueur de la séquence. A ce titre, des approches alternatives ont été développées afin de prendre en compte les positions relatives entre les éléments. La méthode la plus simple, consiste à calculer la distance (bornée) entre deux

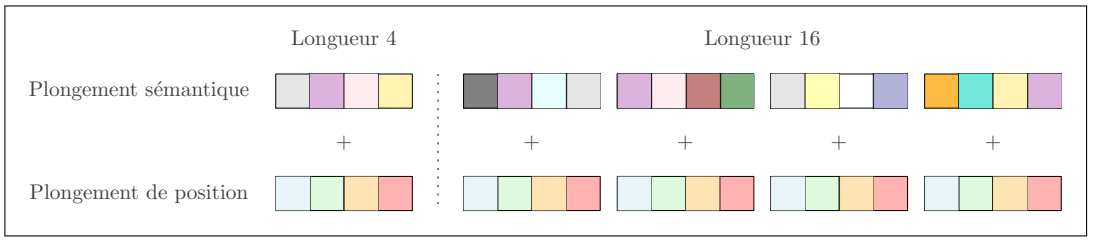


FIGURE 4.1 – Duplication du plongement de position.

tokens et d'encoder cette dernière [Shaw et al., 2018]. Dans cette situation, la position est directement intégrée dans la matrice des scores et non pas ajoutée dans la couche initiale du modèle. Cela nécessite donc de calculer explicitement la *softmax*, ce qui n'est pas toujours souhaitable dans des approches efficientes. Une autre alternative [Ho et al., 2019], réutilisée dans *Reformer*, consiste à factoriser la matrice de positions afin de réduire significativement le nombre de paramètres à stocker dans le cadre de très longues séquences tout en assurant l'unicité de chaque position. L'usage de fonctions périodiques comme présenté dans le papier originel [Vaswani et al., 2017] est repris dans le modèle *Roformer* [Su et al., 2021] en intégrant dans les calculs de $\boldsymbol{Q}$, $\boldsymbol{K}$ et $\boldsymbol{V}$ une matrice supplémentaire indépendante de la longueur de la séquence et contenant implicitement l'information concernant les distances relatives. Enfin, le modèle *ALiBi* [Press et al., 2021] se passe entièrement de la prise en compte des positions sur la couche initiale et ajoute des constantes dans les matrices de scores. Ces constantes dépendent de la distance entre deux éléments et de la tête utilisée afin d'augmenter mécaniquement la pondération entre tokens proches. Cela permet entre autre, de faciliter la généralisation du modèle à des séquences plus longues.

Bien que ces mécanismes de traitement positionnel peuvent présenter un intérêt dans un entraînement à partir de zéro, leur utilisation reste difficile dans des modèles existants puisqu'il est souvent nécessaire de réentraîner la couche de positionnement, voire l'attention si celle-ci requiert des paramètres supplémentaires. En réalité, les modèles intègrent implicitement les concepts de proximité et de localité, les matrices de scores montrant systématiquement une surpondération sur leur bande diagonale [Clark et al., 2019]. Toutes les expériences menées par la suite, feront usage de l'approche par duplication.

Caractéristiques souhaitables Les contributions sur les Transformers efficients tendent à se dégager de plus en plus des modèles initiaux, en intégrant des mécanismes complexes ne cherchant pas toujours à reproduire les principes de base. Cet aspect rend l'utilisation des modèles existants et déjà entraînés difficile puisque le comportement à l'intérieur de la couche d'attention n'est plus adapté aux poids déjà estimés. Substituer la *softmax* par des kernels par exemple ne permet pas de reproduire son comportement, dès lors, des pertes importantes de performances peuvent être observées. Dans ces conditions les auteurs préfèrent généralement réentraîner le modèle à partir de zéro mais cette opération est coûteuse [2].

En pratique, seuls quelques modèles de base tels que BERT et RoBERTa possèdent des checkpoints pour un grand nombre de langues et de types de données (français, portugais, suédois, langage médical, réseaux sociaux...), la capacité des approches efficientes à pouvoir s'y greffer avec un *fine-tuning* minimal est primordial afin de réduire significativement les coûts et le temps alloué à l'entraînement. Ce principe d'*adaptation* est d'autant plus important pour les modèles très larges car les Transformers apprennent en *few-shot*, leurs capacités évoluant avec le nombre de paramètres et le nombre d'exemples qu'ils traitent [Brown et al., 2020]. Dans ces conditions, compenser une perte de performance est d'autant plus coûteux que la taille du jeu de données et le temps d'entraînement initiaux sont importants.

La seconde caractéristique souhaitable concerne l'ajout de paramètres qui dépendent de la taille de la séquence. Ce problème est adressé par la duplication de l'embedding de position mais il n'existe pas de solution simple, hormis la modification de l'architecture, lorsque ces paramètres se situent ailleurs ou au niveau de l'attention. Le modèle *Linformer* par exemple, exploite des matrices dont la taille dépend de la longueur t afin de factoriser la *softmax*. Or, ces dernières doivent être modifiées en fonction de la taille des séquences et être réinitialisées lors d'un entraînement à partir d'une attention traditionnelle. Une perte significative est dès lors automatique et le modèle doit à nouveau être entraîné pour combler ce déficit de performances.

2. Le coût d'un entraînement de BERT-base est de l'ordre de 1000 heures-GPU sur des Nvidia V100.

La dernière caractéristique concerne la conservation des performances à des longueurs de séquences plus grandes : c'est la capacité d'*extrapolation*. Ce principe pourtant essentiel n'est pas respecté par l'attention vanille. BERT et RoBERTa montrent des pertes très importantes et proportionnelles à la taille des entrées. Ainsi, un modèle entraîné sur une longueur maximale de 512 sera inutilisable sur des tailles supérieures. Cette problématique peut remettre en cause la volonté d'approximer la *softmax* comme le fait *Performer*. En réalité, seule une partie de la matrice des scores est utile et exploitable, le reste agissant souvent comme du bruit. Ainsi, les modèles éparses suivant un schéma de sélection fixe ou aléatoire tendent à conserver davantage d'informations comparés aux autres approches puisqu'ils se basent généralement sur une attention locale. À noter qu'il existe certaines techniques qui favorisent l'extrapolation, notamment la suppression des tokens de fin de séquence (EOS) [Newman et al., 2020, Csordás et al., 2021], l'utilisation d'une attention de position [Dubois et al., 2020] et toutes les méthodes favorisant les connexions locales (positions relatives, constantes dans les scores...).

Le respect de ces trois caractéristiques, conservation des performances du modèle initial, architecture indépendante de la longueur et extrapolation peut facilement être observé en pratique. Ces principes sont respectés par *Longformer* et *Big Bird* mais omis dans la plupart des contributions modernes.

4.1.1.2 Taxonomie des modèles

Les modèles efficients se regroupent principalement en quatre catégories, chacune ayant ses propres avantages :

— les approches à base de récurrence qui traitent les entrées par blocs et les connectent par la suite grâce à divers mécanismes ;
— les approches à base de kernelisation ou de factorisation permettant de commuter les opérations dans le premier cas et de réduire la complexité et le coût en mémoire du stockage de la matrice des scores dans le second ;
— les approches à base de clustering dont l'objectif est de regrouper des requêtes et des clés semblables afin de réduire le nombre de connexions par token ;

— les approches par schémas fixes et aléatoires dans lesquelles certaines connexions sont assurées (locales) tandis que d'autres peuvent être choisies aléatoirement ou par le biais d'une métrique spécifique.

Récurrence Les modèles à base de blocs récurrents sont les plus anciens et apparaissent avec les *Transformers-XL* [Dai et al., 2019] dans lequels des segments de 512 éléments sont calculés puis connectés à l'aide d'un mécanisme de récurrence. Pour un nombre de segments τ, l'attention pour chaque tête au segment $s \in [1, \ldots, \tau]$ et à la couche $l \in [1, ..., n_l]$ notée $\boldsymbol{Y}_l^s$ est calculée selon :

$$\begin{cases} \tilde{\boldsymbol{Y}}_{l-1}^s &= [SG(\boldsymbol{Y}_{l-1}^{s-1}), \boldsymbol{Y}_{l-1}^s] \\ \boldsymbol{Q}_l^s &= \boldsymbol{Y}_{l-1}^s \boldsymbol{W}_q \\ \boldsymbol{K}_l^s &= \tilde{\boldsymbol{Y}}_{l-1}^s \boldsymbol{W}_q \\ \boldsymbol{V}_l^s &= \tilde{\boldsymbol{Y}}_{l-1}^s \boldsymbol{W}_v \\ \boldsymbol{Y}_l^s &= Attn(\boldsymbol{Q}_l^s, \boldsymbol{K}_l^s, \boldsymbol{V}_l^s) \end{cases} \tag{4.2}$$

La fonction *Stop-Gradient* notée $SG(.)$ empêche la propagation du gradient pour la matrice concernée. Cette architecture permet d'obtenir une complexité linéaire en fonction du nombre de segments mais nécessite tout de même de stocker de grandes matrices de scores puisque $\tilde{\boldsymbol{Y}}_{l-1}^s$ est la concatenation de deux matrices sur l'axe de la longueur. Le modèle *XLNet* [Yang et al., 2019] se base principalement sur ce type d'attention pour traiter des séquences plus longues.

Compressive Transformers [Rae et al., 2019] généralise le modèle précédent en utilisant une mémoire plus sophistiquée. Plutôt que de conserver uniquement le segment précédent, cette architecture exploite une seconde mémoire dans laquelle sont compressés les segments les plus anciens grâce à diverses fonctions comme du pooling, des convolutions dilatées ou la conservation des éléments ayant les scores les plus élevés uniquement. Afin de conserver une bonne compression, une seconde fonction de coût similaire à celle d'un auto-encoder est ajoutée et doit permettre au modèle d'apprendre à reconstruire la mémoire à partir de sa version compressée.

Kernelisation Les modèles à base de kernels reposent sur l'idée que la fonction de similarité utilisée dans l'attention pour construire la matrice des scores peut être remplacée afin d'exploiter les propriétés associatives du produit matriciel. Pour une requête (token, mot ou caractère) $\boldsymbol{q}_i$ extraite de $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_t]$, l'attention peut être reformulée comme :

$$\boldsymbol{y}_i = \frac{\sum_{j=1}^{\tau} sim(\boldsymbol{q}_i, \boldsymbol{k}_j)\boldsymbol{v}_j}{\sum_{j=1}^{\tau} sim(\boldsymbol{q}_i, \boldsymbol{k}_j)} \tag{4.3}$$

où $sim(.)$ est une fonction de similarité entre deux éléments et $\tau = t$ pour les modèles bidirectionnels. Dans l'attention vanille, $sim(\boldsymbol{q}_i, \boldsymbol{k}_j) = \exp(\boldsymbol{q}_i \boldsymbol{k}_j^\top)\sqrt{d_i}^{-1}$. L'objectif d'une approche par kernel est l'expression de cette fonction en tant qu'un produit de kernels notés $\phi(.)$:

$$\boldsymbol{y}_i = \frac{\sum_{j=1}^{\tau} \phi(\boldsymbol{q}_i)\phi(\boldsymbol{k}_j)^\top \boldsymbol{v}_j}{\sum_{j=1}^{\tau} \phi(\boldsymbol{q}_i)\phi(\boldsymbol{k}_j)^\top} \tag{4.4}$$

Dès lors, l'attention (équation 4.1) peut être plus généralement réécrite comme :

$$\boldsymbol{Y}_i = \Phi(\boldsymbol{Q}_i)\Phi(\boldsymbol{K}_i)^\top \boldsymbol{V}_i \tag{4.5}$$

Cette forme a l'avantage indéniable de permettre le produit $\Phi(\boldsymbol{K}_i)^\top \boldsymbol{V}_i$ dans un premier temps et donc de passer outre le stockage de la matrice des scores $\Phi(\boldsymbol{Q}_i)\Phi(\boldsymbol{K}_i)^\top$. De plus, l'équation 4.4 fonctionne plus généralement pour les approches causales (non bidirectionnelles), lorsque le modèle n'a accès qu'aux tokens précédents. Dans ce cas $\tau_i \leq t$ et celui-ci dépend de la position de $\boldsymbol{q}_i$ dans la séquence [Katharopoulos et al., 2020].

Cette approche peut être réutilisée pour une fonction spécifique comme la similarité de cosinus. En posant simplement $\Phi(\boldsymbol{x}) = {}^{\boldsymbol{x}}/_{||\boldsymbol{x}||}$, est obtenue une variante qui peut se passer de normalisation puisque la métrique est bornée entre -1 et 1. Le modèle peut aussi être modifié pour assurer la positivité de tous les scores.

Le choix du kernel peut suivre une certaine logique et dépendre de certaines propriétés. Dans le modèle *Linear Attention* [Katharopoulos et al., 2020], les auteurs proposent des propriétés souhaitables de ϕ, notamment la positivité à l'aide de la fonction ELU ($\phi(x) = elu(x)+1$) [Clevert et al., 2016b] afin de rendre

la normalisation des scores cohérente. Ils montrent dans un second temps que le calcul des gradients peut être effectué en temps linéaire et à mémoire constante à l'aide de sommes cumulatives et d'un algorithme spécifique. Enfin, les auteurs comparent les performances de leur approche au modèle vanille et à *Reformer* et montrent des performances compétitives. Cependant la comparaison n'est effectuée qu'en traitement d'images.

Dans le modèle *Efficient Attention* [Shen et al., 2020], les auteurs choisissent $\Phi(.) = Softmax(.)$ afin de répliquer un comportement semblable à l'attention vanille [3]. L'expérience est menée en comparaison de l'approche par produit simple (sans *softmax*). Les auteurs concluent que la méthode de normalisation a peu d'effet sur les performances mais ils ne présentent que des expériences en traitement d'images.

Le modèle *Performer* [Choromanski et al., 2021] étend l'utilisation de kernels en proposant un mécanisme d'*attention généralisée*. Celui-ci permet d'obtenir un estimateur sans biais de l'attention vanille à l'aide de kernels aléatoires. Pour cela, les auteurs reformulent l'équation 4.1 :

$$\boldsymbol{Y}_i = \boldsymbol{D}_i^{-1}\boldsymbol{A}_i\boldsymbol{V}_i \quad \boldsymbol{A}_i = \phi(\boldsymbol{Q}_i)\phi(\boldsymbol{K}_i)^\top \quad \boldsymbol{D}_i = diag(\boldsymbol{A}_i\mathbf{1}_t)$$

Les kernels aléatoires $\phi(.)$ sont définis par :

$$\phi(\boldsymbol{x}) = \frac{h(\boldsymbol{x})}{\sqrt{m}}\Big[f_1(\boldsymbol{x}\boldsymbol{w}_1), \ldots, f_1(\boldsymbol{x}\boldsymbol{w}_m), \ldots, f_k(\boldsymbol{x}\boldsymbol{w}_1), \ldots, f_k(\boldsymbol{x}\boldsymbol{w}_m)\Big],$$

dans lesquels les vecteurs $\boldsymbol{w}_j \in \mathbb{R}^d, \forall j \in [1, \ldots, m]$ sont tirés aléatoirement d'une loi normale centrée et réduite. Les fonctions $h(.)$ et $[f_1(.), \ldots, f_k(.)]$ sont choisies spécifiquement pour approximer la *softmax*, avec $k = 2$:

$$\begin{cases} h(\boldsymbol{x}) & = \frac{1}{\sqrt{2}}\exp(-\frac{||\boldsymbol{x}||^2}{2}) \\ f_1(\boldsymbol{x}) & = \exp(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) & = \exp(-\boldsymbol{x}) \end{cases}$$

3. À noter que la fonction est appliquée sur l'axe t.

Afin de réduire la variance de cet estimateur et d'améliorer la qualité de l'approximation, les auteurs proposent d'utiliser des vecteurs aléatoires orthogonaux. Cette étape est effectuée en utilisant l'algorithme de Gram-Schmidt et permet de conserver l'absence de biais [Choromanski et al., 2017]. Le modèle *Performer* ne permet que d'approximer la *softmax*, les erreurs s'accumulant à chaque couche, il est donc nécessaire de réentraîner le modèle à partir d'un checkpoint BERT ou RoBERTa durant au moins 50K étapes pour n'importe quelle longueur. Cependant, le modèle est très rapide et a un faible coût mémoire dans toutes les configurations.

Factorisation Les modèles à base de factorisation cherchent à réduire la taille de la matrice de scores en projetant $\boldsymbol{K}$ et $\boldsymbol{V}$.

Linformer [Wang et al., 2020b] applique cette méthode en modifiant l'équation 4.1 et en intégrant par tête deux nouvelles matrices de paramètres $\boldsymbol{E}_i, \boldsymbol{F}_i \in \mathbb{R}^{t \times t'}, \forall i \in [1, \ldots, h]$ qui dépendent de la longueur t de la séquence avec $t > t'$:

$$\boldsymbol{Y}_i = Softmax\Big(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^\top \boldsymbol{E}_i}{\sqrt{d_i}}\Big) \boldsymbol{F}_i^\top \boldsymbol{V}_i$$

Dans ces conditions la matrice de scores passe d'une taille $t \times t$ à $t \times t'$, réduisant largement le coût mémoire et la complexité globale. La valeur de t' est généralement comprise entre 128 et 512, en fonction du degré de compression choisi. Les auteurs proposent des variantes sur ces modèles, notamment sur le choix du partage des paramètres des nouvelles matrices entre têtes, entre les clés et les valeurs et en partageant une unique matrice $\boldsymbol{E}$ partout. Ils observent de légères améliorations grâce à ce mécanisme. *Linformer* souffre d'un problème de dépendance à la longueur de la séquence, il est nécessaire de recalibrer le modèle si la taille des entrées augmente puisque ces paramètres sont soit absents du modèle, soit n'ont pas été entraînés. Le second défaut réside dans la difficulté d'exploiter un checkpoint connu, les matrices $\boldsymbol{E}$ et $\boldsymbol{F}$ modifient les clés et les valeurs dans l'attention car elles sont initialisées aléatoirement. Il est néanmoins possible de passer outre ces problèmes grâce à des moyens de réduction non paramétriques comme du pooling. Cependant, ces méthodes tendent à affecter significativement les performances sans un réentrainement adéquat.

Les *Synthétiseurs* [Tay et al., 2021] étudient l'attention en passant outre le produit $\boldsymbol{QK}^\top$. Pour une tête i, le modèle calcule une matrice $\boldsymbol{B}_i$ servant de matrice de scores :

$$\boldsymbol{Y}_i = Softmax\big(\boldsymbol{B_i}\big)G(\boldsymbol{X}_i)$$

Dans le modèle initial, $\boldsymbol{B}_i = \sigma(\boldsymbol{X}_i\boldsymbol{W}_i^1)\boldsymbol{W}_i^2$ avec σ une fonction ReLU et $\boldsymbol{W}_i^1 \in \mathbb{R}^{d_i \times d_i}$ et $\boldsymbol{W}_i^2 \in \mathbb{R}^{d_i \times t}$ des matrices de paramètres. La fonction $G(.)$ est choisie arbitrairement et ne modifie pas la taille de $\boldsymbol{X}_i$. Le modèle admet des variantes en remplaçant $\boldsymbol{B}_i$ par un produit de matrices aléatoires. En pratique, cette approche ne permet pas de se passer du stockage complet des scores et le nombre de paramètres dépend de la taille de la séquence. Enfin, il est impossible d'exploiter le checkpoint d'un modèle RoBERTa ou autre dérivé de BERT puisque $\boldsymbol{Q}$, $\boldsymbol{K}$ et $\boldsymbol{V}$ ne sont pas calculées.

Clustering Les modèles à base de clustering ont pour objectif de regrouper les éléments semblables entre eux afin de limiter le nombre de connexions et par conséquent d'opérations à effectuer.

Plutôt que de calculer la matrice de scores entre chaque requête et chaque clé, le modèle *Clustered Attention* [Vyas et al., 2020] regroupe les requêtes en un nombre n_c de clusters et calcule les centroïdes stockés dans une matrice $Q_i^c \in \mathbb{R}^{n_c \times d}$ pour chaque tête $i \in [1, \ldots, h]$. L'équation 4.1 est donc modifiée :

$$\boldsymbol{Y}_i^c = Softmax\Big(\frac{\boldsymbol{Q}_i^c\boldsymbol{K}_i^\top}{\sqrt{d_i}}\Big)\boldsymbol{V}_i = \boldsymbol{A}_i^c\boldsymbol{V}_i$$

Puisque seuls les centroïdes sont évalués à l'issue de l'opération, les requêtes appartenant au même cluster obtiennent la même représentation. Comme ce clustering est effectué tête par tête et que les représentations sont concaténées à la fin, il est peu probable que deux sorties soient identiques. À noter que l'algorithme utilisé est basé dans un premier temps sur l'algorithme LSH (*Locality-Sensitive Hashing*) sur lequel sont appliqués ensuite les *K-means* afin de réduire la complexité pour les très longues séquences. Les auteurs ont ensuite raffiné leur approche pour améliorer la qualité des prévisions :

— dans chaque cluster, les k clés ayant la plus forte pondération sont extraites de la matrice $\boldsymbol{A}_i^c \in \mathbb{R}^{n_c \times t}$;

— dans chaque cluster, les scores d'attention sont calculés entre toutes les requêtes et les *top-k* clés extraites ;
— puisque ces nouveaux scores ont une somme égale à 1, ces derniers sont dénormalisés, fusionnés et repondérés avec la matrice $\boldsymbol{A}_i^c \in \mathbb{R}^{n_c \times t}$.

À l'issue de ce processus, chaque requête possède de l'information issue du centroïde de son cluster et de l'information intra-cluster plus précise.

Le modèle *Reformer* [Kitaev et al., 2020] se base sur un processus analogue en exploitant exclusivement l'algorithme LSH afin d'effectuer un clustering efficace en grande dimension. Puisque dans une fonction *softmax* les éléments les plus grands dominent largement les autres du fait de l'exponentielle, il parait cohérent de se focaliser uniquement sur les k clés les plus proches pour une requête donnée. Pour cela, les auteurs se passent du principe de centroïde et privilégient à la place une fonction de hachage permettant de trier la séquence selon des degrés de similarité. Celle-ci est ensuite découpée en segments dans lesquels sont calculés les scores d'attention entre tous les éléments du segment actuel et du précédent. Le processus est proche d'une attention par blocs dont l'ordre des éléments a été affecté comme illustré dans la Figure 4.2.

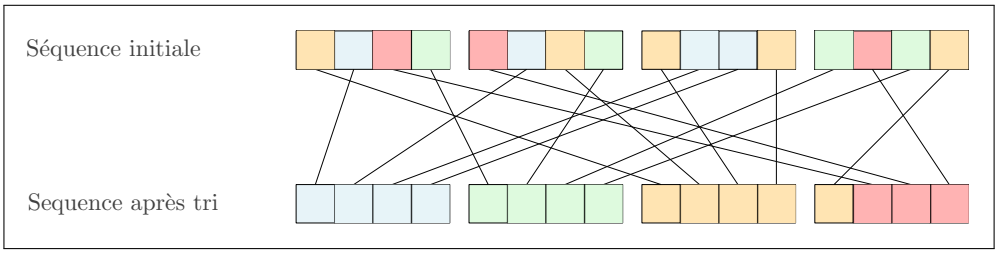


FIGURE 4.2 – Réorganisation de la séquence après hachage.

À noter que cette étape est effectuée plusieurs fois afin d'améliorer la qualité du clustering. Le modèle *Reformer* incorpore le principe de couche inversible [Gomez et al., 2017] : les activations de chaque couche sont reconstruites à partir des activations de la couche suivante, permettant ainsi d'effectuer la rétropropagation sans nécessairement stocker les activations en mémoire. Plus la profondeur du réseau est grande, plus le gain en mémoire est élevé. Le modèle opère enfin au niveau de certaines couches (hors attention), un découpage des entrées lorsqu'un stockage successif de grandes matrices est nécessaire.

Enfin, la variante *Routing Transformers* [Roy et al., 2020] propose une attention éparse basée sur une version des *k-moyennes* en ligne [Bottou and Bengio, 1995]. Pour cela, les auteurs exploitent pour chaque tête $i \in [1, \dots, h]$ une matrice de routage $\boldsymbol{R}_i \in \mathbb{R}^{t \times d}$ à l'aide d'une matrice de projection $\boldsymbol{W}_i^R \in \mathbb{R}^{d \times d}$:

$$\boldsymbol{R}_i = \boldsymbol{Q}_i \boldsymbol{W}_i^R + \boldsymbol{K}_i \boldsymbol{W}_i^R \tag{4.6}$$

L'algorithme des *k-moyennes* est appliqué sur la matrice $\boldsymbol{R}$ afin d'obtenir un ensemble de centroïdes appris durant l'entraînement. Le modèle utilise $\sqrt{t}$ centroïdes, calcule la distance de chaque token avec ces derniers et sélectionne les k tokens les plus proches afin d'assurer un même nombre d'éléments par cluster. Enfin, une attention intra-cluster est calculée, donnant la structure éparse du modèle.

Schémas fixes et aléatoires Ces modèles se basent sur l'utilisation d'un schéma à suivre durant le calcul des scores afin de réduire significativement la complexité de l'attention. Ce schéma est généralement prédéfini mais peut faire intervenir des éléments aléatoires afin de couvrir davantage de connexions.

Les premiers modèles à utiliser une forme de schéma se basent principalement sur la notion de blocs [Qiu et al., 2020]. Plutôt que de calculer tous les scores, l'idée est de se limiter à un certain entourage. Pour cela, les matrices $\boldsymbol{Q}$, $\boldsymbol{K}$ et $\boldsymbol{V}$ sont découpées en b segments de longueur fixe $t' = t/b$ appelés blocs. Le calcul de l'attention est dès lors effectué à l'intérieur des blocs. Pour une tête $i \in [1, \dots, h]$ et un bloc $j \in [1, \dots, b]$:

$$\boldsymbol{Y}_i^j = Softmax\Big(\frac{\boldsymbol{Q}_i^j \boldsymbol{K}_i^{j\top}}{\sqrt{d_i}}\Big)\boldsymbol{V}_i^j \tag{4.7}$$

La matrice de scores est dès lors modifiée, d'une taille $t \times t$, elle est désormais de taille $b \times t' \times t' = t \times t'$. Le gain en mémoire est donc d'un facteur b. Cependant, la taille des segments agit fortement sur les performances et, l'asymétrie des connexions, notamment sur le bord des blocs, peut largement réduire les capacités du modèle.

Cette première tentative, forçant le réseau à se focaliser sur les éléments proches entre eux est approfondie par le modèle *Longformer* [Beltagy et al.,

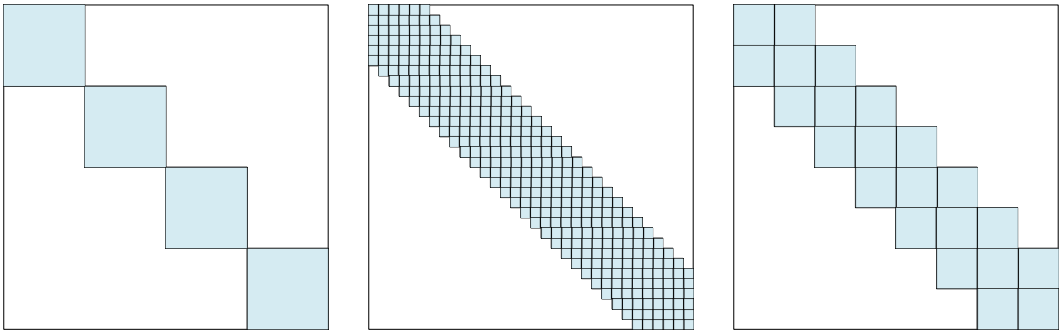
2020] qui met en place le principe d'attention locale. Plutôt que de travailler par blocs, celui-ci fait coulisser une fenêtre de taille prédéfinie afin que chaque élément de la séquence puisse accéder symétriquement aux k éléments précédents et aux k éléments suivants. Bien que simple dans l'idée, l'attention locale est difficile à mettre en place et requiert l'écriture de kernels CUDA spécifiques pour rendre l'opération efficiente. Le modèle *Longformer* ajoute des connexions supplémentaires prédéfinies qui améliorent significativement les performances des modèles :

— chaque élément de la séquence possède aussi une connexion envers les tokens spéciaux tels que [SEP] et [CLS] dans BERT ;
— les tokens spéciaux ont une attention globale et sont donc connectés à tous les éléments de la séquence ;
— durant le fine-tuning, des tokens globaux peuvent être ajoutés en fonction de la tâche afin de faciliter le transfert de l'information, ces derniers bénéficiant de matrices de projection $\tilde{\boldsymbol{W}}_q$, $\tilde{\boldsymbol{W}}_k$ et $\tilde{\boldsymbol{W}}_v$ propres mais initialisées sur celles existantes.

En pratique, *Longformer* a une grande capacité à conserver les performances d'un modèle déjà entraîné sans passer par une longue phase de réapprentissage. Les auteurs montrent notamment que la copie de l'embedding de position a un impact majeur sur cette conservation. Quant aux différentes tâches, le modèle montre des performances compétitives sans faire appel à des mécanismes complexes tels que présentés dans *Reformer* ou *Performer*.

Le fait que l'attention locale soit difficile à mettre en place, et parfois sous efficiente à cause de problèmes de mémoire non contiguë, a fait émerger une nouvelle approche par blocs afin d'approximer cette dernière (Figure 4.3).

Le modèle ETC (*Extended Transformers Construction*) [Ainslie et al., 2020] propose de décomposer les entrées en deux séquences, une locale et une globale. L'attention est alors calculée selon 4 schémas : globale vers locale (attention globale), globale vers globale, locale vers globale et locale vers locale (attention locale par blocs). Le modèle intègre un embedding de positions relatives de la



Attention par blocs | Attention locale | Attention locale par blocs

FIGURE 4.3 – Attention locale dans la matrice des scores.

forme la plus simple dans lequel est encodé une distance bornée [4] injectée directement dans l'attention. Enfin, la fonction de coût traditionnelle (MLM) est remplacée par la CPC (*Contrastive Predictive Coding*) [van den Oord et al., 2019] dont l'objectif est de prédire les représentations latentes de blocs de tokens. Contrairement à *Longformer*, les tokens globaux sont entraînés dès le départ et non pas ajoutés durant le fine-tuning sur une tâche annexe. ETC permet de conserver une partie des caractéristiques de *Longformer*, c'est-à-dire l'adaptation rapide à partir d'un checkpoint et la généralisation à n'importe quelle taille de séquence tout en améliorant les performances sur plusieurs benchmarks.

Le modèle *Big Bird* [Zaheer et al., 2020] reprend en partie l'idée du modèle ETC en ajoutant une subtilité. En plus des quatre types d'attention, sont ajoutées des connexions à des blocs choisis aléatoirement. Les éléments locaux ont des connexions :

— sur le voisinage à l'aide de l'attention locale par blocs (3 blocs) ;
— sur des blocs choisis aléatoirement (3 blocs) ;
— sur les tokens considérés comme globaux (2 blocs) ;
— sur les tokens spéciaux pour la version ETC de *Big Bird* (optionnel).

Bien que le modèle soit très performant, en particulier lorsqu'il est initialisé de RoBERTa ou Pegasus [Zhang et al., 2020], peu d'informations sont données sur la vitesse de convergence du modèle du fait des connexions aléatoires. Les auteurs précisent aussi que le fait de se passer de connexions globales, c'est-à-dire de se restreindre aux approches locales et aléatoires, ne permet pas d'être

4. $d(i,j) = clip(i-j, \alpha, \beta)$ où α et β sont des bornes choisies.

compétitif par rapport à BERT. Enfin, les auteurs montrent que l'hyperparamétrage est important dans la résolution de tâches annexes, la taille des blocs et le nombre de tokens globaux doivent être choisis et testés préalablement.

PoolingFormer [Zhang et al., 2021] se focalise principalement sur l'aspect local en y ajoutant du pooling. Contrairement aux modèles précédents, celui-ci exploite une attention à deux niveaux dans laquelle l'estimation est effectuée en deux temps. L'attention locale est calculée deux fois avec deux fenêtres différentes, le résultat de la première couche servant d'entrée dans la seconde après la projection à l'aide de nouvelles matrices $\tilde{\boldsymbol{W}}_q$, $\tilde{\boldsymbol{W}}_k$ et $\tilde{\boldsymbol{W}}_v$. Puisque le deuxième niveau a pour objectif d'étendre le contexte, la séquence est préalablement compressée par le biais de pooling (moyen, max...). Ce mécanisme permet donc d'élargir la bande diagonale de la matrice des scores exploitée tout en gardant un nombre de connexions relativement faible. Ce modèle, bien que très performant sur certaines tâches, nécessite l'estimation de paramètres supplémentaires, rendant les comparaisons parfois discutables. De plus, le calcul en deux temps ralentit la vitesse générale de cette architecture et complexifie l'adaptation à partir d'un checkpoint existant.

4.1.2 Nouveaux modèles alternatifs

Cette section présente un nouveau modèle d'attention efficiente ayant de fortes capacités d'adaptation et d'extrapolation grâce un mélange d'attention locale, éparse et globale. [5]

4.1.2.1 Modèle Local-Sparse-Global

La section précédente a montré l'existence d'un nombre important de modèles, chacun suivant des caractéristiques uniques. Les architectures capables d'adaptation et d'extrapolation se retrouvent majoritairement dans les approches à schéma fixe car l'attention locale couplée à la duplication de l'embedding de position permet de conserver une grande part des performances d'un

5. L'architecture et les poids sont disponibles sur le hub d'*HuggingFace* pour la version anglaise du modèle, voir https://huggingface.co/ccdv/lsg-base-4096.

modèle pré-entrainé. Cependant, les auteurs de ces modèles font parfois des choix discutables pouvant entacher leurs qualités : *Longformer* se base presque exclusivement sur l'approche locale et rend plus difficile la prise en compte d'un contexte élargi malgré les connexions globales, il faut donc un nombre important de couches pour que l'information transite entre éléments éloignés. *Big Bird* quant à lui, qui se base en partie sur de l'attention aléatoire, peut être critiqué sur ce dernier aspect puisque l'évaluation est conditionnée au tirage des blocs. Ce modèle doit être entraîné plus longtemps que ses homologues afin de limiter la variance importante des résultats et d'assurer sa convergence pour résoudre les tâches.

L'architecture présentée que nous nommerons LSG (*Local-Sparse-Global*) tente de répondre à ces problématiques en proposant une attention éparse élargissant le contexte en exploitant principalement des clés plus susceptibles d'être fortement pondérées. L'intuition derrière ce modèle est simple :

— l'attention vanille utilisée dans BERT et RoBERTa ne permet pas d'extrapoler à des séquences plus longues sans une perte significative de performances ;
— ces architectures surpondèrent systématiquement la bande diagonale de la matrice des scores ;
— les éléments distants partagent généralement peu d'information.

À la vue de ces éléments, on peut supposer que localement, une information bas niveau est importante (un accès à tous les tokens proches) tandis que globalement, une information haut niveau suffit (capturer un thème par exemple). Il parait donc important d'exploiter l'attention locale parallèlement à une attention éparse qui sélectionne des éléments éloignés, susceptibles d'apporter une information complémentaire.

Les techniques les plus utilisées pour compresser l'information à moindre coût sans dépendre de la taille de la séquence sont à base de pooling. Cependant, ces dernières peuvent avoir une incidence significative. Par exemple, le *max-pooling* effectué sur l'axe de la longueur pour filtrer les clés, tend à rendre les modèles instables et à destructurer les entrées puisqu'il recompose l'embedding en sélectionnant les éléments indépendamment pour chaque dimension de l'embedding. Le *pooling-moyen*, plus naturel, rend difficile la discrimination des tokens à cause du lissage.

Le modèle que nous proposons exploite trois variantes de pooling : la première permettant d'extraire les clés les plus susceptibles d'être fortement pondérées, la seconde utilisant du pooling moyen et la dernière à base de clustering afin de construire des centroïdes. Chaque architecture est basée sur trois éléments, l'attention locale par blocs, une attention éparse avec les mécanismes présentés ci-dessus et une attention globale qui exploite des tokens spéciaux ajoutés à la séquence. Contrairement à la plupart des modèles et plus particulièrement *Big Bird*, ces tokens spéciaux ne sont pas choisis dans la séquence initiale mais concaténés. Leur utilisation et leur sélection est donc indépendante des données en entrée, il est donc possible de les entraîner sans toucher aux autres paramètres. Cela limite l'effet de sur-apprentissage pour les tâches avec peu de données puisque le réseau peut être figé. À noter que pour chaque architecture, la sélection des éléments et le pooling sont appliqués indépendamment sur chaque tête afin de démultiplier les schémas de sélection sur chaque couche.

Sélection des clés d'intérêt Pour une matrice de scores $\boldsymbol{S} \in]0,1[^{t\times t}$, la clé $\boldsymbol{k}_a$ est plus importante que la clé $\boldsymbol{k}_b$ si celle-ci est en moyenne surpondérée dans la séquence :

$$\frac{1}{t}\sum_{k=1}^{k=t} S_{k,a} > \frac{1}{t}\sum_{k=1}^{k=t} S_{k,b} \quad \forall a, b \in [1,\dots,t] \text{ et } a \neq b$$

Cette matrice étant inconnue, il n'est pas possible de connaître les éléments surpondérés a priori. Le problème peut être réécrit comme la recherche d'un vecteur de pondération $\boldsymbol{\alpha} = [\alpha_1,\dots,\alpha_t] \in \mathbb{R}^t$ qui est lui aussi inconnu :

$$\Big[\sum_{k=1}^{k=t} \alpha_k \boldsymbol{q}_k\Big]\boldsymbol{k}_a^\top > \Big[\sum_{k=1}^{k=t} \alpha_k \boldsymbol{q}_k\Big]\boldsymbol{k}_b^\top \quad \forall a, b \in [1,\dots,t] \text{ et } a \neq b$$

Plusieurs pistes de résolution peuvent être envisagées. La première concerne la résolution de problématiques MIPS (*Maximum Inner Product Search*) consistant à déterminer de façon approximative ou exacte la ou les clés les plus proches d'une requête. Ainsi, pour une requête $\boldsymbol{q} \in \mathbb{R}^{1\times d}$ et un ensemble de

clés $\mathcal{K} \subset \mathbb{R}^{1\times d}$:

$$p = \arg\max_{\boldsymbol{k}\in\mathcal{K}} \boldsymbol{q}\boldsymbol{k}^\top$$

Ce problème peut être résolu efficacement par l'utilisation d'une LSH asymétrique [Shrivastava and Li, 2014] ou symétrique [Neyshabur and Srebro, 2015, Yan et al., 2018]. Cependant cette résolution s'applique principalement dans les cas où le nombre de requêtes est relativement faible face à un nombre de clés très grand. En pratique, l'application de cette stratégie est peu efficiente dans le cadre d'un Transformer puisque l'algorithme requiert l'évaluation d'une fonction distance suite à une fonction de hachage, et doit s'appliquer sur chaque requête de chaque tête et de chaque couche. Pour un modèle standard et des séquences de 4096 éléments, cela représente un demi-million de requêtes à effectuer pour un seul document en entrée.

La connaissance des k éléments les plus proches de chaque requête nécessite le stockage de $t \times k$ clés ce qui n'est pas toujours possible en pratique. De plus, le calcul de l'attention est plus difficile puisque l'approche par blocs ne peut plus être appliquée. Pour réduire la complexité, il est donc nécessaire de déterminer les k éléments les plus discriminants en moyenne pour toute la séquence.

La seconde piste consiste à utiliser une métrique simple à calculer permettant d'augmenter les chances de sélectionner de bonnes clés. Puisque la fonction *softmax* permet aux éléments les plus grands de largement dominer les autres, la distribution de la matrice de scores est modifiée et devient asymétrique du fait de l'exponentielle (Figure 4.4). Cet effet est d'autant plus fort que la distribution initiale est décentrée ou à forte variance. Étant donnée la nature de la *softmax*, il est nécessaire de trouver les cas pour lesquels le produit entre la clé et la requête a une forte amplitude. La méthode la plus simple est d'exploiter la norme de la requête :

$$\boldsymbol{q}\boldsymbol{k}^\top = \cos(\theta)\|\boldsymbol{q}\|\|\boldsymbol{k}\|$$

Puisque sont recherchées des clés indépendamment des requêtes, le terme $\|\boldsymbol{q}\|$ n'a pas d'influence. Cependant, le signe de $cos(\theta)$ est quant à lui inconnu. Si la norme de la clé est importante et que le cosinus est positif, celle-ci a de fortes chances de dominer la *softmax*.

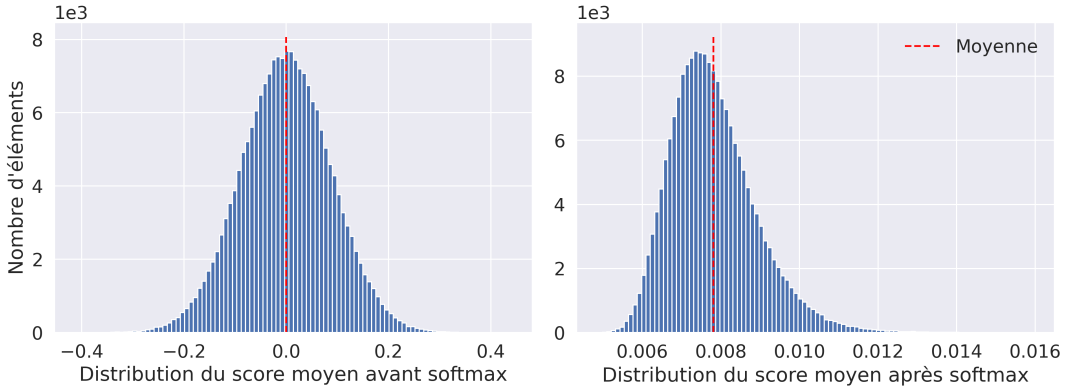


FIGURE 4.4 – Distribution des scores moyens avant et après softmax.

Puisque le calcul de la matrice des scores dépend d'une exponentielle, l'effet de la norme est d'autant plus fort. En pratique, on observe généralement une concentration des normes autour de valeurs centrales pour lesquelles le paramètre θ joue un rôle important. On observe en parallèle une présence de normes beaucoup plus élevées, notamment chez certains tokens spéciaux (début de phrase). Ces éléments dominent très fortement la *softmax* et servent principalement à décharger une partie des poids afin de sous-pondérer le reste de la séquence. La relation entre poids moyen et norme observée est présentée Figure 4.5. Pour le second graphique, les clés sont triées en fonction de leur norme

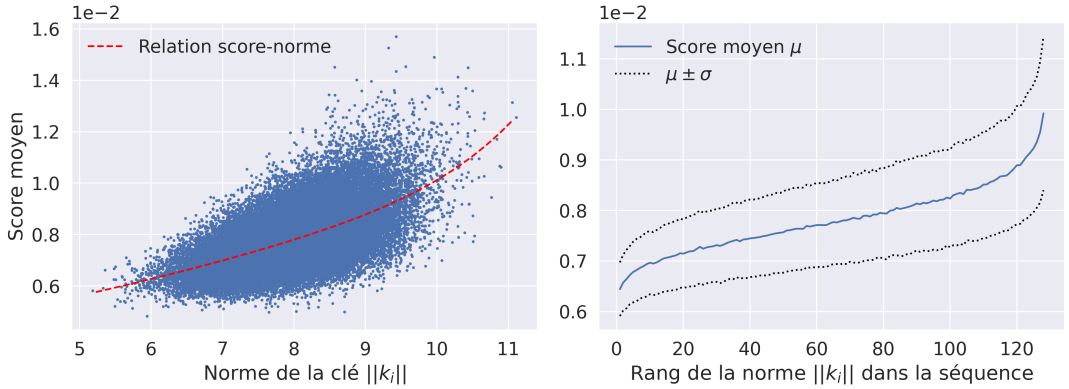


FIGURE 4.5 – Relations entre les scores moyens et les normes.

sur plusieurs séquences puis le score moyen est calculé en fonction du rang. Le critère de la norme permet de discriminer à moindre coût un ensemble de clés

distantes. À noter que la sélection se fait par tête et donc la sélection peut être très hétérogène de l'une à l'autre.

Approches par centroïdes Les deux autres variantes exploitent des centroïdes afin de compresser les éléments éloignés. La première approche se base sur du pooling moyen simple sur l'axe de la longueur tandis que la seconde fait appel à des techniques de clustering. Les architectures à base de clustering présentées dans *Reformer* ou *Clustered Attention* ont des paradigmes différents. La première suggère que le calcul de l'attention doit être cantonné aux éléments similaires. La seconde approche fait l'hypothèse que les requêtes similaires peuvent être regroupées et représentées par le biais de centroïdes.

Le modèle que nous proposons se base sur une troisième idée selon laquelle des clés similaires peuvent être regroupées dès lors qu'elles se situent en dehors de la fenêtre locale. Un autre élément important concerne le traitement non global. Plutôt que de regrouper des éléments issus de la séquence complète, celle-ci est d'abord découpée en sous-blocs et le clustering est effectué indépendamment dans chacun d'eux et sur chaque tête. Cela permet de faire une distinction entre information passée et information future, et d'adapter facilement le modèle aux approches causales contrairement aux autres architectures. Le calcul des centroïdes est effectué par les *k-moyennes* grâce à l'algorithme de Lloyd. Puisqu'il est nécessaire d'appliquer ce principe sur chaque couche, les centroïdes sont initialisés par LSH et seules 8 étapes d'optimisation sont effectuées. Contrairement aux deux autres variantes, celle-ci n'est pas déterministe mais reste bien moins variable qu'une sélection purement aléatoire comme dans *Big Bird*. Pour un nombre de centroïdes n_c, les indices sont calculés à l'aide d'une matrice de projection tirée d'une loi normale centrée réduite $\boldsymbol{R}_i \in \mathbb{R}^{d_i \times s}$ où $s = \log_2(n_c)$ est le nombre de bits et $\boldsymbol{p} = [2^0, 2^1, \ldots, 2^{s-1}] \in \mathbb{N}^s$ est un vecteur des puissances de 2. Pour une matrice de clés $\boldsymbol{K}_i \in \mathbb{R}^{t \times d_i}$ issue de la tête i :

$$\boldsymbol{c}_i = \mathcal{H}\big(\boldsymbol{K}_i \boldsymbol{R}_i\big)\boldsymbol{p} \tag{4.8}$$

où $\mathcal{H}(.)$ est la fonction de Heaviside et $\boldsymbol{c}_i$ le vecteur des indices d'appartenance aux clusters. Les centroïdes sont initialisés en faisant la moyenne des éléments qui appartiennent à chaque cluster. Le contexte de chaque requête pour le calcul

de l'attention est composé des éléments locaux et des clusters des blocs les plus proches. Le modèle exploitant le pooling se base sur ce dernier principe mais ne nécessite pas d'algorithme spécifique.

Construction des modèles Les trois variantes ont une structure par blocs commune : 3 blocs servant pour l'attention locale et 2 pour l'attention éparse. Le choix de la taille des blocs b (64 ou 128) est un hyperparamètre qui a une incidence sur les performances du modèle, la vitesse d'exécution et la convergence générale. Le second hyperparamètre f permet de quantifier le taux d'éléments distants retenus. Pour $f = 4$, 25% des tokens seront sélectionnés en dehors du contexte local. À noter ici que la sélection est structurée : la séquence est préalablement subdivisée en $n_b = t/b$ blocs et chacun sélectionne selon l'exemple, 25% de ses éléments. Enfin, le dernier paramètre concerne le nombre de tokens globaux qui est généralement fixé à 1 pour le MLM et augmenté pour le fine-tuning. Seule la méthode de pooling change entre les trois variantes.

La Table 4.1, les Figures 4.6 et 4.7 comparent la structure de différents modèles et la construction de ces derniers, on notera que *Longformer* n'utilise pas de bloc mais une fenêtre d'attention généralement fixée à 512 éléments.

Connexions	Locales	Eparses	Globales	Total*	Contexte**
Longformer (w)	w	-	g	$w + g$	w
Block-Local (b)	$b \times 3$	-	-	$b \times 3$	$b \times 3$
Big Bird ITC (b)	$b \times 3$	$b \times 3$	$b \times 2$	$b \times 8$	t
LSG (b, f)	$b \times 3$	$b \times 2$	g	$b \times 5 + g$	$(3 + 2f) \times b$

* Nombre de connexions par requête, hors requêtes globales.

** Distance maximale théorique entre deux clés, hors connexions globales.

TABLE 4.1 – Connexions par modèle.

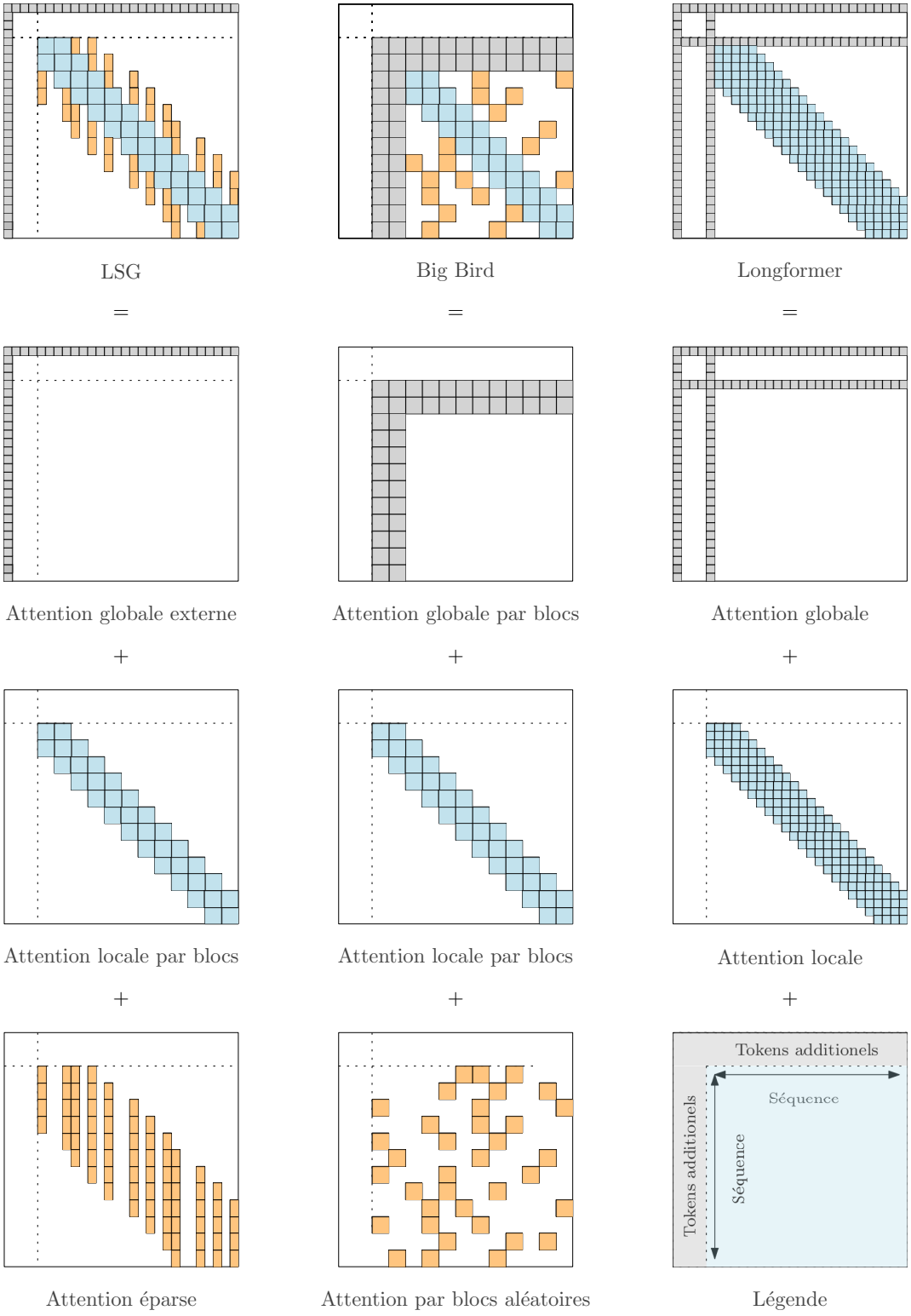


FIGURE 4.6 – Construction des matrices de scores de différents modèles.

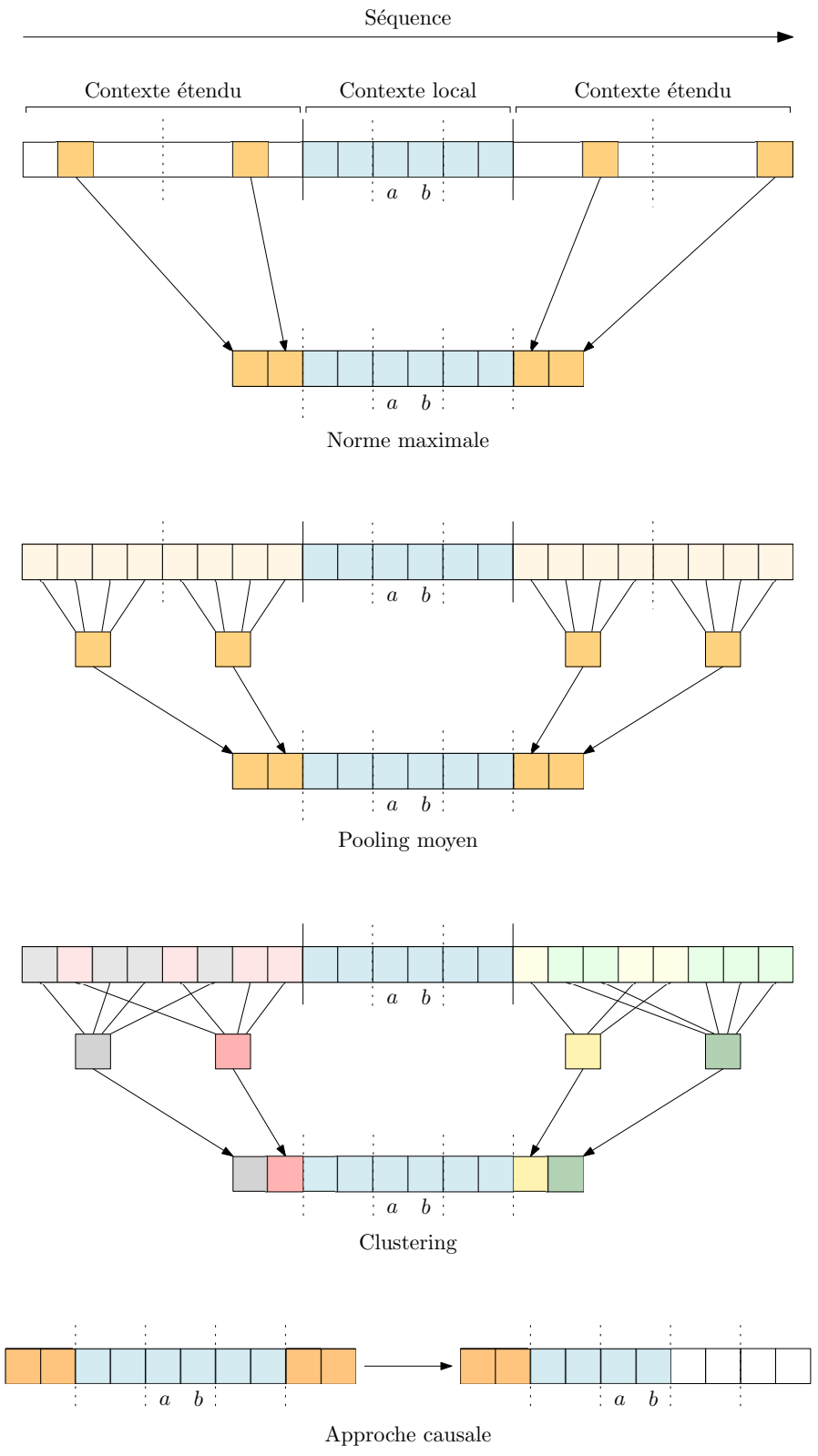


FIGURE 4.7 – Construction du contexte pour les éléments a et b (LSG).

La structure par blocs permet d'optimiser les approches causales, c'est-à-dire les modèles pour lesquels seuls les éléments passés sont disponibles. Dans la grande majorité des modèles, cette variante est effectuée par le biais d'un masque triangulaire permettant de masquer tous les éléments qui suivent. Cependant, cette technique n'empêche pas le modèle de calculer entièrement la matrice de scores entraînant ainsi une perte de ressources. Les approches LSG optimisent ce processus en supprimant les deux derniers blocs. Cela a deux conséquences : 40% des opérations sont supprimées et le masque triangulaire est limité au bloc central, la perte de ressources est donc minimisée. La méthode de construction des contextes est présentée dans la Figure 4.7. À noter que dans ces modèles chaque connexion n'est calculée qu'une fois.

La section suivante présente des expérimentations très générales sur les variantes efficientes des Transformers et montre l'intérêt de certaines approches par rapport à d'autres.

4.1.2.2 Analyse empirique en MLM

Les expérimentations restreintes suivantes permettent de mettre en lumière certaines caractéristiques omises par les auteurs, notamment la capacité d'adaptation et d'extrapolation dans un contexte où les modèles ont accès au checkpoint de RoBERTa. L'expérimentation est menée sur une structure commune et basée sur la bibliothèque *HuggingFace* : l'architecture RoBERTa est copiée et seul le module d'auto-attention [6] (*RobertaSelfAttention*) est modifié. Pour chaque modèle, les poids de toutes les matrices sont copiés dans l'attention, notamment les matrices de projection $\boldsymbol{W}^q$, $\boldsymbol{W}^k$ et $\boldsymbol{W}^v$. Les expérimentations sont menées sur du MLM qui est une tâche simple que le modèle est en mesure de résoudre et qui ne nécessite pas de recherche d'hyperparamètres. Les implémentations sont basées sur :

— le code d'*HuggingFace* lorsque disponible : *Reformer*, *Linformer*, *Longformer*, *Big Bird*, *Performer* [7] ;
— l'implémentation propre pour les autres modèles.

6. https://github.com/huggingface/Transformers/tree/master/src/Transformers/models
7. https://github.com/Muennighoff/Transformers/tree/master/src/Transformers/models

La première expérimentation reprend un jeu d'entraînement similaire à celui utilisé par Longformer copiant lui-même RoBERTa et composé de *Book Corpus* [Zhu et al., 2015], *Wikipedia*, *Realnews* [Zellers et al., 2019] et *Stories* [Trinh and Le, 2019]. Un échantillon de test de 5000 documents de longueurs variables est extrait dont 15% des éléments sont masqués. La métrique utilisée est la BPC, pour un token, un nombre de classes c (vocabulaire), un modèle f qui estime une probabilité et une étiquette y :

$$BPC = -\frac{1}{\log(2)} \sum_{i=1}^{c} y_i f_i(x)$$

Afin de s'assurer que l'échantillon est conforme aux autres papiers, RoBERTa-base est évalué sur ces données et donne une BPC de 1.881 contre 1.880 dans l'article original et 1.846 dans ceux de *Longformer* et *BigBird*. On peut donc conclure que le jeu d'entraînement est identique. L'expérimentation est menée en vérifiant la capacité des modèles à extrapoler à un contexte plus long. Dans un premier temps, chaque modèle effectue une passe sur l'échantillon de test avec des séquences de 512 puis celui-ci est entrainé pendant 250 étapes sur des séquences pouvant atteindre 4096 éléments. L'objectif est de vérifier si le modèle est en capacité d'exploiter l'information des poids de RoBERTa puis de vérifier s'il peut, grâce à un fine-tuning relativement court, se rapprocher des performances sur séquences plus courtes.

Chaque modèle est entrainé avec un taux d'apprentissage croissant linéairement de 0 à 1e-5 (warmup) durant toutes les étapes. La taille des batchs est fixée à 256 pour des séquences de 4096 éléments. La configuration du modèle est identique à RoBERTa-base, c'est-à-dire 12 couches, 12 têtes d'attention, embedding de taille 768, taille intermédiaire de 3072, dropout de 10% et 50265 mots de vocabulaire. Les résultats sont présentés en Table 4.2.

Longueur	512		4096		4096*	
Modèle	BPC	P.	BPC	P.	BPC	P.
Avg-Pooling	10.106	0.102	10.589	0.079	8.157	0.167
Max-Pooling	15.340	0.032	18.550	0.006	11.584	0.045
Linear Attn.	11.324	0.061	11.474	0.058	9.369	0.139
Efficient Attn.	21.022	0.102	20.574	0.097	9.954	0.133
Kernel-Cosinus	14.692	0.018	16.280	0.026	11.343	0.069
Performer	10.382	0.107	10.556	0.102	8.963	0.144
Linformer (128)	22.176	0.098	20.386	0.032	10.532	0.091
Reformer	17.602	0.003	18.608	0.002	6.861	0.217
Block Attn. (512)	1.881	0.732	2.039	0.709	1.854	0.717
Block-local Attn. (256)	1.881	0.732	2.018	0.713	1.832	0.721
Longformer (512)	1.929	0.726	2.051	0.708	1.826	0.719
Big Bird ITC (64)	1.881	0.732	2.439	0.659	2.056	0.684
LSG-Norme (128, 2)	1.895	0.729	**2.014**	**0.714**	**1.798**	**0.725**
LSG-Norme (64, 4)	1.968	0.719	2.092	0.701	1.844	0.717
LSG-Pooling (128, 2)	1.955	0.723	2.079	0.704	1.848	0.716
LSG-Cluster (128, 2)	1.945	0.724	2.064	0.705	1.824	0.720
RoBERTa	1.881	0.732	4.335	0.359	3.450	0.417

* Après 250 étapes d'entraînement.

TABLE 4.2 – BPC et précision des Transformers efficients.

À la vue des résultats, il est évident que seuls les modèles avec un schéma fixe ou aléatoire sont en mesure d'exploiter un checkpoint RoBERTa. C'est une caractéristique essentielle qui justifie presque totalement leur utilisation. Plus généralement, ces derniers sont compatibles avec n'importe quelle architecture

se basant sur l'attention vanille. À noter que les modèles d'attention par blocs, d'attention locale par blocs et *Big Bird* ont des performances égales à celles de RoBERTa (512) puisqu'ils couvrent tous les éléments de la séquence.

Pour les séquences longues, ce sont les deux modèles LSG et l'attention locale par blocs qui ont la meilleure faculté d'adaptation initiale. À noter que ce second modèle utilise une fenêtre de 256×3 tokens ce qui rend le contexte naturellement large avec un grand nombre de connexions. On remarque que *Big Bird* est contraint par son attention aléatoire et n'arrive pas à égaler les performances de LSG même après 250 étapes d'entraînement. La version LSG à base de normes converge plus vite que les autres, probablement du fait de gradients plus grands. Enfin, RoBERTa perd la moitié de sa précision en passant à des séquences de 4096, un réentraînement important est nécessaire pour espérer approcher les performances initiales.

Une seconde expérimentation est menée afin de mesurer la capacité de sur-apprentissage des modèles. La tâche est simple : prendre un jeu de données (*Wikitext-103*) pour lequel les architectures sont déjà performantes, puis effectuer plusieurs époques afin de vérifier que le modèle continue de converger convenablement en MLM. L'entraînement est effectué sur 3000 étapes avec des séquences de 4096, ce qui représente environ 7 époques. L'expérimentation est menée sur des batchs de 64, un taux d'apprentissage de 1e-4 avec une baisse linéaire jusqu'à 0. Les résultats sont présentés en Table 4.3.

	0 étapes		3000 étapes		Vitesse*	
	BPC	P.	BPC	P.	Entraînement	Inférence
Longformer	2.598	0.678	1.584	0.757	×1.12	×1.26
Big Bird ITC	3.665	0.585	1.671	0.747	×1.29	×1.41
LSG (128, 2)	**2.530**	**0.684**	**1.558**	**0.761**	×1.91	×1.67
LSG (64, 4)	2.668	0.667	1.584	0.757	×2.09	×1.80
LSG-Pooling	2.651	0.672	1.587	0.755	×1.93	×1.69
LSG-Cluster	2.658	0.672	1.592	0.753	×1.77	×1.51

* RoBERTa 4096 est utilisé comme référence.

TABLE 4.3 – Comparaisons des performances sur Wikitext-103 (3000 étapes).

Le modèle LSG est à nouveau plus performant que ses homologues tout en assurant une vitesse d'exécution accrue. Sa variante plus petite (blocs de 64) démarre plus haut que *Longformer* mais converge au même niveau à l'issue de l'entraînement. Ce modèle a pourtant la particularité de ne posséder que 321 connexions par token par rapport aux 513 de *Longformer*, soit une baisse de plus de 37%. Il est donc capable d'apprendre aussi bien avec moins d'information. *Big Bird* reste moins performant à cause de son attention aléatoire. Sa BPC de 1.671 est atteinte en 250 étapes par le modèle LSG(128). À noter que le coût en mémoire est similaire pour toutes les architectures hormis le petit modèle LSG qui bénéficie d'une utilisation mémoire 8% plus faible.

La troisième expérimentation reprend la base utilisée par *Longformer* pour effectuer un entraînement beaucoup plus long à partir d'un checkpoint RoBERTa. Sont reportées les performances de *Longformer*, LSG, *Big Bird* ITC et *Big Bird* ETC [8] qui remplace ici les blocs aléatoires par 128 tokens globaux supplémentaires. À noter que *Longformer* utilise 2^{18} tokens par étape (64×4096) tandis que *Big Bird* en fait passer 4 fois plus pour un taux d'apprentissage 4 fois plus élevé aussi. On peut donc considérer qu'une étape *Big Bird* vaut approximativement 4 étapes *Longformer*. LSG est entraîné sur des batchs de 2^{18} tokens aussi mais effectue l'entraînement en deux temps : 1 époque (15000 étapes) sur des séquences tronquées de 512 pour calibrer le modèle puis un fine-tuning de 10000 étapes sur les séquences de 4096 puisque le modèle a montré une bonne capacité d'extrapolation [9]. Enfin, il faut remarquer que l'échantillon de test est ici plus difficile puisque *Longformer* annonçait une BPC initiale de 1.957 dans le papier contre une de 2051 calculée (Table 4.2). Les résultats sont reportés en Table 4.4 et reprennent les performances annoncées par les auteurs.

8. Cette variante n'est pas disponible dans *HuggingFace*. Ce modèle du fait du nombre important de connexions globales (256) et d'une taille de blocs plus grand (84) est plus lent que sa version ITC.

9. Un entraînement sur des séquences de 4096 uniquement est plus lent mais permet d'atteindre une BPC inférieure à 1.70 en 10000 étapes.

	BPC initiale	BPC finale	Nombre d'étapes
Longformer	1.957	1.705	65000
Big Bird ITC (64)	-	1.678	N/C*
Big Bird ETC (84, 256)	-	1.611	N/C*
LSG (128, 2)	2.014	1.598	25000**

* Non communiqué par les auteurs mais warmup de 10000 étapes avec batch 256
** 15000 étapes sur 512 et 10000 sur 4096

TABLE 4.4 – BPC après fine-tuning intensif.

4.2 Application aux décisions de la CrEDH

Déterminer l'issue d'une affaire *a priori* est un objectif essentiel dans la justice prédictive. En pratique, l'exploitation de décisions complètes, de motifs ou de dispositifs n'est pas une configuration réaliste et n'a d'utilité que dans la construction de bases annotées. Dans le travail d'un juriste, seules des circonstances factuelles peuvent être connues concernant une affaire donnée. Nous nous intéressons aux décisions françaises de la *Cour Européenne des Droits de l'Homme* (CrEDH) qui ont pour particularité de posséder une structure relativement bien définie. Il est ainsi possible d'extraire des éléments tels que les faits à base de règles simples sans nécessairement recourir à un expert.

Cette section s'articule autour de la présentation, l'extraction et la construction de jeux de données afin de mettre en place des tâches de prédiction du sens du résultat pour lesquelles des séquences particulièrement longues doivent être traitées.

4.2.1 Données et vocabulaire

La tâche principale consiste à prédire la violation d'un article de la Convention Européenne des Droits de l'Homme (CEDH) [10] à partir d'un ensemble de

10. La CEDH (convention) regroupe un ensemble d'articles et de protocoles, la CrEDH (cour) juge sur la base de la convention.

faits. Pour cela, les arrêts publiés par la CEDH servent comme principale source de données [11].

Seront présentés dans les paragraphes suivants, les travaux effectués sur des tâches similaires dans d'autres langues puis la structure des données permettant de mettre en place l'extraction automatisée.

4.2.1.1 Structure des données

Les jugements de CrEDH sont constitués de centaines de phrases souvent structurées autour de titres et de paragraphes pour en faciliter la lecture. Un document peut être divisé en quatre parties. La première est la procédure et permet de fournir diverses informations sur les étapes et résultats précédents, notamment les décisions des tribunaux nationaux puisque la CrEDH oppose toujours une personne morale ou physique à un état (pays).

La deuxième section concerne les faits, ils constituent les données principales qui permettront de nourrir les modèles. Cette section fournit des informations générales sur l'affaire elle-même et tout ce qui n'est pas lié aux arguments juridiques, c'est-à-dire tout ce qui ne concerne pas les articles de la CEDH. Cette partie est généralement divisée en deux sous-sections : les circonstances de l'affaire et les lois pertinentes. La première relate les faits formulés par la Cour elle-même, nous supposerons cette représentation comme raisonnable et suffisamment exhaustive pour décrire des éléments factuels. Cette section est aussi la plus hétérogène car sa taille et son vocabulaire peuvent fortement varier même dans le cadre d'affaires similaires. La partie sur les lois pertinentes ajoute des informations sur les lois nationales et certains éléments juridiques, à l'exception des articles de la CEDH.

La troisième partie est juridique et s'appuie sur des arguments légaux pour examiner le bien-fondé de l'affaire. Pour se prononcer, la Cour doit justifier sa décision en exploitant des règles et des principes tenant compte d'une violation alléguée d'un article de la CEDH et des arguments fournis par les parties.

11. https://hudoc.echr.coe.int/

Enfin, la dernière section est celle des résultats de l'affaire. Elle énumère toutes les violations potentielles des articles de la Convention et indique si elles ont effectivement eu lieu. La structure générale est présentée dans la Figure 4.8.

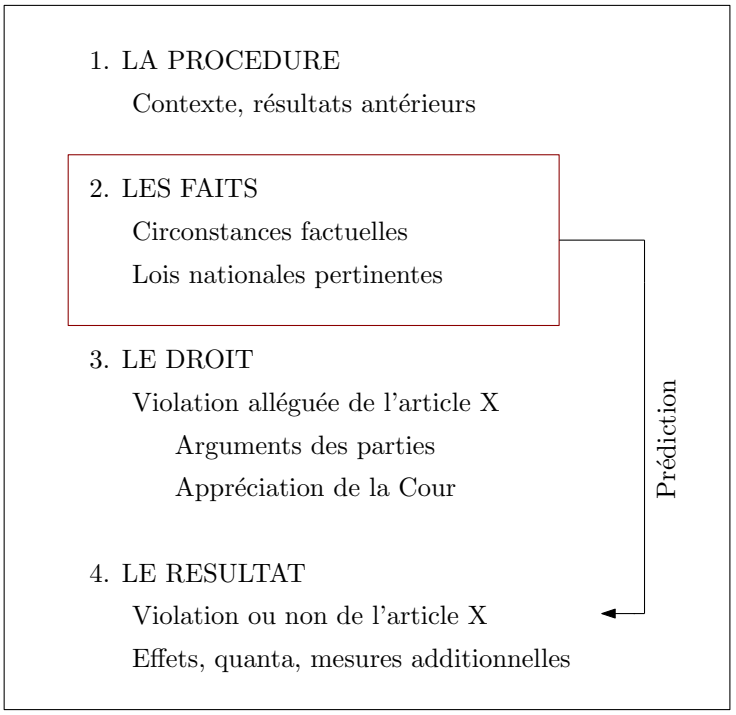


FIGURE 4.8 – Structure d'une décision de CrEDH.

4.2.1.2 Extraction des données

La majorité des documents suit la structure présentée quelle que soit la langue. Cependant, toutes les décisions ne sont pas traduites, les papiers traitant de problématiques similaires en langue anglaise ne se basent donc pas exactement sur les mêmes décisions [Aletras et al., 2016, Medvedeva et al., 2020].

L'extraction est simple en théorie car il suffit de repérer un ensemble de titres connus et imbriqués et d'extraire les paragraphes associés. En pratique, il existe une certaine variabilité entre les documents qui rend l'extraction automatique plus délicate. Certains titres peuvent changer, être déplacés dans une section inadéquate ou disparaître sans raison particulière. À titre d'exemple, la procédure est parfois intégrée dans la section des faits, ou la section des faits

peut ne pas être découpée en sous-sections. Le deuxième problème consiste en la suppression des résultats ambigus. Comme la tâche a pour objectif de déterminer si un article de la CEDH a été violé ou non, il est nécessaire de filtrer les décisions qui pour un même article de la CEDH ont des résultats différents. La procédure d'extraction est effectuée en 6 étapes :

1. filtrage des décisions par la langue pour sélectionner les françaises ;
2. vérification des 4 grandes sections à l'aide de regex ;
3. vérification que la section des faits est effectivement divisée en deux sous-sections à l'aide de regex ;
4. extraction des phrases du résultat mentionnant un ou plusieurs articles de la CEDH et le terme "violation" qui est systématiquement présent ;
5. inférence du sens du résultat pour chaque phrase sélectionnée et association avec les articles mentionnés ;
6. suppression des cas ambigus.

À l'issue de l'extraction, sont uniquement conservées les décisions relatives aux six articles les plus fréquents afin d'assurer un nombre suffisant d'observations : article 3 (tortures ou traitements inhumains et dégradants), article 5 (droit à la liberté et à la sécurité), article 6 (droit à un procès équitable), article 8 (respect de la vie privée et familiale), article 10 (liberté d'expression) et article 13 (accès à la justice). Des statistiques descriptives sont présentées en Table 4.5.

	Base initiale		Base équilibrée	
	Nombre	% Violations	Nombre	% Violations
Article 3	540	0.832	182	0.500
Article 5	561	0.892	122	0.500
Article 6	3704	0.937	468	0.500
Article 8	571	0.776	256	0.500
Article 10	476	0.853	140	0.500
Article 13	541	0.887	122	0.500
Base cumulée	6393	-	2327	-

TABLE 4.5 – Taille des jeux de données avant et après équilibrage.

D'après les statistiques, les bases sont très déséquilibrées et la CrEDH tend à condamner en majorité des états au profit du demandeur. Afin de reproduire le protocole des expérimentations similaires effectuées en anglais et de mesurer des performances plus réalistes, les bases sont équilibrées. La base cumulée permet de créer une seconde tâche de classification pour laquelle l'objectif est de déterminer quel article est pertinent pour un ensemble de faits.

4.2.2 Modèles et résultats

Les estimations se basent principalement sur l'utilisation de Transformers et notamment les variantes LSG présentées précédemment. Les expérimentations sont ensuite menées sur deux types de données : soit la section complète des faits, soit seulement sur la sous-section des circonstances factuelles.

4.2.2.1 Entraînement des modèles

Afin d'entraîner le modèle LSG-Norme, une base de 30Go de données textuelles juridiques est compilée. Elle est composée de plus de 1.5 millions de décisions toutes juridictions confondues, de l'ensemble des contenus des codes [12], de l'ensemble des débats parlementaires disponibles en ligne [13] et des questions-réponses au gouvernement [14].

Afin d'éviter de réentrainer un Transformer de zéro, les modèles LSG sont initialisés à partir d'un checkpoint CamemBERT [Martin et al., 2019] entraîné sur un large corpus de textes français. Cette architecture est équivalente à celle de RoBERTa à la différence du tokenizer dont la taille du vocabulaire est limitée à seulement 30000 tokens. Cela a pour effet un entraînement plus rapide mais limite les capacités du modèle dans la construction de mots plus rares.

Le préentraînement est effectué en 2 étapes : il est dans un premier temps basé sur des séquences de 512 tokens avec des batchs de 2048 entrées, un taux d'apprentissage de 2e-4 et 7500 étapes dont 500 de warmup. La seconde phase

12. https://www.legifrance.gouv.fr/
13. https://www.assemblee-nationale.fr/14/debats/index.asp
14. https://www2.assemblee-nationale.fr/recherche/questions

est effectuée sur des séquences de 4096 avec un taux d'apprentissage de 2e-5, des batchs de 256 durant 7500 étapes et un warmup de 500 étapes à nouveau. La base est structurée avec un document par ligne : lorsque le document est plus long que 512 ou 4096 entrées, celui-ci est découpé en plusieurs morceaux de même taille avec une intersection sur 10%. Les performances sont présentées Table 4.6.

	Longueur	BPC	Précision	Batch	Etapes	Warmup
LSG (128, 2)*	512	5.079	0.518	-	0	-
	512	0.741	0.873	2048	7500	500
LSG (128, 2)**	4096	0.740	0.873	-	0	-
	4096	0.634	0.888	256	7500	500

* Entraîné à partir de CamemBERT

** Entraîné à partir du modèle LSG sur des séquence de 512

TABLE 4.6 – Performances après fine-tuning MLM.

Les métriques montrent que CamemBERT est initialement mal adapté au langage juridique puisque la précision est d'à peine 51.8% pour un vocabulaire de seulement 30000 tokens comparé aux 50000 de RoBERTa. Ce phénomène peut remettre en cause la pertinence du tokenizer initial qui est peut-être mal adapté au découpage de termes juridiques peu communs dans les corpus généraux comme Wikipédia. De plus, les documents juridiques ont tendance à utiliser un grand nombre de sommes d'argent et de noms ou d'abréviations qui n'existent pas ailleurs. Il est important de noter que le passage de séquences de 512 éléments (tronquées) à celles de 4096 n'engendre pas de diminution des performances, signe d'une bonne extrapolation. Ceci est probablement dû au nombre très important de documents longs qui nécessitent un contexte suffisamment grand pour prédire les bons masques. L'entraînement sur des séquences 4096 permet de réduire la BPC de 15% mais la précision reste proche, une précision sur les k premiers éléments serait plus judicieuse.

À titre d'exemple, la structure de l'attention est représentée en Figure 4.9 et découpée par blocs. Le but est de visualiser comment les scores se concentrent en moyenne en fonction de la tête et de la couche.

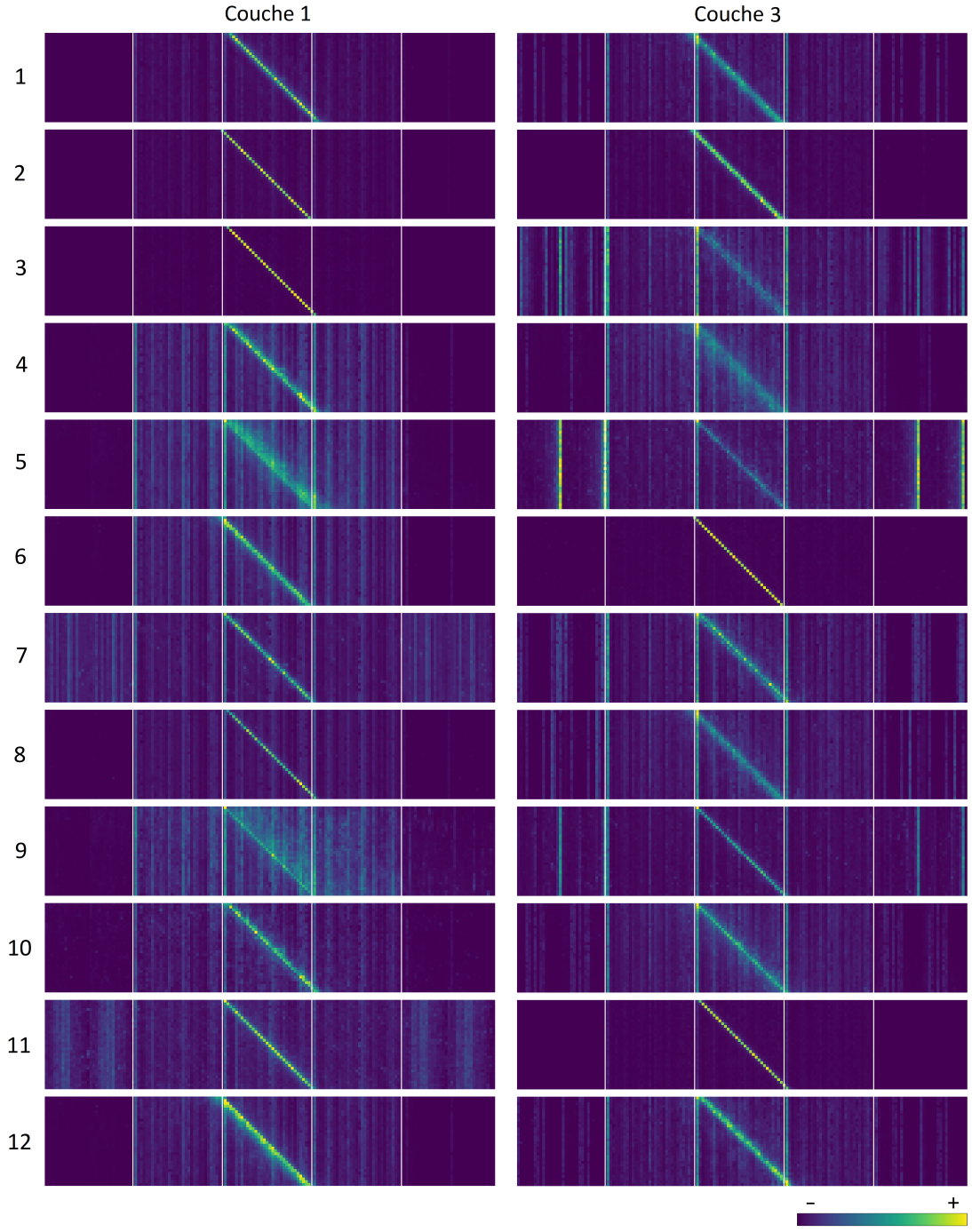


FIGURE 4.9 – Score d'attention moyen par tête, couches 1 et 3

Il est assez évident de voir que le modèle se focalise très fortement sur une petite bande diagonale dans le bloc central, c'est-à-dire autour des tokens les plus proches. Les blocs épars situés aux deux extrémités sont pondérés de façon

intermittente en fonction des besoins du modèle. À noter que dans la couche 3, certains éléments éloignés ont un poids très élevé ce qui amoindrit mécaniquement l'attention locale. On peut donc conclure que chaque tête et chaque couche ont un rôle particulier, chacune discriminant des éléments spécifiques, ce qui conforte certaines observations déjà établies [Clark et al., 2019].

4.2.2.2 Résultats

Cette section présente plusieurs expérimentations menées sur des tâches de prédiction du résultat par l'exploitation des circonstances factuelles puis de la section complète des faits (rappel de la procédure et circonstances factuelles). Sont testées plusieurs variantes du modèle LSG en modifiant le facteur f et en substituant la version exploitant la norme maximale et celle à base de pooling puis de clustering. Toutes les expériences sont menées sur une validation croisée en 10 étapes. Les résultats sont présentés en Tables 4.7 et 4.8.

Le modèle CamemBERT non entraîné sur le corpus juridique est largement moins performant comparé aux autres approches, de plus il n'est pas en mesure de traiter des séquences longues et se base sur les 512 premiers tokens des entrées. Le modèle LSG à base de norme est aussi testé sur cette taille de séquence pour vérifier que les différences de performances sont bien liées au préentrainement. Les précisions vérifient cette hypothèse puisque dans tous les cas de figure, cette architecture surpasse celle de CamemBERT. Du côté des variantes entraînées sur des séquences de taille 4096 maximum, les modèles sont très majoritairement plus performants lorsque le facteur de compression des éléments éloignés est fixé à 4, cela correspond à un contexte théorique de 1408 par couche. Il faut donc, dans ces conditions, 3 couches pour qu'une information entre le premier et le dernier token soit transmise (hors éléments globaux). Le modèle à base de clusters garde des performances semblables lorsque le facteur passe à 8 contrairement aux autres approches. Enfin, la variante à base de pooling est systématiquement en deça des autres, le pooling rend en réalité la convergence difficile.

Circonstances	Art. 3	Art. 5	Art. 6	Art. 8	Art. 10	Art. 13	Cumulé
(128, 2, 1)							
LSG-Norme	**0.781**	**0.792**	**0.824**	**0.777**	**0.786**	**0.872**	**0.764**
LSG-Pooling	0.778	0.790	0.813	0.769	0.770	0.864	0.764
LSG-Cluster	**0.781**	0.790	0.822	0.770	0.782	0.868	0.762
(128, 4, 1)							
LSG-Norme	0.790	**0.806**	0.830	0.784	**0.802**	**0.888**	**0.774**
LSG-Pooling	0.770	0.792	0.811	0.762	0.767	0.866	0.760
LSG-Cluster	**0.785**	0.801	**0.832**	**0.785**	0.798	0.876	0.772
(128, 8, 1)							
LSG-Norme	0.783	**0.804**	0.828	0.781	0.794	0.872	0.769
LSG-Pooling	0.766	0.788	0.807	0.758	0.767	0.844	0.752
LSG-Cluster	**0.788**	0.801	**0.831**	**0.783**	**0.800**	**0.881**	**0.772**
LSG-Norme*	0.754	0.787	0.804	0.755	0.762	0.845	0.743
CamemBERT*	0.723	0.757	0.771	0.725	0.742	0.815	0.723

* Entrées tronquées sur les 512 premiers éléments de la séquence.

TABLE 4.7 – Performances sur les circonstances factuelles.

Les observations menées sur le traitement des circonstances factuelles uniquement se transposent à nouveau ici. CamemBERT est moins performant à nouveau, les autres modèles sont quant à eux plus précis lorsque le facteur de compression vaut 4. Globalement les performances sont meilleures ici puisque la section englobe un rappel de procédure des tribunaux nationaux qui peut contenir des éléments discriminants concaténés aux circonstances factuelles traitées avant. Cela rallonge légèrement les séquences car la procédure est une sous-section généralement plus courte que les autres. La version à base de *clustering* est à nouveau stable avec un facteur 8 alors que les deux autres variantes perdent en performance. À ce niveau de compression, le contexte peut atteindre une taille maximale de 2432 par couche.

Faits complets	Art. 3	Art. 5	Art. 6	Art. 8	Art. 10	Art. 13	Cumulé
(128, 2, 1)							
LSG-Norme	**0.791**	**0.794**	**0.834**	**0.788**	**0.793**	**0.880**	**0.772**
LSG-Pooling	0.782	0.790	0.824	0.779	0.780	0.865	0.770
LSG-Cluster	0.788	0.785	0.822	0.784	0.786	0.876	0.769
(128, 4, 1)							
LSG-Norme	0.800	**0.816**	0.843	0.794	**0.812**	0.888	**0.786**
LSG-Pooling	0.790	0.796	0.830	0.782	0.789	0.879	0.764
LSG-Cluster	**0.802**	0.811	**0.844**	**0.795**	0.809	**0.890**	0.783
(128, 8, 1)							
LSG-Norme	0.794	**0.809**	0.838	0.791	0.804	0.874	0.779
LSG-Pooling	0.776	0.789	0.816	0.765	0.769	0.864	0.755
LSG-Cluster	**0.780**	0.805	**0.842**	**0.796**	0.810	**0.883**	**0.785**
LSG-Norme*	0.755	0.790	0.808	0.756	0.766	0.849	0.748
CamemBERT*	0.725	0.758	0.779	0.724	0.744	0.814	0.725

* Entrées tronquées sur les 512 premiers éléments de la séquence.

TABLE 4.8 – Performances sur la section FAITS.

Conclusion

Ce chapitre a montré qu'il est possible de construire des variantes efficientes de l'attention qui sont capables d'exploiter des modèles déjà entraînés tout en assurant une forte capacité d'extrapolation. Les architectures LSG, en plus d'être rapides peuvent se substituer entre elles puisqu'elles partagent l'attention locale et les connexions globales. La variante à base de norme tend à être la plus performante dans les tâches de MLM tandis que celle à base de clustering est en capacité de traiter un contexte très étendu pour des problématiques qui nécessitent un accès à l'information sur des positions très éloignées. Cette faculté de traitement des séquences longues permet aussi d'améliorer la qualité des prévisions puisque le fait de tronquer les entrées entraîne une baisse non

négligeable des performances ; baisse qui pourrait donc être évitée à l'aide de mécanismes semblables à celui proposé.

Chapitre 5

Interprétabilité et théorie des jeux coopératifs

La capacité d'expliquer les prédictions de modèles d'apprentissage automatique est essentielle lorsque celles-ci peuvent entraîner un préjudice important, irréparable, ou lorsque le praticien est tenu de fournir les éléments justifiant sa décision afin de couvrir des risques éthiques ou moraux. De nombreux travaux s'intéressent aux notions d'interprétabilité (*Explainable AI*) dans le but de déterminer les caractéristiques discriminantes d'une entrée au regard d'une prédiction obtenue. En traitement du langage, cela correspond généralement à l'extraction de termes ou de passages que le modèle considère comme importants pour motiver la prédiction. Les méthodes dites d'attribution visent à attribuer un score à chaque caractéristique afin de refléter leur contribution dans la résolution d'une tâche. Pour cela, certaines s'appuient sur la valeur de Shapley qui consiste à mesurer l'impact moyen d'une modification des entrées sur les prédictions du modèle en testant toutes les combinaisons de caractéristiques possibles. Si la logique sous-jacente de cette valeur motive son utilisation, elle est aujourd'hui remise en cause en raison de sa complexité et de son inadéquation pour certaines tâches, notamment en traitement d'images ou du langage.

Ce chapitre explore les méthodes d'attribution existantes ainsi que leurs limites, puis propose deux méthodes alternatives basées sur la valeur de Shapley et applicables dans les contextes sus-cités.

5.1 Méthodes d'attribution

Les modèles d'apprentissage profond sont aujourd'hui prédominants dans la résolution d'une grande variété de problèmes en traitement d'images ou de langues. Cependant, la compréhension du lien entre les entrées et les prédictions de ces boîtes noires, reste un problème ouvert [Linardatos et al., 2021, Das and Rad, 2020]. Or, de nombreux contextes applicatifs nécessitent à la fois des modèles très performants mais aussi de comprendre et d'interpréter pleinement les sorties des prédicteurs. Cela est vrai pour les cas d'utilisation critiques évidents, notamment dans le domaine médical où des décisions sensibles doivent être étayées par des preuves [Ching et al., 2018, Rudin, 2019], mais aussi dans des cas plus classiques où le refus d'un service par exemple doit être justifié (assurance). Cette volonté de transparence s'applique aussi aux problématiques de justice prédictive où l'intégrité et l'intelligibilité des outils doivent être respectées [1]. D'une façon générale, des préoccupations légitimes concernant les biais potentiels induits par des modèles boîte noire sont de plus en plus exprimées. En raison de ces préoccupations, les régulateurs introduisent donc des exigences légales imposant que les décisions automatisées ayant un impact sur la vie soient explicables [Goodman and Flaxman, 2017, Ras et al., 2018].

Cette section énumère les différentes méthodes d'attribution de l'état de l'art en décrivant dans un premier temps les approches les plus simples à travers un exemple illustratif et les modèles d'occlusion. Dans un second temps, sont présentées les méthodes modernes en lien avec la valeur de Shapley qui prédominent actuellement la littérature.

Cette section présente le concept d'attribution à travers un exemple simple permettant de situer les problèmes relatifs à ce type d'approche. Dans un second temps ce concept est plus formellement défini et sa portée est discutée.

5.1.1 Concept d'attribution et intérêt

La compréhension des relations entre les entrées et les sorties d'un modèle n'est pas aisée. Dans le cadre d'un modèle de régression linéaire simple, il

1. https://rm.coe.int/charte-ethique-fr-pour-publication-4-decembre-2018/16808f699b

n'existe pas de critère unanime permettant de déterminer quelle variable explication contribue le plus à la prédiction. Le choix de ce dernier est subjectif et peut être remis en cause.

5.1.1.1 Ambiguïté de l'attribution

Les méthodes d'attribution consistent à associer à une caractéristique, un score reflétant une contribution relativement aux autres caractéristiques par rapport à une prédiction dans un cadre local ou plusieurs prédictions dans un cadre global. Tous les modèles peuvent être la cible de méthodes d'interprétation sans pour autant être compréhensibles par un humain. Ce principe se vérifie dans certaines tâches d'apprentissage par renforcement où un algorithme trop performant, avec des capacités surhumaines, est mal compris dans ses décisions alors que celles-ci sont visiblement rationnelles. Cet aspect s'observe notamment sur les jeux de plateaux (go, échecs) [Silver et al., 2017] où de grands maîtres sont dans l'incapacité de comprendre certains choix de la machine ou d'évaluer la force de positions spécifiques. Cela peut s'expliquer par diverses raisons : un coup trop théorique pour un humain, un coup n'amenant des avantages qu'à très long terme alors que la partie s'interrompt avant, un coup dans le vide pour consolider un avantage. La capacité de compréhension des prédictions est donc limitée.

Dans le cas d'un prédicteur, une tâche d'attribution consiste à associer un nombre réel à chaque caractéristique d'entrée relativement à la valeur prédite, cette dernière pouvant être une probabilité dans le cadre d'un problème de classification ou une valeur réelle dans les problèmes de régression. Ce nombre représente une contribution et permet d'ordonner par importance les caractéristiques qui motivent la prédiction.

En pratique, le critère permettant de discriminer peut prendre de multiples formes et peut dépendre de propriétés souhaitables. Afin d'illustrer cet aspect, posons un problème de régression linéaire à d variables où $\boldsymbol{y} \in \mathbb{R}^n$ est la variable expliquée et $\boldsymbol{x}_j \in \mathbb{R}^n$ est une caractéristique j, telle que $j \in [1, \ldots, d]$:

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x}_1 + \ldots + \beta_d \boldsymbol{x}_d + \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{5.1}$$

L'interprétation du modèle est simple ici puisqu'il suffit de lire les valeurs des paramètres. La connaissance des contributions est plus complexe car elle a pour but de comparer l'impact des caractéristiques sur les prédictions selon un critère donné, c'est une mesure relative.

La contribution marginale La contribution marginale s'obtient par le calcul d'une dérivée partielle. Ainsi, le coefficient $\partial \boldsymbol{y}/\partial \boldsymbol{x}_j = \beta_j$, ou plus précisément son estimateur par moindres carrés ordinaires (MCO) $\hat{\beta}_j$ est la contribution marginale de la variable $\boldsymbol{x}_j$ lorsque les autres variables sont incluses dans le modèle. De façon naïve, on peut supposer que si pour les variables $\boldsymbol{x}_j$ et $\boldsymbol{x}_k$, $\beta_j > \beta_k$, alors la variable j contribue davantage que la variable k. Cet argument est faux et pose plusieurs problèmes. Premièrement, il est fait abstraction de tous les problèmes d'échelle puisqu'un coefficient aura tendance à compenser la norme de la variable associée. Si la variable $\boldsymbol{x}_j$ prend des valeurs très élevées, son coefficient β_j sera souvent proche de zéro, la standardisation permet cependant de régler cet aspect.

La significativité Il existe un problème relatif à la significativité des variables. Si pour un risque de première espèce α [2], la valeur 0 appartient à l'intervalle de confiance, il n'est pas possible de conclure sur le signe du coefficient. Le calcul de la contribution marginale à l'aide d'un gradient ne permet donc pas de déterminer l'importance d'une variable dans la majorité des cas. Par ailleurs, dans une régression linéaire, un test de Student peut être effectué pour mesurer à quel point une variable est significative, le critère souvent utilisé est la *p-value*. Or, ce critère dépend de plusieurs caractéristiques, notamment de la taille de l'échantillon. Ainsi peuvent être observées des variations importantes entre différents sous-échantillons.

Le coefficient de détermination Noté $R^2 \in [0, 1]$, il permet de mesurer la qualité d'une régression et représente la part de la variance de $\boldsymbol{y}$ expliquée par le modèle. Puisque ce critère est d'ordre qualitatif, il peut être exploité pour mesurer l'impact marginal d'une variable. Ainsi, en procédant par occlusion,

2. Probabilité de rejeter par erreur la non significativité de l'estimateur.

c'est-à-dire en comparant le critère avec et sans la variable dans le modèle, il est possible d'obtenir un ordonnancement des caractéristiques. Le choix du coefficient de détermination amène cependant un biais important. Le R^2 augmente mécaniquement pour chaque variable ajoutée dans le modèle même si celle-ci n'a pas réellement d'influence. Une solution consiste à pénaliser sa valeur en fonction du nombre de variables explicatives d présentes dans le modèle.

Le problème de l'occlusion L'occlusion est relative à un critère et peut être effectuée de plusieurs façons : soit en considérant un modèle dans lequel seule la caractéristique à évaluer est présente, soit en considérant un modèle complet dans lequel celle-ci est retirée. Dans le premier cas, le critère fait abstraction des relations entre les différentes variables. Dans le second cas, deux problèmes peuvent apparaître : l'un concerne la dépendance au nombre de variables d comme expliqué dans le paragraphe précédent, l'autre est relatif à la colinéarité. Une variable peut paraître inutile ou néfaste si une autre, très corrélée à cette dernière, est déjà présente dans le modèle. Dans ce cas, la valeur attribuée peut perdre son sens. Ce principe d'occlusion est généralisé dans le calcul de la valeur de Shapley pour laquelle les occlusions sont calculées pour toutes les combinaisons de caractéristiques possibles.

Pour un modèle aussi simple qu'une régression linéaire, il n'y a pas de règle unanimement reconnue permettant de classer des variables par ordre d'importance. Cette subjectivité des méthodes d'attribution s'observe quel que soit le modèle prédictif étudié. La recherche de propriétés souhaitables permet cependant d'éliminer certaines méthodes inadaptées pour des tâches spécifiques. Dans la suite de ce chapitre, seront considérés des problèmes de classification par le biais de prédicteurs de type réseaux de neurones.

5.1.1.2 Définition et intérêt

On considère un cadre de classification multi-classes avec un ensemble de classes $\mathcal{C} = [\![1, C]\!]$, avec $[\![a, b]\!]$ désignant l'intervalle de tous les entiers compris entre a et b inclus. Dans ce contexte, un prédicteur f prend une entrée de caractéristiques à N dimensions $\boldsymbol{x} = [x_1, \ldots, x_N] \in \mathbb{R}^N$ et produit une distribution

de probabilité $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_C(\boldsymbol{x})] \in [0,1]^C$, avec $f_i(\boldsymbol{x})$ la probabilité attribuée à la classe $i \in \mathcal{C}$ par f pour $\boldsymbol{x}$. On notera par la suite $\mathcal{N} = [\![1, N]\!]$ l'ensemble des indices des caractéristiques.

Dans ce cadre, étant donné le prédicteur f et une entrée $\boldsymbol{x} \in \mathbb{R}^N$, la méthode d'attribution (MA) φ vise à calculer un vecteur de contribution $\varphi(\boldsymbol{x}, f_i)$ pour toutes classes $i \in \mathcal{C}$ telle que $\varphi(\mathbf{x}, f_i) = [\varphi_1(\boldsymbol{x}, f_i), \ldots, \varphi_N(\boldsymbol{x}, f_i)] \in \mathbb{R}^N$, avec $\varphi_j(\boldsymbol{x}, f_i)$ la valeur d'attribution de la caractéristique $j \in \mathcal{N}$ par rapport à $f_i(\boldsymbol{x})$. Autrement dit, en considérant la MA φ, $\varphi_j(\boldsymbol{x}, f_i)$ est la contribution de la caractéristique j à la probabilité calculée par le prédicteur f pour la classe i et l'entrée $\boldsymbol{x}$.

En pratique, deux éléments supplémentaires sont à prendre en compte avant l'utilisation d'une méthode d'attribution : la portée des attributions (locales ou globales), c'est-à-dire si celles-ci doivent être obtenues pour chaque observation de façon indépendante ou si l'objectif est de déterminer les éléments que le modèle considère comme discriminants pour l'ensemble des données. L'autre aspect important concerne la nature des données, la définition d'une caractéristique n'est pas la même selon la forme que prennent les entrées. A titre d'exemple, une caractéristique pour un modèle de langue sera généralement un mot et pour une tâche de traitement d'images celle-ci sera un pixel. Dès lors, le choix d'une approche globale a peu de sens dans ces contextes puisque la méthode d'attribution associera un score à une position. Dans le cadre de données qui ne font pas intervenir de relations spatiales ou temporelles, les méthodes globales permettent d'obtenir des indicateurs et des tendances générales permettant notamment de filtrer de grandes bases.

Dans le cadre d'attributions locales, de nombreuses approches se basent sur des perturbations ou sur l'occlusion afin d'évaluer la contribution d'une caractéristique $j \in \mathcal{N}$ comme sa contribution à une ou des coalitions de caractéristiques. Si l'on considère une coalition comprenant toutes les caractéristiques à l'exception de j (c'est-à-dire $\mathcal{N} \setminus \{j\}$), la contribution de j à celle-ci est mesurée en évaluant l'impact d'une perturbation de x_j sur $f_i(\boldsymbol{x})$. Celle-ci visant à simuler la suppression de l'élément étudié en remplaçant sa valeur par une valeur de référence pour du traitement d'images, ou en utilisant un masque dans les tâches de TAL à base de Transformers.

Pour toute coalition $\mathcal{S} \subseteq \mathcal{N}$, $\boldsymbol{x}(\mathcal{S})$ désigne le vecteur $\boldsymbol{x}$ dans lequel toutes les valeurs de caractéristiques x_k, $k \in \mathcal{N} \setminus \mathcal{S}$ ont été substituées par une valeur de référence comme énoncé ci-dessus. Puisque l'entrée $\boldsymbol{x}$ est toujours la même dans nos discussions, $f_i(\mathcal{S})$ est utilisé pour désigner $f_i(\boldsymbol{x}(\mathcal{S}))$, qui est la probabilité attribuée par f à la classe $i \in \mathcal{C}$ par rapport à $\boldsymbol{x}(\mathcal{S})$.

La contribution marginale par occlusion d'une caractéristique $j \in \mathcal{N}$ à une coalition $\mathcal{S}$ ($j \notin \mathcal{S}$) est donc définie par $f_i(\mathcal{S} \cup \{j\}) - f_i(\mathcal{S})$. De nombreux modèles d'attribution basés sur cette notion ont été étudiés en exploitant des concepts de la théorie des jeux coopératifs [Funaki et al., 1997, Brink et al., 2013, Young, 1985, Wang et al., 2020a]. La valeur de Shapley est dès lors utilisée comme référence puisqu'elle permet d'expliquer le rôle d'une variable donnée dans un modèle en se basant sur cette notion de contribution [Ancona et al., 2019].

5.1.2 Méthodes d'attribution modernes

Le calcul de scores d'attribution est un problème ouvert et étudié sous de nombreux angles et à l'aide de diverses approches. Un grand nombre de travaux se concentrent sur les méthodes d'attribution en s'appuyant sur des motivations axiomatiques permettant de définir des propriétés intuitives et souhaitables de ces méthodes [Sun and Sundararajan, 2011, Sundararajan et al., 2017, Montavon et al., 2017, Lundberg and Lee, 2017]. Nombreuses d'entre elles considèrent la valeur de Shapley [Shapley, 1953] comme une mesure de référence puisque celle-ci définit l'unique façon de résoudre des problématiques d'attribution selon les axiomes admis dans le cadre de problèmes à base de coalitions. L'attribution est dès lors effectuée en considérant un jeu coopératif dans lequel les caractéristiques sont les joueurs. Dans ce contexte, plusieurs approches ont été proposées afin d'approximer et de réduire drastiquement la complexité de cette valeur qui nécessite l'évaluation de 2^N sous-ensembles pour un problème à N caractéristiques [Ancona et al., 2019]. Cependant, même si les contributions soulignent que les méthodes d'attribution basées sur la valeur de Shapley semblent correspondre aux attentes intuitives humaines, il n'existe pas d'unanimité quant à la pertinence absolue de cette valeur qui est donc remise en question par l'état de l'art récent [Kumar et al., 2020]. La sous-section suivante présente la valeur de Shapley et certaines approches parfois dérivées de celle-ci.

5.1.2.1 Valeur de Shapley

La valeur de Shapley φ_j^{Sh} est la moyenne des contributions marginales par occlusion de toutes les coalitions de caractéristiques possibles :

$$\varphi_j^{Sh}(\boldsymbol{x}, f_i) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} P(\mathcal{S}) \Big(f_i(\mathcal{S} \cup \{j\}) - f_i(\mathcal{S}) \Big), \tag{5.2}$$

pour tout $j \in \mathcal{N}$; $f_i(\emptyset) = 0$ pour tout $i \in \mathcal{C}$ par convention. Le facteur $P(\mathcal{S})$ est un facteur de pondération qui peut être adapté dans le but de filtrer voire d'éliminer certaines coalitions spécifiques. Dans le calcul de la valeur de Shapley traditionnelle, ce terme est défini par :

$$P(\mathcal{S}) = \frac{(N - |\mathcal{S}| - 1)!|\mathcal{S}|!}{N!} \tag{5.3}$$

La valeur de Shapley implique (et est impliquée par) quatre axiomes : l'*efficience*, l'*additivité*, la *symétrie* et l'*axiome du joueur nul* [Shapley, 1953] [3]. Ces axiomes rendent la valeur de Shapley attrayante d'un point de vue théorique, et motivent son statut de valeur de référence pour les problèmes d'attribution.

Cependant, si l'on considère N caractéristiques, 2^N coalitions doivent être évaluées, ce qui rend le calcul de la valeur de Shapley prohibitif. Une façon naturelle de réduire la complexité consiste à s'appuyer sur des circuits booléens [Arenas et al., 2021] ou sur de l'échantillonnage de coalitions pour calculer les contributions marginales [Castro et al., 2009]. Cette dernière approche souffre cependant de problèmes de convergence lorsque le nombre de caractéristiques est important. Plutôt que de s'appuyer sur les entrées originales, DASP [Ancona et al., 2019] exploite et propage la distribution des données à l'aide d'un réseau de neurones auxiliaire spécifique (Lightweight Probabilistic Deep Networks [Gast and Roth, 2018]). Ce modèle produit séquentiellement une estimation pour chaque taille de coalition, permettant ainsi de réduire considérablement la complexité de $O(2^N)$ à $O(N^2)$. Bien que cette approximation soit précise, la construction d'un réseau parallèle dont chaque couche et chaque

3. Il convient de noter que l'*additivité* implique la *linéarité* mais que l'inverse n'est pas vrai. Invoquer la *linéarité* élargit la classe des méthodes d'attribution admissibles, voir le théorème 5.4.

fonction d'activation doivent être adaptées est fastidieuse, en particulier lorsqu'il s'agit de traiter un modèle pré-entraîné dont l'architecture peut être particulièrement complexe voire incompatible. D'autres variantes de la valeur de Shapley relâchent certains axiomes, notamment la *symétrie* [Frye et al., 2020] et permettent d'incorporer des relations causales directement dans le modèle tout en adaptant le schéma de pondération.

La valeur de Shapley est privilégiée pour ses propriétés, mais inappropriée dans de nombreuses situations car elle ignore la structure des données. En traitement d'images, la plupart des prédicteurs ont des architectures à base de réseaux convolutifs qui traitent l'information par le biais d'une fenêtre de taille prédéfinie. Or, la prise en compte de coalitions dans lesquelles les pixels sont sélectionnés de façon erratique nuit fortement à ce type d'opération dont le résultat sera peu prévisible. La restriction à des coalitions dont les éléments sont spatialement ou temporellement proches est donc très importante car ces structures sont extrêmement rares parmi toutes les combinaisons possibles (Figure 5.1).

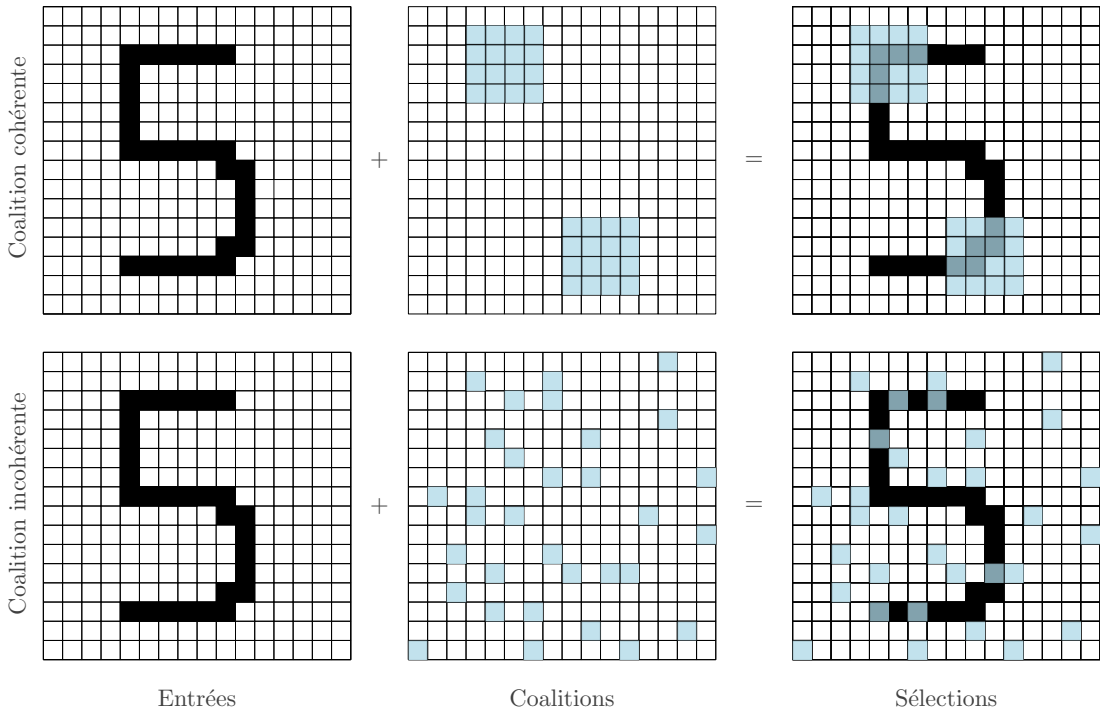


FIGURE 5.1 – Coalitions de taille 32.

5.1.2.2 Approches alternatives

Les méthodes d'attribution peuvent se focaliser sur différents objectifs. Bien que celui présenté dans ce chapitre est orienté vers l'évaluation de caractéristiques servant d'entrée à un prédicteur, d'autres approches tentent non pas d'expliquer une entrée mais d'expliquer l'activation de neurones ou le comportement de couches spécifiques dans un réseau. Ces approches se basent souvent sur des modèles d'attribution existants auxquels sont greffés des calculs de gradients [Dhamdhere et al., 2018, Shrikumar et al., 2018, Ancona et al., 2018, Lundberg and Lee, 2017].

La seconde branche des modèles d'attribution s'intéresse aux méthodes par perturbations. Ces approches se basent soit sur la perturbation des entrées, soit plus rarement sur la perturbation de paramètres pour comprendre le fonctionnement interne de certaines couches. Le prédicteur est sondé de façon itérative par des données dont les caractéristiques ont été remplacées par du bruit ou masquées. Ce principe est largement exploité dans les approches par occlusion en modifiant des superpixels dans le cadre du traitement d'images et des séquences de mots pour du langage. Ce mécanisme se retrouve dans la plupart des méthodes dites agnostiques telle que LIME (*Local Interpretable Model-Agnostic Explanations*) [Ribeiro et al., 2016], qui approxime à l'aide d'un modèle linéaire les attributions après construction d'exemples perturbés. D'autres variantes existent à base de déconvolutions [Zeiler and Fergus, 2013] et de multiplications par des masques aléatoires [Petsiuk et al., 2018].

La troisième branche des modèles d'attribution se base exclusivement sur des calculs de gradients et sur la rétropropagation. La plupart de ces approches sont peu coûteuses à calculer puisqu'une seule prédiction permet d'obtenir toutes les matrices de dérivées partielles. Il est notamment montré que la multiplication d'une entrée par son gradient permet de générer une représentation facilement compréhensible et interprétable, en particulier dans les tâches de traitement d'images [Ancona et al., 2018]. Cette approche est cependant critiquée à cause du problème de saturation : la modification d'une seule caractéristique (un pixel ou un mot) même si celle-ci est importante peut n'avoir qu'une répercution marginale sur la sortie du prédicteur, le gradient étant dans ce cas sous-estimé. Le

modèle *DeepLift* [Shrikumar et al., 2019] corrige cet effet en exploitant une valeur de référence pour comparer les activations des neurones. Cette approche s'inspire de la LRP (*Layer-wise Relevance Propagation*), qui repose sur une idée similaire sans utiliser de référence [Bach et al., 2015]. Ces méthodes à base de gradients sont cependant critiquées pour des raisons théoriques car elles ne respectent pas certains axiomes souhaitables. Le modèle *Integrated Gradient* [Sundararajan et al., 2017] satisfait l'axiome de complétude, axiome très proche de celui d'efficience en théorie des jeux coopératifs : pour une référence $\boldsymbol{x'}$, la valeur φ doit respecter la relation $\sum_{j\in\mathcal{N}} \varphi_j(\boldsymbol{x}, f) = f(\boldsymbol{x}) - f(\boldsymbol{x'})$. Ce modèle calcule les contributions en faisant la moyenne de différents gradients grâce à de nouvelles entrées générées comme des combinaisons linéaires de l'entrée initiale et d'une référence. Ce principe est relié à une autre branche de la littérature basée sur l'analyse des coalitions dans un cadre continu, comme la valeur Aumann-Shapley [Sundararajan and Najmi, 2020].

Enfin, la dernière branche est inspirée de la valeur de Shapley avec diverses variations et approximations pour en réduire la complexité. La méthode la plus naïve consiste à estimer la valeur par échantillonnage [Castro et al., 2009], en tirant aléatoirement des coalitions pour espérer converger vers la vraie valeur. Cette approche n'est pas applicable lorsque le nombre de caractéristiques est important. La seconde façon d'approximer la valeur est d'exploiter la distribution des données par le biais d'un réseau auxiliaire [Ancona et al., 2019], cette méthode est cependant très contraignante puisque tous les types de couches ne sont pas compatibles. La bibliothèque SHAP [Lundberg and Lee, 2017] propose, en suivant la même idée d'approche locale que LIME, de s'appuyer sur un modèle simplifié et défini comme une approximation interprétable du modèle original. En posant f le prédicteur à expliquer et g son approximation interprétable, on considère l'entrée $\boldsymbol{x}$ et $\boldsymbol{x'}$ tel que $\boldsymbol{x} = h_x(\boldsymbol{x'})$. L'approche locale vérifie que $g(\boldsymbol{z'}) \approx f(h_x(\boldsymbol{z'}))$ quand $\boldsymbol{z'} \approx \boldsymbol{x'}$. Avec M le nombre de caractéristiques simplifiées, la méthode d'attribution additive proposée par les auteurs est définie par :

$$g(\boldsymbol{z'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i z'_i \qquad \text{avec} \quad \boldsymbol{z'} \in [0,1]^M$$

En fusionnant cette approche avec d'autres modèles, notamment DeepLift et

Integrated Gradient, les auteurs proposent des variantes adaptées aux réseaux de neurones profonds (DeepExplainer, ShapExplainer), aux arbres de décision (TreeExplainer), aux SVM (KernelExplainer) et au calcul de gradients (GradientShap).

5.2 Méthodes proposées

Les méthodes d'attribution les plus utilisées se heurtent à plusieurs difficultés. La première concerne la non prise en compte de l'information spatiale ou temporelle du fait de l'utilisation de gradients. L'autre aspect important concerne les valeurs attribuées elles-mêmes qui ne sont pas forcément interprétables ou ne donnent pas d'information sur le signe de l'attribution (occlusion). Dans ces conditions, il n'est pas possible de déterminer quelles caractéristiques contribuent négativement à la prédiction. La valeur de Shapley reste séduisante mais du fait de sa complexité n'est pas envisageable pour des approches à base de réseaux de neurones. Afin d'exploiter certaines caractéristiques de cette valeur, tout en assurant une complexité réduite, sont proposées deux nouvelles méthodes basées sur les valeurs LES (valeurs Linéaires, Efficientes et Symétriques) dont l'une est tirée de la littérature de la théorie des jeux. Ces méthodes sont confrontées aux méthodes existantes et étudiées sur des tâches standards de classification puis sur les problématiques exposées aux chapitres précédents à savoir le traitement d'images et du langage. [4]

5.2.1 Famille des LES et alternatives

Les deux nouvelles méthodes d'attribution que nous proposons exploitent des propriétés de la valeur de Shapley tout en filtrant une grande part des coalitions pour permettre une réduction drastique de la complexité. La valeur de Shapley est considérée comme relativement simple à interpréter puisqu'elle représente la moyenne de toutes les contributions marginales par occlusion de chaque caractéristique, c'est donc à cet égard une valeur *marginaliste*. Elle partage certaines

4. https://github.com/benderama3/fesp_es

propriétés communes avec d'autres valeurs marginalistes qui forment la famille des valeurs LES (*linéaires-efficientes-symétriques*) [Ruiz et al., 1998].

5.2.1.1 Propriétés des valeurs LES

Cette famille particulière n'a à notre connaissance pas été étudiée dans le contexte de problèmes d'attribution. Les axiomes respectés par les valeurs LES sont présentés ci-après.

La *linéarité* postule que la contribution de chaque caractéristique à une combinaison de deux modèles pondérés f et g correspond à la somme de la contribution pondérée de chaque caractéristique sur les deux modèles. Il s'agit d'une propriété de base des valeurs marginalistes. Cependant, elle sera relâchée dans la méthode d'attribution du modèle FESP présenté ensuite.

Axiome 5.1. Linéarité : *Pour les prédicteurs f et g, une méthode d'attribution φ satisfait l'axiome de linéarité si, $\varphi(\boldsymbol{x}, \alpha_1 f_i + \alpha_2 g_i) = \alpha_1 \varphi(\boldsymbol{x}, f_i) + \alpha_2 \varphi(\boldsymbol{x}, g_i)$, pour tout $\alpha_1, \alpha_2 \in \mathbb{R}$ et pour toutes les classes $i \in \mathcal{C}$.*

L'*efficience* (ou complétude) postule que la somme de toutes les contributions des caractéristiques fournit la probabilité issue du modèle f (pour une classe i donnée) sur la grande coalition de caractéristiques $f_i(\mathcal{N})$. Il n'y a donc pas de perte de contribution dans la méthode d'attribution.

Axiome 5.2. Efficience : *Pour tout prédicteur f, une méthode d'attribution φ satisfait l'efficience si, $\sum_{j \in \mathcal{N}} \varphi_j(\boldsymbol{x}, f_i) = f_i(\mathcal{N})$, pour toutes classes $i \in \mathcal{C}$.*

La *symétrie* postule que la méthode d'attribution ne dépend pas de l'ordre des caractéristiques employées dans le modèle.

Axiome 5.3. Symétrie : *Pour tout prédicteur f, une méthode d'attribution φ satisfait la symétrie si, pour toutes les caractéristiques $j \in \mathcal{N}$, $\varphi_j(\boldsymbol{x}, f_i) = \varphi_{\pi(j)}(\boldsymbol{x}_\pi, f_i)$ pour toute permutation π sur l'ensemble des $N!$ permutations de $\mathcal{N}$ et pour toutes classes $i \in \mathcal{C}$.*

Les valeurs LES ont été largement étudiées et caractérisées en dehors de la littérature sur l'apprentissage automatique [Ruiz et al., 1998, Hernández-Lamoneda et al., 2007, Nembua and Andjiga, 2008, Chameni Nembua, 2012, Radzik and Driessen, 2013]. Elles sont toutes basées sur le principe de contributions marginales et peuvent donc être interprétées de façon similaire à la valeur de Shapley. La substitution de l'axiome d'additivité par celui de linéarité présente un avantage. Puisque l'additivité implique la linéarité, invoquer la linéarité élargit la classe des valeurs admissibles pour l'interprétation du prédicteur. La famille LES est caractérisée par le théorème suivant [Ruiz et al., 1998] :

Théorème 5.4. *Pour tout prédicteur f et toutes classes $i \in \mathcal{C}$, une méthode d'attribution φ satisfait la linéarité, l'efficience et la symétrie (LES) si et seulement s'il existe une séquence unique de $N-1$ nombres réels $\{b_s\}_{s=1}^{N-1}$ telle que pour chaque caractéristique $j \in \mathcal{N}$ avec $b_0 = 0$ et $b_N = 1$:*

$$\varphi_j(\boldsymbol{x}, f_i) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} P(\mathcal{S}) \Big(b_{s+1} f_i(\mathcal{S} \cup \{j\}) - b_s f_i(\mathcal{S}) \Big).$$

La valeur de Shapley φ^{Sh} est en réalité un cas particulier de la famille LES dans lequel les contributions marginales sont équipondérées ($b_s = 1$ pour tous les $s = 1, \ldots, N-1$). Il existe des valeurs LES connues et largement étudiées dans la littérature de la théorie des jeux coopératifs avec des complexités variables (Table 5.1), parmi lesquelles la valeur égalitaire de surplus (ES, φ^{Es}) [Driessen and Funaki, 1991], la valeur de solidarité (φ^{So}) [Nowak and Radzik, 1994], le valeur prénucléolus (φ^{LS}) [Ruiz et al., 1996], et la valeur de consensus (φ^{Co}) [Ju et al., 2007].

φ_j^{Sh}	φ_j^{ES}	φ_j^{So}	φ_j^{LS}	φ_j^{Co}
$O(2^N)$	$O(N)$	$O(2^N)$	$O(2^N)$	$O(2^N)$

TABLE 5.1 – Complexité des LES les plus connues.

La valeur ES est particulièrement intéressante car elle est de complexité $O(N)$ contrairement aux autres variantes sus-citées de complexité $O(2^N)$. Cette

LES se base exclusivement sur des prédicteurs n'exploitant qu'une seule caractéristique à la fois :

$$\varphi_j^{ES}(\boldsymbol{x}, f_i) = f_i(\{j\}) + \frac{f_i(\mathcal{N}) - \sum_{k=1}^{N} f_i(\{k\})}{N} \quad \forall i, j \in \mathcal{C}, \mathcal{N} \tag{5.4}$$

Puisque le second terme est une constante, ce modèle revient a évaluer le prédicteur avec une seule caractéristique puis à retrancher un biais pour obtenir les attributions finales. Ce biais correspond au gain supplémentaire produit par la grande coalition par rapport à la somme des contributions marginales individuelles des caractéristiques x_j. A noter que le terme $f_i(\{j\})$ correspond à une occlusion sur les coalitions de taille 1, puisqu'une prédiction sur un ensemble de caractéristiques vide n'a pas de sens, $f_i(\emptyset) = 0$.

5.2.1.2 Méthodes alternatives

Bien que l'ES soit séduisante, elle possède deux défauts dans sa formulation initiale. Le premier concerne la non prise en compte de relations entre caractéristiques puisque celles-ci sont évaluées indépendamment par le biais du terme $f_i(\{j\})$. Le second concerne l'application d'une telle approche sur des tâches où le nombre de caractéristiques est élevé, notamment en traitement d'images et du langage. Le fait d'effectuer une prédiction avec un seul pixel ou un seul mot ne permet pas de comprendre le comportement du modèle étudié.

Le modèle proposé nommé FESP (*Fair-Efficient-Symmetric-Perturbation*) répond à ces problématiques en réutilisant le principe du modèle d'occlusion simple. Plutôt que de se focaliser dans le calcul de contributions marginales uniquement sur des coalitions de taille minimale ($f_i(\{j\}) - f_i(\emptyset)$), il semble tout aussi logique d'exploiter les grandes coalitions. En effet, lorsqu'une caractéristique discriminante est exclue de l'ensemble, la probabilité associée à la classe prédite initialement doit diminuer significativement. L'occlusion liée à la caractéristique x_j sur la classe i peut être simplement caractérisée par $f_i(\mathcal{N} \setminus \{j\})$ au lieu du surplus $f_i(\mathcal{N}) - \sum_{j=1}^{N} f_i(\{j\})$ calculé par l'ES. Par exemple, certains mots peuvent être discriminants pour trouver l'étiquette associée à un paragraphe (que ce soit pour un humain ou un réseau de neurones). Ainsi, la

suppression de certaines parties du document peut diminuer la probabilité associée à une classe spécifique. Il est dès lors possible de considérer une méthode d'attribution exploitant les coalitions extrêmes $f_i(\{j\})$ et $f_i(\mathcal{N} \setminus \{j\})$ et de définir de nouveaux axiomes. Nous proposons l'axiome suivant :

Axiome 5.5. Coalitions de caractéristiques extrêmes : *Pour tout prédicteur f, une méthode d'attribution φ satisfait les coalitions de caractéristiques extrêmes, pour toute caractéristique $j \in \mathcal{N}$ et toutes classes $i \in \mathcal{C}$ si :*

$$\varphi_j(\boldsymbol{x}, f_i) = w_i f_i(\{j\}) + (1 - w_i)(-f_i(\mathcal{N} \setminus \{j\})),$$

avec $w_i \in [0, 1]$ la pondération entre les coalitions extrêmes.

Puisque les termes $f_i(\{j\})$ et $f_i(\mathcal{N} \setminus \{j\})$ évoluent dans des sens différents, le signe négatif est ajouté au second. Une méthode d'attribution respectant cet axiome est composée de deux éléments. Le premier $w_i f_i(\{j\})$ est fondé sur la contribution marginale individuelle de la caractéristique : plus la caractéristique est importante, plus ce terme le sera puisque la probabilité associée à la classe i augmentera. Le deuxième élément, $(1 - w_i)(-f_i(\mathcal{N} \setminus \{j\}))$, correspond à la contribution de l'occlusion. Le signe négatif permet de corriger la direction de ce terme puisque $f_i(\mathcal{N} \setminus \{j\})$ est faible si la caractéristique j est importante pour la prédiction de la classe i. Le modèle proposé FESP (*Fair-Efficient-Symmetric-Perturbation*) est basé sur les *coalitions de caractéristiques extrêmes* et l'*efficience* pour déterminer les poids w_i. L'efficience est respectée si :

$$\sum_{j \in \mathcal{N}} [w_i f_i(\{j\}) + (1 - w_i)(-f_i(\mathcal{N} \setminus \{j\}))] = f_i(\mathcal{N})$$

L'écriture de w_i est déduite directement :

$$w_i = \frac{f_i(\mathcal{N}) + \sum_{k=1}^{N} f_i(\mathcal{N} \setminus \{k\})}{\sum_{k=1}^{N} f_i(\{k\}) + \sum_{k=1}^{N} f_i(\mathcal{N} \setminus \{k\})} \quad \forall i \in \mathcal{C} \tag{5.5}$$

Une autre propriété que seuls certains membres de la famille LES satisfont (valeur de Shapley et ES) est l'axiome de *traitement équitable*. Pour $k, \ell \in \mathcal{N}$, la caractéristique x_k est plus pertinente que la caractéristique x_ℓ si l'association

de x_k avec toutes les coalitions de caractéristiques $\mathcal{S} \setminus \{k, \ell\}$ fournit une valeur d'attribution supérieure à celle de x_ℓ [Radzik and Driessen, 2013].

Axiome 5.6. Traitement équitable : *Pour tout prédicteur f, et deux caractéristiques x_k, x_ℓ, une MA φ satisfait le traitement équitable si, chaque fois que la caractéristique x_k est plus pertinente que x_ℓ, $f_i(\mathcal{S} \cup \{k\}) \geq f_i(\mathcal{S} \cup \{\ell\})$ pour tout $\mathcal{S} \subseteq \mathcal{N} \setminus \{k, \ell\}$, alors $\varphi_k(\boldsymbol{x}, f_i) \geq \varphi_\ell(\boldsymbol{x}, f_i)$, pour toutes classes $i \in \mathcal{C}$.*

Le modèle FESP vérifie cette propriété. Soit $\mathcal{S} = \mathcal{N} \setminus \{k, \ell\}$, tel que $k, \ell \in \mathcal{N}$:

$$\begin{aligned}
&\varphi_k^{FESP}(\boldsymbol{x}, f_i) - \varphi_\ell^{FESP}(\boldsymbol{x}, f_i) \\
&= w_i[f_i(\{k\}) - f_i(\{\ell\})] + (1 - w_i)[f_i(\mathcal{N} \setminus \{\ell\}) - f_i(\mathcal{N} \setminus \{k\})] \\
&= w_i[f_i(\{k\}) - f_i(\{\ell\})] + (1 - w_i)[f_i(\mathcal{S} \cup \{k\}) - f_i(\mathcal{S} \cup \{\ell\})]
\end{aligned}$$

Puisque $f_i(\mathcal{S} \cup \{k\}) \geq f_i(\mathcal{S} \cup \{\ell\})$ pour tout $\mathcal{S} \subseteq \mathcal{N} \setminus \{k, \ell\}$, alors :

$$\varphi_k^{FESP}(\boldsymbol{x}, f_i) - \varphi_\ell^{FESP}(\boldsymbol{x}, f_i) \geq 0 \quad \forall i \in \mathcal{C} \tag{5.6}$$

Enfin FESP respecte l'axiome de symétrie. Considérons une permutation $\pi \in \Pi$ telle que $\pi(j) = k$ pour n'importe quel $k \neq j \in \mathcal{N}$. Alors,

$$\begin{aligned}
\varphi_j^{FESP}(\boldsymbol{x}_\pi, f_i) &= w_i f_i(\{\pi(j)\}) + (1 - w_i)(-f_i(\mathcal{N} \setminus \{\pi(j)\})) \\
&= w_i f_i(\{k\}) + (1 - w_i)(-f_i(\mathcal{N} \setminus \{k\})) \\
&= \varphi_k^{FESP}(\boldsymbol{x}, f_i) \\
&= \varphi_{\pi(j)}^{FESP}(\boldsymbol{x}, f_i)
\end{aligned}$$

FESP conserve une complexité faible $O(N)$ tout en ayant une structure proche des LES. Cette méthode d'attribution respecte l'efficience, la symétrie et le traitement équitable. L'axiome de linéarité n'est pas respecté du fait de la structure des poids w_i qui permettent de calculer une somme pondérée entre les coalitions extrêmes.

Dans la suite de ce chapitre, seront considérées des expérimentations dans lesquelles ES et FESP exploitent non pas des caractéristiques individuelles mais des ensembles de caractéristiques localisées, notamment des superpixels, des n-grammes ou des segments de mots contigus et de tailles variables.

5.2.2 Applications

La nature des méthodes d'attribution rend difficile leur évaluation puisque l'aspect discriminant d'une caractéristique peut être subjectif selon la tâche, les approches qualitatives prévalent donc à ce jour [Adebayo et al., 2018]. L'une des méthodes permettant de vérifier que le modèle exploite bien les caractéristiques sélectionnées consiste à effectuer des prédictions en se servant uniquement des éléments ayant les contributions les plus élevées. Sont développées dans les sections suivantes des expérimentations sur diverses tâches générales puis sur les problématiques de séquences longues présentées dans le chapitre précédent.

5.2.2.1 Applications générales et comparaisons

Le protocole proposé consiste à entraîner un modèle sur unê tâche puis à en expliquer les prédictions selon les entrées considérées. Afin de se concentrer sur l'évaluation de la MA et d'éviter tout biais interprétatif, des tâches simples sont considérées pour lesquelles de bonnes performances prédictives sont aujourd'hui facilement atteignables. Sur la base du prédicteur obtenu, une MA est ensuite évaluée sur de nouvelles données. Les caractéristiques sélectionnées dépendent nécessairement du prédicteur, de la MA et de l'entrée, l'analyse étant locale. Aucune phase d'apprentissage n'est impliquée durant l'évaluation des MA : les poids du réseau sont figés et seules des prédictions sont faites (ou des calculs de gradient sans mise à jour des poids).

On pose l'hypothèse que si une MA distingue correctement les caractéristiques les plus discriminantes, ces dernières doivent impacter la qualité des prédictions si elles sont isolées ou éliminées de l'entrée. L'évaluation est effectuée en sélectionnant incrémentalement les caractéristiques ayant les contributions les plus élevées au sens de la MA afin de déterminer quelle quantité d'information est nécessaire au prédicteur pour atteindre les niveaux de performance d'une évaluation non contrainte. Cette opération est aussi appliquée dans l'autre sens en enlevant incrémentalement les pixels ou mots ayant la plus forte contribution afin d'évaluer les pertes de précision du modèle au fur et à mesure.

Puisque certaines méthodes d'attribution à base d'occlusion nécessitent de masquer une partie des entrées, cette stratégie est modifiée selon la nature de la tâche. Pour le traitement d'images, les zones retirées sont remplacées par des 0 sur tous les canaux. Pour le traitement du langage, les mots sont masqués dans l'attention puisque des modèles de type Transformers sont utilisés. Plutôt que de s'appuyer sur des pixels ou des mots individuels, des blocs sont masqués afin d'obtenir des différences significatives durant les prédictions. Pour traiter toutes les caractéristiques, le bloc ou la fenêtre est déplacé sur l'entrée jusqu'à ce que chaque caractéristique soit couverte au moins une fois. Puisque ces blocs sont assez grands et se déplacent lentement, il y a un processus de chevauchement. Tous les éléments à l'intérieur d'un bloc donné obtiennent le même score d'attribution pour le passage en cours ; ces derniers sont ensuite moyennés pour lisser les scores d'attribution. Ce protocole est similaire aux modèles d'Occlusion et DeepExplain que mettent à disposition les bibliothèques SHAP [Lundberg and Lee, 2017] et Captum [Kokhlikyan et al., 2020]. Pour assurer l'équité, la taille des blocs est réduite au voisinage des bords de l'image de sorte que chaque caractéristique soit masquée le même nombre de fois. Après avoir calculé les cartes lissées $f(\{j\})$ et $f(\mathcal{N} \setminus \{j\})$, les contributions finales sont déduites à l'aide des formules ES et FESP (voir l'équation 5.4 et la proposition 5.5).

Pour l'expérimentation, des tailles de blocs 64×64 avec un stride [5] de 8 pour les images et une fenêtre de taille 1 avec un stride de 1 pour le langage sont utilisés. Ces paramètres sont choisis car ils permettent une certaine équité entre les trois modèles ES, FESP et l'Occlusion. A noter que le choix des tailles est un hyperparamètre supplémentaire dont le choix dépend de la tâche considérée. Pour le traitement du langage, lorsqu'un mot est découpé par le tokenizer en plusieurs syllabes, la valeur maximale est utilisée pour le mot complet afin d'obtenir des résultats intelligibles. Les données utilisées sont issues de deux tâches volontairement simples : l'une relative à la classification de chiens et de chats [6], l'autre basée sur de l'analyse de sentiments (IMDB [7]). Les prédicteurs entraînés sur ces tâches sont un modèle VGG16 [Simonyan and Zisserman, 2015] et un modèle RoBERTa [Liu et al., 2019] qui obtiennent respectivement

5. Vitesse de déplacement du bloc à travers l'entrée.
6. https://www.robots.ox.ac.uk/∼vgg/data/pets/
7. https://ai.stanford.edu/ãmaas/data/sentiment/

des précisions de 99% et 95.5%. L'évaluation est effectuée sur 1024 observations exclues lors de l'apprentissage. Les résultats sont présentés en Figure 5.2.

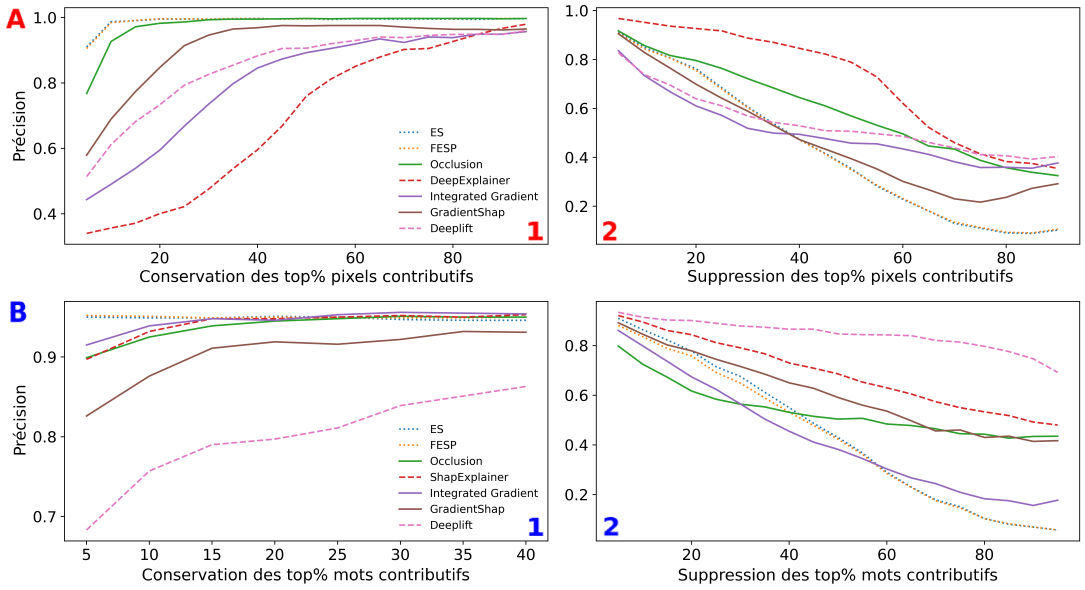


FIGURE 5.2 – Effets de la sélection des caractéristiques sur les performances des méthodes (A) Image et (B) Texte.

On observe que les modèles se basant sur le principe de l'occlusion sont en mesure d'effectuer de bonnes prédictions avec une quantité d'information très faible (A1). Dans le cadre du traitement d'images, l'extraction de 10% des pixels contribuant le plus permettent d'obtenir des prédictions dont les performances sont proches de celles mesurées lorsque le prédicteur a accès à toutes les données (précision de 99%). Cet effet est aussi observé en traitement du langage où 5% des mots suffisent (B1). En dehors des méthodes à base d'occlusion, les approches fonctionnant le mieux dans une tâche ne sont pas forcément pertinentes dans l'autre (DeepLift étant mal adapté au traitement du langage par exemple). Le comportement du prédicteur lorsque les caractéristiques les plus importantes sont enlevées incrémentalement change puisque les méthodes à base de gradient déstructurent les entrées, celles-ci bruitent l'image dès la suppression de quelques caractéristiques. L'illustration Figures 5.3 montre que FESP, ES et l'Occlusion ont un comportement proche tandis que les autres méthodes ne permettent pas toujours de déterminer ce que le réseau discrimine.

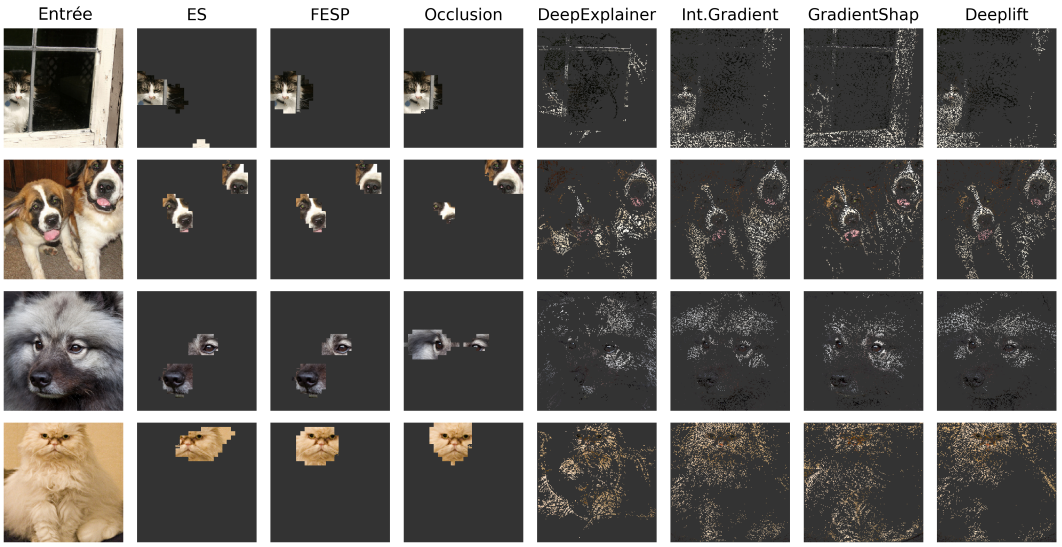


FIGURE 5.3 – Top 10% des pixels contributifs.

En traitement du langage (Figure 5.4), les extractions de l'ES et de FESP paraissent identiques. En réalité, les scores sont légèrement différents mais la normalisation de ces derniers entre 0 et 1 pour les comparer les rend difficilement distinguables. Les mots sélectionnés par les méthodes à base de gradient sont majoritairement mal appropriés à la compréhension de la prédiction. Cet effet était déjà observé dans la partie B de la Figure 5.2. Enfin, le modèle ShapExplainer extrait des passages cohérents mais exploite des règles de sectionnement internes. Cela a pour but de diminuer la complexité de cet algorithme qui se basent non plus sur des permutations globales mais locales. Les extractions obtenues sont donc particulièrement faciles à interpréter.

FIGURE 5.4 – Pondération des mots en fonction de la méthode d'attribution.

5.2.2.2 Application aux séquences longues

La section précédente a montré que la majorité des méthodes d'attribution existantes peinent à proposer des représentations facilement interprétables même pour des tâches considérées comme simples et pour lesquelles le prédicteur est capable de discriminer compte tenu de ses performances. Les approches à base de gradient ne permettent pas de comprendre les prédictions des modèles puisque les attributions calculées paraissent bruitées, approximatives et ne prennent pas en compte l'aspect local ou temporel. Pour les tâches de traitement du langage, le seul modèle montrant une certaine cohérence dans le choix des éléments discriminants est *ShapExplainer* qui exploite certaines règles de découpage prédéfinies afin de réduire la complexité de l'algorithme.

Dans les expérimentations suivantes, FESP et ES sont adaptées sur une approche similaire pour traiter de longs documents. Plutôt que de se baser sur l'occlusion de mots ou de segments de tailles identiques, de simples règles de segmentation sont ajoutées afin de diviser le document en sous-segments bien délimités. Le texte est découpé en phrases pouvant elles-mêmes être coupées en fonction d'un retour à la ligne, de la présence d'un point-virgule ou de deux points. En opérant de cette façon, si le paragraphe est composé de s segments, il suffira à l'ES s évaluations pour parcourir l'ensemble de l'entrée. Cela réduit considérablement la complexité et rend l'exercice plus cohérent par rapport aux situations où l'on supprime quelques mots dans un document qui peut en contenir plusieurs milliers et sans prendre en compte la structure.

Les tâches considérées ici reprennent les expérimentations menées dans le chapitre précédent sur les séquences longues. Sont donc mélangés des modèles capables de traiter des contextes élargis aux méthodes d'attribution proposées. La tâche est cette fois-ci beaucoup plus complexe puisque premièrement, les prédicteurs montrent des performances moyennes comprises entre 80 et 85% de précision et que dans l'exposé des faits, tous les éléments peuvent être potentiellement exploités. Enfin, les modèles se doivent d'intégrer une forme de subjectivité par rapport à certaines affaires et certaines décisions du juge car des faits d'une apparence anodine peuvent largement influencer le résultat.

L'évaluation est effectuée sur les 3 modèles qui ont montré les sélections les plus cohérentes dans la section précédente, c'est-à-dire FESP, ES et ShapExplainer. L'évaluation est spécifique ici puisque les jeux de données sont relativement petits. Nous utilisons une validation croisée en 10 étapes : un nouveau modèle est à chaque fois estimé puis l'échantillon de test est évalué en sélectionnant les 10, 20 et 30% des segments ayant le plus haut score d'attribution. Les résultats sur les circonstances de l'espèce sont présentés en Table 5.2.

Circonstances	Art. 3	Art. 5	Art. 6	Art. 8	Art. 10	Art. 13
Top 10%*						
FESP	0.761	0.775	0.782	0.740	0.762	0.848
ES	0.755	0.770	0.777	0.735	0.756	0.843
ShapExplainer	0.731	0.741	0.761	0.712	0.732	0.807
Top 20%						
FESP	0.788	0.800	0.824	0.777	0.790	0.876
ES	0.783	0.797	0.820	0.771	0.786	0.871
ShapExplainer	0.768	0.781	0.802	0.760	0.776	0.860
Top 30%						
FESP	0.789	0.804	0.829	0.782	0.797	0.885
ES	0.786	0.801	0.824	0.779	0.795	0.884
ShapExplainer	0.787	0.800	0.822	0.778	0.794	0.882
LSG-Norme**	0.790	0.806	0.830	0.784	0.802	0.888

* Sélection des top k% sous-séquences ayant le score le plus élevé.

** Modèle LSG-Norme (128, 4)

TABLE 5.2 – Performances sur les circonstances factuelles filtrées.

Le modèle FESP obtient des performances légèrement plus élevées lorsque la sélection concerne seulement les 10% des segments jugés les plus discriminants, suivi de l'ES légèrement en dessous puis de ShapExplainer. Ce comportement est proche de celui observé dans l'expérimentation précédente où ces trois méthodes finissent par converger vers des niveaux de précision semblables suite à la sélection de 30% des segments. Dans ces contextes de séquences longues,

ShapExplainer est beaucoup plus lent pour évaluer les entrées puisque le modèle perturbe les sous-segments contrairement aux deux autres approches qui ne font que masquer la zone. A noter qu'il n'y a pas de chevauchement, les segments sont traités un à un. Les résultats de l'expérimentation menée sur les faits complets, c'est-à-dire circonstance et procédure sont présentés en Table 5.3.

Faits complets	Art. 3	Art. 5	Art. 6	Art. 8	Art. 10	Art. 13
Top 10%*						
FESP	0.774	0.785	0.803	0.768	0.771	0.851
ES	0.767	0.778	0.797	0.760	0.766	0.845
ShapExplainer	0.741	0.748	0.770	0.738	0.742	0.811
Top 20%						
FESP	0.791	0.809	0.838	0.785	0.802	0.880
ES	0.787	0.802	0.831	0.780	0.794	0.873
ShapExplainer	0.772	0.791	0.851	0.770	0.783	0.862
Top 30%						
FESP	0.796	0.814	0.840	0.790	0.809	0.885
ES	0.793	0.813	0.837	0.787	0.804	0.883
ShapExplainer	0.793	0.811	0.835	0.786	0.804	0.881
LSG-Norme**	0.800	0.816	0.843	0.794	0.812	0.888

* Sélection des top k% sous-séquences ayant le score le plus élevé.

** Modèle LSG-Norme (128, 4)

TABLE 5.3 – Performances sur les faits complets filtrés.

Les observations sont identiques à l'expérimentation précédente. FESP démarre légèrement plus haut car c'est une méthode qui corrige et améliore l'ES. ShapExplainer arrive quant à lui en dernier. La sélection de 30% des sous-segments ayant les scores les plus élevés est à nouveau un seuil de convergence des performances.

Afin d'illustrer le comportement de FESP, des exemples sont extraits sur les articles 3 et 6 de la CEDH. Le premier concerne les faits de torture, de

traitements inhumains ou dégradants et le second est relatif au droit à un procès équitable. Les sélections montrent généralement une certaine subjectivité dans le choix des éléments jugés importants. Cette subjectivité reste cependant cohérente avec le résultat de l'étiquette.

Le requérant est né en 1975 et réside à Sântana.
En 2010, une procédure pénale pour évasion fiscale fut menée à son encontre.
Du 10 mai au 8 juin 2010, il fut placé en détention provisoire au dépôt de police d'Arad.
Le 8 juin 2010, il fut transféré à la prison d'Arad, où il fut détenu jusqu'au 14 décembre 2010, quand il fut remis en liberté.
La version du requérant s'agissant des conditions de détention Dans sa lettre initiale adressée à la Cour, le requérant décrivit les conditions de détention comme suit :
Au dépôt de police d'Arad, la cellule mesurait 12 m2, comportait trois lits ainsi que deux fenêtres, munies de barreaux intérieurs, qui mesuraient 40 cm x 50 cm.
Il n'y avait pas de toilettes et les détenus devaient utiliser un seau à cette fin.
L'accès aux toilettes n'était possible que deux fois par jour pour dix minutes à 6 h et à 18 h.
Les gardiens refusaient de permettre aux détenus l'accès aux toilettes en dehors de ces horaires.
L'accès aux douches était possible deux fois par semaine.
Le requérant avait le droit de sortir de sa cellule une heure par jour, dans la cour de promenade et une heure pour regarder la télévision.
La nourriture était de mauvaise qualité et le requérant n'a pas reçu d'objets d'hygiène personnelle.
S'agissant de la prison d'Arad, la cellule mesurait 16 m2 et contenait six lits superposés pour cinq personnes.
Le requérant indiqua également que plusieurs meubles s'y trouvaient (une armoire, un porte-manteau, trois tables de nuit, trois petits bancs et une table).
La nourriture était servie dans des récipients sans couvercle.
La cellule était en outre infestée de punaises et de cafards.
Le requérant avait accès à la cour de promenade pour une durée de trois heures par jour.
En outre, le transport des détenus de la prison aux tribunaux se faisait dans des conditions inhumaines, dans la mesure où les fourgons utilisés pour le transport n'avaient que deux petites fenêtres et transportaient quarante personnes, l'air devenant ainsi irrespirable.
Dans son formulaire de requête, il se plaignit des " conditions inhumaines tant au dépôt de police d'Arad qu'à la prison [d'Arad] " sans donner d'autres indications à l'exception de celles relatives aux problèmes d'accès aux toilettes au dépôt de police.
Il rappela qu'en dehors des horaires d'accès aux toilettes, les détenus devaient utiliser un seau pour satisfaire leurs besoins physiologiques.
La version du Gouvernement Au dépôt de police d'Arad, le requérant fut détenu dans plusieurs cellules de 13 m2 qu'il ne partagea qu'avec deux autres détenus.
Les cellules disposaient d'illumination naturelle et artificielle (ampoules électriques) et d'aération par les fenêtres, ainsi que de deux radiateurs qui assuraient une température entre 18o et 22o C. Les cellules ne disposaient pas de toilettes, mais les détenus pouvaient utiliser les toilettes à tout moment entre 6 h et 18 h.
Les toilettes étaient nettoyées deux fois par jour ou en cas de besoin.
La nourriture était servie trois fois par jour et faisait l'objet de contrôles systématiques ;
les détenus pouvaient en outre recevoir de la nourriture de l'extérieur.
Ils bénéficiaient de soixante minutes de promenade par jour dans la cour de promenade ou pouvaient regarder la télévision ou pratiquer des activités récréatives.
À la prison d'Arad, le requérant bénéficia d'un espace personnel d'au moins 3,5 m2.
Les cellules disposaient de toilettes, d'illumination naturelle et artificielle (ampoules électriques), d'aération par les fenêtres, ainsi que d'eau potable.
Le requérant avait accès aux douches deux fois par semaine et à la cour de promenade trois heures par jour.
La nourriture faisait l'objet de contrôles réguliers.
Les cellules ont été en outre désinfectées deux fois pendant la détention du requérant.
S'agissant des conditions de transport, elles étaient conformes aux exigences législatives et étaient ainsi adéquates.

FIGURE 5.5 – Non violation de l'article 3 de la CEDH.

La prison d'Ioannina, d'une capacité de 80 détenus, en accueillait 220 lors de l'introduction de la présente requête le 13 décembre 2008.
Dans leur requête, les requérants décrivent comme suit les conditions de vie dans la prison à la fin de 2008 : les détenus dormaient dans des couchettes réparties dans quatre grands dortoirs (occupés chacun par 32 détenus) et quatre petits (occupés chacun par 8 à 20 détenus).
Il y avait en plus des lits dans le couloir, où dormaient 45 détenus, et dans un espace ayant servi dans le passé de blanchisserie.
Aucun des dortoirs ne comportait de chaise ou de table et n'offrait le moindre espace libre.
Les détenus passaient dix-huit heures par jour enfermés dans les dortoirs – mal ventilés – où chacun disposait de 2 m2. Les dortoirs accueillaient vingt ou trente détenus qui étaient obligés de se tenir sur leurs lits.
Plusieurs d'entre eux souffraient de maladies graves pour lesquelles ils n'étaient pas traités et les détenus qui étaient en bonne santé étaient exposés à des risques de contagion, du fait de cette promiscuité.
Les malades ne bénéficiaient pas de soins satisfaisants à l'intérieur de la prison.
Les toxicomanes, les détenus souffrant de maladies chroniques et ceux dont l'état nécessite une opération ne recevaient aucun soin.
Les requérants ajoutent que la loi no 2776/1999, qui prévoit la séparation des détenus en fonction des catégories de peines, n'était pas respectée.
Les détenus purgeant une peine d'emprisonnement ou une peine de réclusion ou même des personnes en détention préventive partageaient le même espace.
A une date non précisée, les requérants et les autres détenus saisirent le médiateur de la République et remirent une pétition au conseil de direction de la prison.
D'après les requérants, tant le ministère de la Justice que la direction de la prison avaient déjà connaissance de la situation, en ayant été informés par des requêtes antérieures et le mouvement de boycott des réfectoires, déclenché par les détenus dans toutes les prisons grecques en novembre 2008.
Ils se réfèrent à une lettre adressée le 19 janvier 2008 par le médecin de la prison d'Ioannina au directeur de celle-ci qui précisait ce qui suit : " Monsieur le Directeur, [Vous avez répondu favorablement] à ma demande de pouvoir visiter les principaux espaces de détention et de séjour des détenus de la prison (...) j'ai été choqué. (..) Maintenant je comprends plusieurs choses qui me paraissaient " extrêmes " dans les données de la bibliographie internationale en matière de santé, car les conditions de vie sont effectivement " extrêmes ", voire inadmissibles pour un pays européen : – Le taux le plus élevé de traitements médicamenteux (notamment de médicaments contre l'insomnie et d'anxiolytiques) par rapport à celui mentionné dans la bibliographie internationale (20 % des détenus consomment des médicaments psychotropes en Grande-Bretagne contre 30 % à la prison d'Ioannina), avec l'insomnie comme symptôme le plus courant.
Je suis convaincu que la résistance de ce symptôme au traitement médicamenteux ordinaire (...) est due dans une large mesure aux conditions de détention (...) – Les demandes fréquentes de détenus ayant des problèmes psychologiques particuliers, susceptibles de les rendre dangereux pour eux-mêmes ou pour les autres, et souhaitant être placés dans un espace moins encombré sont traitées par leur transfert à l'hôpital psychiatrique des détenus de Korydallos [à Athènes]. Un tel transfert n'a pas d'effet thérapeutique et répond de manière insuffisante au besoin de ces personnes à disposer d'un espace personnel (...) où elles pourraient gérer leurs problèmes psychologiques. – Les problèmes fréquents résultant des piqûres de punaises et l'inefficacité des efforts pourtant généreux de la direction visant à la désinsectisation de la prison.
J'attire aussi votre attention sur la probabilité accrue d'épidémies de maladies infectieuses qui se transmettent par simple contact (hépatites A et B, tuberculose) ;
cette probabilité est accrue par l'état de surpopulation (...) Dans la même logique, il convient de noter le risque accru de maladies cardio-vasculaires, un risque difficile à gérer en raison de l'impossibilité (...) de tenir compte des exigences nutritionnelles particulières des personnes souffrant de ces maladies.
De plus, recommander de faire de l'exercice physique sonne comme une plaisanterie, compte tenu de l'encombrement de la cour de la prison. (...) Il est manifeste que plusieurs des problèmes de santé présents dans la prison d'Ioannina ont un dénominateur commun qui peut et doit changer IMMÉDIATEMENT : c'est la surpopulation de la prison, qui accueille trois fois plus de détenus que ne le permet sa capacité.
En dépit de vos bonnes intentions et de vos efforts généreux pour faire baisser le nombre des détenus à la prison d'Ioannina, j'ai le regret de vous exprimer ma conviction, fondée sur les résultats obtenus, que vos efforts sont dans une large mesure insuffisants (...) " Les requérants relèvent également que le 23 février 2009, la direction de la prison d'Ioannina rappela au procureur près le tribunal correctionnel que tant le ministère que lui-même – qui en avait fait le constat lors de ses visites – étaient au courant du problème de surpopulation de la prison.
Elle reconnaissait que la prison, construite pour 80 personnes mais en accueillant 220 en moyenne, n'était pas en mesure d'offrir aux détenus une formation professionnelle ou des activités récréatives.

FIGURE 5.6 – Violation de l'article 3 de la CEDH.

LES CIRCONSTANCES DE L'ESPCE Le requérant est né en 1936 et réside à Luxembourg.
Le 18 juillet 2003, le requérant fut assigné en divorce par son épouse.
Le Gouvernement précise, et le requérant ne le conteste pas, que l'assignation fut déposée au tribunal (" mise au rôle ") le 5 décembre 2003.
Parallèlement, des mesures provisoires furent ordonnées le 14 novembre 2003.
Ainsi, le juge des référés ordonna au requérant de quitter le domicile conjugal et de verser à son épouse un secours alimentaire mensuel de 1 500 euros (EUR).
Dans l'affaire au fond, les parties au litige déposèrent neuf corps de conclusions entre le 5 décembre 2003 et le 4 janvier 2006.
Les parties furent notamment en désaccord sur la loi applicable au litige.
L'affaire fit l'objet de six remises et fut fixée pour plaidoiries au 11 mai 2006.
A cette dernière audience, l'affaire fut mise en suspens.
Selon le Gouvernement, celle-ci intervint alors que le requérant semblait être sans mandataire.
Le requérant, en revanche, indique avoir été personnellement présent dans la salle d'audience, avec son nouveau mandataire, Maître P., qui se serait dit prêt à plaider l'affaire ;
face au refus de plaider de la partie adverse, au motif qu'une pièce du requérant n'était pas traduite de l'italien vers une des langues officielles, le tribunal aurait mis en suspens l'affaire.
Le 11 octobre 2006, le requérant adressa directement un courrier au tribunal et sollicita la " reprise de l'audience ".
Il y joignit des courriers de relance adressés les 10 juin et 11 août 2006 à Maître P. Sur demande du tribunal le 18 octobre 2006, l'avocate qui avait initialement représenté le requérant dans la procédure, Maître R., informa le juge de la mise en état qu'elle s'était déchargée de son mandat le 30 janvier 2006 ;
elle précisa qu'il lui semblait qu'elle en avait informé le tribunal à l'audience.
Sur demande du tribunal du 19 octobre 2006, Maître P. confirma, le 25 octobre 2006, être en charge du dossier.
Le 6 juillet 2007, Maître P. demanda au tribunal de " faire réappeler l'affaire " à une des prochaines audiences utiles.
Le 4 septembre 2007, il fit parvenir au tribunal sa constitution de nouvel avocat.
Le 11 octobre 2007, le juge de la mise en état délivra un échéancier aux parties, accordant en premier lieu à Maître P. un délai pour conclure au 12 novembre 2007.
Maître P. n'ayant pas conclu dans le délai, un nouvel échéancier fut délivré ;
il conclut ainsi le 18 décembre 2007.
Le délai pour conclure accordé à la partie adverse du requérant fut en conséquence reporté au 18 février 2008.
Par la suite, le juge de la mise en état délivra plusieurs injonctions aux deux parties, qui restèrent sans suite.
Ainsi, Maître P. se vit enjoindre le 22 février 2008, puis de nouveau le 22 avril 2008, de verser un certificat de résidence du requérant.
Le 6 juin 2008, le juge informa les parties que l'affaire était fixée à l'audience du 12 juin 2008 " pour conférer de l'état de la cause ".
Lors de cette audience, la partie adverse du requérant fut invitée à verser ses pièces avant le 15 septembre 2008 et fut informé des sanctions en cas de non-respect de l'échéancier.
Cette invitation étant restée sans suites, elle fut suivie d'une injonction en date du 15 octobre 2008.
Le 10 novembre 2008, le juge invita les parties à l'informer si l'affaire pouvait être clôturée.
Le 19 décembre 2008, l'affaire fut fixée pour clôture à l'audience du 2 avril 2009, à laquelle elle fut prise en délibéré.
Par un jugement du 7 mai 2009, le tribunal prononça le divorce entre les parties.
Les juges rejetèrent des moyens d'irrecevabilité et d'incompétence soulevés par le requérant.
Sur appel du requérant en date du 21 août 2009, la cour d'appel confirma, le 7 juillet 2010, le jugement de première instance.

FIGURE 5.7 – Non violation de l'article 6 de la CEDH.

Le 17 avril 1987, la requérante et Mme M. L. déposèrent un recours en référé au greffe du juge d'instance de Cammarata à l'encontre de M. L. et Mme R., visant à obtenir la constitution d'une servitude de passage dans le terrain de ces derniers.
La mise en état de l'affaire commença le 28 avril 1987, date à laquelle le juge fixa la date de la descente sur les lieux au 22 mai Le jour venu, la requérante demanda au juge de nommer un expert.
Par une ordonnance du 27 mai 1987, le juge nomma un expert qui prêta serment le 9 juin Le 14 juillet 1987, le juge demanda au greffe de solliciter le dépôt du rapport d'expertise et ajourna l'affaire au 8 août A cette date, les défendeurs demandèrent un renvoi et le juge ajourna l'affaire au 10 août Après un renvoi d'office, le 13 août 1987 le juge d'instance émit une ordonnance visant à la constitution provisoire d'une servitude de passage et fixa un délai de quatre-vingt-dix jours pour reprendre la procédure devant le tribunal, quant au fond de l'affaire.
Le 25 septembre 1987, les défendeurs reprirent la procédure devant le tribunal d'Agrigente afin d'obtenir la révocation de l'ordonnance du juge d'instance ainsi que la suspension de son exécution.
La mise en état de l'affaire commença le 6 novembre 1987, date à laquelle le juge de la mise en état demanda au greffe de verser au dossier les actes examinés par le juge d'instance et ajourna l'affaire au 27 mai Entre-temps, le 4 décembre 1987, les défendeurs avaient déposé un recours en référé devant le tribunal d'Agrigente afin d'obtenir la révocation de l'ordonnance du juge d'instance ou la suspension de son exécution.
Suite à l'audience du 15 janvier 1988, par une ordonnance du 19 janvier 1988 le juge rejeta ladite demande.
Le 27 mai 1988, la requérante demanda l'audition de témoins et le juge se réserva.
Par une ordonnance du 22 septembre 1988, le juge rejeta ladite demande et ajourna l'affaire au 3 février Cette audience fut reportée d'office au 24 novembre 1989 en raison de la mutation du juge.
Ce jour-là, l'audience fut renvoyée en raison de l'absence des parties qui n'avaient pas été informées de la mutation du juge.
Après un renvoi d'office, le 22 juin 1990 la requérante demanda une expertise.
Par une ordonnance du 28 juin 1990, dont le texte fut déposé au greffe le 2 juillet 1990, le juge nomma un expert et fixa le serment de ce dernier au 18 janvier Après un renvoi d'office, le 7 juin 1991 le juge ajourna l'affaire car l'expert n'avait pas été informé de la date de l'audience.
Des quatorze audiences fixées entre le 8 novembre 1991 et le 20 juin 1997, une fut reportée car l'expert ne s'était pas présenté, une fut consacrée à la prestation de serment de ce dernier et deux au dépôt de documents, sept audiences furent reportées car l'expert n'avait pas déposé au greffe le rapport d'expertise, une fut reportée d'office, une fut ajournée car les avocats faisaient grève et une fut reportée pour permettre aux parties de présenter leurs conclusions.
Le 24 avril 1998, les défendeurs présentèrent leurs conclusions.
Néanmoins les défendeurs demandèrent un renvoi pour verser des documents au dossier et le juge se réserva.
Par une ordonnance du 8 mai 1998, dont le texte fut déposé au greffe le 13 mai 1998, le juge rejeta ladite demande et fixa l'audience de plaidoiries au 24 février Entre-temps, la loi concernant les " sezioni stralcio " étant entrée en vigueur, l'affaire fut reportée au 18 avril 2000 pour une nouvelle audience de présentation des conclusions.
A la demande des parties, cette audience fut avancée au 6 mars L'audience du 7 avril 2000 fut consacrée au dépôt des documents et le juge ajourna l'affaire au 6 mai 2002 pour la présentation des conclusions.
Selon les informations fournies par les héritiers le 5 novembre 2001, ils n'ont pas l'intention de se constituer dans la procédure nationale.

FIGURE 5.8 – Violation de l'article 6 de la CEDH.

Conclusion

Les modèles présentés, FESP et ES, montrent plusieurs qualités en comparaison des approches existantes. Premièrement, ces deux modèles peuvent être justifiés théoriquement et possèdent tous les deux des propriétés souhaitables, que cela soit du point de vue axiomatique ou de la complexité. Ce sont de bons candidats de substitution à la valeur de Shapley. Deuxièmement, ces modèles, de

par leur construction, peuvent s'adapter aux contextes où des relations spatiales ou temporelles sont présentes dans les données. Cet aspect est particulièrement important pour expliquer des prédictions de modèles d'apprentissage profond puisque ces tâches y sont prédominantes. Contrairement aux approches plus simples par différence ou occlusion, les valeurs attribuées aux caractéristiques ont un sens et une polarité et ne permettent donc pas seulement de construire un ordre. Une valeur peut être positive ou négative pour exprimer le fait qu'une caractéristique contribue positivement ou négativement à une prédiction. Enfin, les méthodes d'attribution doivent aussi être évaluées qualitativement par un expert. Dans le cadre de textes juridiques, seule la connaissance de la décision complète permet de pleinement vérifier que les passages extraits motivent le résultat, la sensibilité des juges envers certains faits n'étant pas prévisible.

Conclusion générale

i Synthèse des contributions

Cette thèse porte principalement sur la résolution de tâches de classification de catégories de demandes et de sens du résultat dans des décisions de justice issues des juridictions françaises et européennes. Les thèmes abordés sont relatifs à des problématiques de faible volumétrie, au traitement de séquences longues et à l'interprétabilité de prédictions. La faible volumétrie est traitée en faisant référence à des modélisations de type *one-shot* couplées à de l'augmentation de données afin de réduire les risques de sur-apprentissage dans l'usage de réseaux de neurones profonds. Le traitement de séquences longues, difficile pour les Transformers standards, est effectué grâce à une modification du mécanisme d'attention. Plutôt que de connecter toutes les requêtes à toutes les clés, celui-ci exploite un ensemble de connexions locales, éparses et globales afin d'accroître significativement le contexte tout en assurant une complexité réduite. Nos expérimentations soulignent que cette architecture possède de bonnes capacités d'adaptation et d'extrapolation afin de tirer avantage de modèles préalablement entraînés sur des séquences courtes. Enfin deux méthodes d'attribution sont proposées dans le but d'interpréter les prédictions des modèles entraînés. Ces méthodes sont dérivées des approches par occlusion et de la valeur de Shapley pour en tirer plusieurs avantages. Elles permettent de maintenir une complexité linéaire tout en assurant des propriétés souhaitables de la théorie des jeux coopératifs. Contrairement aux méthodes par occlusion classiques, les scores obtenus permettent d'ordonner les caractéristiques et de déterminer leur polarité sur la prédiction.

ii Critique du travail

La volonté d'anticiper la décision d'un juge se confronte à trois problèmes difficilement surmontables. Premièrement, une décision contient une part d'aléa difficilement mesurable qui peut dépendre de facteurs multiples aussi bien juridiques que contextuels ou sociétaux. Le deuxième problème est relatif aux capacités des modèles utilisés dans la littérature en TALN. Les architectures les plus profondes telles que GPT-3 [Brown et al., 2020] ou MT-NLG [8], malgré leurs performances satisfaisantes pour une mise en production, restent en deçà des capacités humaines. Il est donc peu vraisemblable qu'en l'état des connaissances actuelles, des solutions compétitives soient découvertes pour la résolution de tâches juridiques à court terme. La dernière contrainte est relative à l'annotation des données et à la subjectivité de celles-ci. Bien que le sens du résultat ne soit pas un sujet de débat en pratique, les choix des passages dans les faits ou les motifs justifiant la décision du juge peuvent être discutés entre juristes. L'étiquetage requiert beaucoup d'investissement aussi bien en temps que dans les outils d'annotation, il est donc difficile d'organiser la création de grands jeux de données et d'assurer un consensus dans l'annotation.

Cette thèse s'est en partie focalisée sur des approches de l'état de l'art sans atteindre des niveaux de performances suffisants pour des applications pratiques. Pourtant, plusieurs difficultés ont pu être contournées, notamment en ce qui concerne le peu de données disponibles et le traitement de séquences longues. Malgré ces propositions en pratique efficaces, les performances observées sont peu satisfaisantes à l'exception des tâches d'indexation. Se posent aussi d'autres problèmes non abordés et relatifs aux aspects multi-labels de certaines tâches. Chaque décision peut être composée de plusieurs demandes ayant chacune son propre résultat. Or, la connaissance du nombre de prétentions est inconnu *a priori* et souvent difficile à déterminer.

En ce qui concerne les modèles d'attributions, il est en théorie nécessaire d'effectuer manuellement l'évaluation grâce à des experts juristes. Or en pratique, le temps à allouer pour cette évaluation est proche de celui d'une annotation

8. Voir ce blog

d'un jeu de données, il est donc difficile de la mettre en place. Quant à l'interprétation des données, celle-ci reste subjective car les scores d'attribution ne permettent pas de comprendre les interactions entre caractéristiques menant à une prédiction. Lorsque l'objectif est de déterminer le raisonnement logique opéré par le modèle, la seule connaissance des éléments discriminants n'est pas satisfaisante. Tenter de transposer un raisonnement humain à un modèle boite noire est discutable, d'autant plus lorsque celui-ci est peu performant.

Notons cependant que les contributions apportées dans le cadre de cette thèse peuvent être intégrées dans des tâches de TALN plus générales. Les ressources permettant l'utilisation de plongements sémantiques proposés et entraînés sont mis à disposition. Le mécanisme d'attention et les poids du modèle *LSG-Norm* (pour les séquences longues) sont disponibles [9] grâce à l'API de chargement dynamique d'architecture [10] d'*HuggingFace* pour les textes en anglais. Les méthodes d'attribution proposées pour l'interprétation des prédictions (FESP et ES, Chapitre 5), sont implémentées et compatibles [11] avec les modèles d'*HuggingFace* pour les tâches de TALN.

iii Pistes envisageables

La principale contrainte rencontrée dans la majorité des tâches est relative à l'accès à des données étiquetées. Il est cependant envisageable de recourir à des approches comme le *meta-learning* pour transposer la résolution d'un problème sur de nouveaux jeux de données. Il est notamment possible d'entraîner un modèle sur la prédiction du sens du résultat pour une catégorie de demande puis de l'adapter sur une nouvelle catégorie en supposant que des connaissances peuvent être partagées et qu'elles sont transférables. Cette approche peut être encore plus transversale en exploitant différentes langues ou différentes juridictions [Savelka et al., 2021]. L'autre alternative consiste en des problématiques multi-tâches pour lesquelles plusieurs tâches sont résolues ensemble suivant une architecture commune. En pratique, le *fine-tuning* de plongements sémantiques

9. https://huggingface.co/ccdv/lsg-base-4096
10. https://github.com/huggingface/transformers/pull/13467
11. https://github.com/benderama3/fesp_es

est proche de cette notion bien que les entraînements se fassent indépendamment.

Les autres aspects pouvant potentiellement accroître les performances des prédicteurs est l'utilisation de méta-données, d'entités nommées pour reconnaître les références aux parties et de la jurisprudence. Si pour le traitement d'une décision de Cour d'appel le modèle a accès à la décision de première instance, certaines caractéristiques peuvent être exploitées car des dépendances existent. Cependant, cette stratégie ne s'applique que dans certaines situations et cela suppose la transparence et l'accessibilité à ces documents.

Enfin, les méthodes d'attributions ne sont que partiellement adaptées lorsque des juristes les évaluent [Górski and Ramakrishna, 2021]. Il est notamment observé une certaine variance des réponses entre les experts en fonction des tâches. Or, si les experts eux-mêmes ne sont pas en mesure d'apporter des réponses unanimes, l'intérêt de telles méthodes est amoindri. L'utilisation d'ensembles de méthodes d'attribution semble néanmoins une piste viable pour une interprétation des prédictions plus cohérente.

Bibliographie

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv :1810.03292*, 2018.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC : Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online, November 2020. Association for Computational Linguistics. doi : 10.18653/v1/2020.emnlp-main.19.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair : An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the european court of human rights : A natural language processing perspective. *PeerJ in Computer Science*, 2, 2016.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv :1711.06104*, 2018.

Marco Ancona, Cengiz Öztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. Named entity recognition, linking and generation for greek legislation. In *Legal Knowledge and Information Systems - JURIX 2018 : The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018.*, pages 1–10, 2018. doi : 10.3233/978-1-61499-935-5-1.

Marcelo Arenas, Pablo Barceló, Leopoldo E. Bertossi, and Mikaël Monet. The tractability of shap-score-based explanations for classification over deterministic and decomposable boolean circuits. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6670–6678. AAAI Press, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv :1607.06450*, 2016.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wiserelevance propagation. *PloS one*, 10, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv :1409.0473*, 2014.

Purbid Bambroo and Aditi Awasthi. Legaldb : Long distilbert for legal document classification. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–4, 2021. doi : 10.1109/ICAECT49130.2021.9392558.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer : The long-document transformer. *arXiv :2004.05150*, 2020.

Joseph Berkson. Application of the logistic function to bio-essay. *Journal of the American Statistical Association*, 39 :357–365, 1944.

Vithor Gomes Ferreira Bertalan and E. Ruiz. Predicting judicial outcomes in the brazilian legal system using textual features. In *DHandNLP@PROPOR*, 2020.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3 :993–1022, March 2003. ISSN 1532-4435.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5 :135–146, 2017. ISSN 2307-387X.

Léon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1995.

K. Branting, B. Weiss, B. Brown, C. Pfeifer, A. Chakraborty, L. Ferro, M. Pfaff, and A. Yeh. Semi-supervised methods for explainable legal prediction. ICAIL '19, page 22–31, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367547. doi : 10.1145/3322640.3326723.

L. Karl Branting, Craig Pfeifer, Bradford Brown, Lisa Ferro, John Aberdeen, Brandy Weiss, Mark Pfaff, and Bill Liao. Scalable and explainable legal prediction. *Artificial Intelligence and Law*, 29(2) :213–238, Jun 2021. ISSN 1572-8382. doi : 10.1007/s10506-020-09273-1.

L. Breiman. Application of the logistic function to bio-essay. *Machine Learning*, pages 5—-32, 2001.

René Brink, Yukihiko Funaki, and Yuan Ju. Reconciling marginalism with egalitarianism : consistency, monotonicity, and implementation of egalitarian shapley values. *Social Choice and Welfare*, 40 :693–714, 2013.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv :2005.14165*, 2020.

Éloi Buat-Ménard. La justice dite « prédictive » : prérequis, risques et attentes - l'expérience française. *Les Cahiers de la Justice*, 2(2) :269–276, 2019. doi : 10.3917/cdlj.1902.0269.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5) : 1726–1730, 2009.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. Obligation and prohibition extraction using hierarchical RNNs. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 254–259, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi : 10.18653/v1/P18-2041.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. *arXiv :1906.02059*, 2019a.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Extreme multi-label legal text classification : A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi : 10.18653/v1/W19-2209.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert : The muppets straight out of law school. *arXiv :2010.02559*, 2020.

C. Chameni Nembua. Linear efficient and symmetric values for tu-games : Sharing the joint gain of cooperation. *Games and Economic Behavior*, 74 : 431–433, 2012.

Wen-Han Chao, Xin Jiang, Zhunchen Luo, Yakun Hu, and Wenjia Ma. Interpretable charge prediction for criminal cases with dynamic rationale attention. *J. Artif. Intell. Res.*, 66 :743–764, 2019.

Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep

learning in biology and medicine. *Journal of The Royal Society Interface*, 15 (141) :20170387, 2018.

Krzysztof Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. NIPS'17, page 218–227, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. *arXiv :2009.14794*, 2021.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv :1412.3555*, 2014.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at ? an analysis of bert's attention. *arXiv :1906.04341*, 2019.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016a.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv :1511.07289*, 2016b.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The devil is in the detail : Simple tricks improve systematic generalization of transformers. *arXiv :2108.12284*, 2021.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl : Attentive language models beyond a fixed-length context. *arXiv :1901.02860*, 2019.

Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai) : A survey. *arXiv :2006.11371*, 2020.

Emile de Maat and Radboud Winkels. A next step towards automated modelling of sources of law. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 31–39, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-597-0. doi : 10.1145/1568234.1568239.

Christophe Denis and Franck Varenne. Interprétabilité et explicabilité pour l'apprentissage machine : entre modèles descriptifs, modèles prédictifs et modèles causaux. Une nécessaire clarification épistémologique. In *National (French) Conference on Artificial Intelligence (CNIA) - Artificial Intelligence Platform (PFIA)*, pages 60–68, Toulouse, France, July 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi : 10.18653/v1/N19-1423.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron ? *arXiv :1805.12233*, 2018.

Stella Douka, Hadi Abdine, Michalis Vazirgiannis, Rajaa El Hamdani, and David Restrepo Amariles. Juribert : A masked-language model adaptation for french legal text. *arXiv :2110.01485*, 2021.

T. S. H. Driessen and Y. Funaki. Coincidence of and collinearity between game theoretic solutions. *Operations-Research-Spektrum*, 13(1) :15–30, 1991.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In M. C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1997.

Yann Dubois, Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. Location Attention for Extrapolation to Longer Sequences. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 403–413, Online, July 2020. Association for Computational Linguistics. doi : 10.18653/v1/2020.acl-main.39.

Moussa Kamal Eddine, Antoine J. P. Tixier, and Michalis Vazirgiannis. Barthez : a skilled pretrained french sequence-to-sequence model. *arXiv :2010.12321*, 2021.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2) :179–211, 1990. ISSN 0364-0213. doi : https://doi.org/10.1016/0364-0213(90)90002-E.

Emad Elwany, Dave Moore, and Gaurav Oberoi. Bert goes to law school : Quantifying the competitive advantage of access to large legal corpora in contract understanding, 2019.

Li Fei-fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 28 :2006, 2006.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values : incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33, 2020.

Yukihiko Funaki, Kees Hoede, and Harry Aarts. A marginalistic value for monotonic set games. *International Journal of Game Theory*, 26 :97–111, 1997.

Nicolas Garneau, Eve Gaumond, Luc Lamontagne, and Pierre-Luc Déziel. Criminelbart : A french canadian legal language model specialized in criminal law. ICAIL '21, page 256–257, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi : 10.1145/3462757.3466147.

Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Ingo Glaser, Elena Scepankova, and Florian Matthes. Classifying semantic types of legal sentences : Portability of machine learning models. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 121–1230, 2018.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network : Backpropagation without storing activations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Teresa Gonçalves and Paulo Quaresma. Is linguistic information relevant for the classification of legal texts ? In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, pages 168–176, New York, NY, USA, 2005. ACM. ISBN 1-59593-081-7. doi : 10.1145/1165485.1165512.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3) :50–57, 2017.

Łukasz Górski and Shashishekar Ramakrishna. Explainable artificial intelligence, lawyer's perspective. ICAIL '21, page 60–68, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi : 10.1145/3462757.3466145.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv :1410.5401*, 2014.

Hla Hart. *The Concept of Law*. Oxford University Press, 1961.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv :1512.03385*, 2015.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv :1606.08415*, 2020.

L. Hernández-Lamoneda, R. Juárez, and F. Sánchez-Sánchez. Dissection of solutions in cooperative game theory using representation techniques. *International Journal of Game Theory*, 35 :395–426, 2007.

G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786) :504–507, July 2006. doi : 10.1126/science.1127647.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *CoRR*, abs/1912.12180, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8) :1735–1780, 1997.

Thorsten Joachims. Text categorization with support vector machines : Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv :1602.02410*, 2016.

Yuan Ju, Peter Borm, and Pieter Ruys. The consensus value : a new solution concept for cooperative games. *Social Choice and Welfare*, 28 :685–703, 2007.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns : Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam : A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv :1312.6114*, 2014.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer : The efficient transformer. *arXiv :2001.04451*, 2020.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum : A unified and generic model interpretability library for pytorch. *arXiv :2009.07896*, 2020.

Kankawin Kowsrihawat, Peerapon Vateekul, and Prachya Boonkwan. Predicting judicial decisions of criminal cases from thai supreme court using bidirectional gru with attention mechanism. In *2018 5th Asian Conference on*

Defense Technology (ACDT), pages 50–55, 2018. doi : 10.1109/ACDT.2018.8592948.

Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2) :233–243, 1991. doi : https://doi.org/10.1002/aic.690370209.

I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR, 2020.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France, 07–09 Jul 2015. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 2019.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai : A review of machine learning interpretability methods. *Entropy*, 23(1) :18, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Zhenyu Liu and Huanhuan Chen. A predictive performance comparison of machine learning models for judicial cases. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, 2017. doi : 10.1109/SSCI.2017.8285436.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus,

S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

Louis Martin, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert : a tasty french language model. *CoRR*, abs/1911.03894, 2019.

Masha Medvedeva, Michel Vols, and Martijn Wieling. Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28(2) :237–266, June 2020. ISSN 0924-8463. doi : 10.1007/s10506-019-09255-y.

Masha Medvedeva, Ahmet Üstun, Xiao Xu, Michel Vols, and Martijn Wieling. Automatic judgement forecasting for pending applications of the european court of human rights. 06 2021.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv :1301.3781*, 2013.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65 :211–222, 2017.

Kody Moodley, Pedro Hernández Serrano, Gijs van Dijck, and Michel Dumontier. Similarity and relevance of court decisions : A computational study on cjeu cases. In Michał Araszkiewicz and Víctor Rodríguez-Doncel, editors, *Legal knowledge and information systems*, Frontiers in Artificial Intelligence and Applications, pages 63–72, Netherlands, December 2019. IOS Press. ISBN 978-16-4368-048-4. doi : 10.3233/FAIA190307.

Célestin Chameni Nembua and Nicolas Gabriel Andjiga. Linear, efficient and symmetric values for tu-games. *Economics Bulletin*, 3 :1–10, 2008.

Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. The eos decision and length extrapolation. *arXiv :2010.07174*, 2020.

Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1926–1934. JMLR.org, 2015.

Andrzej S Nowak and Tadeusz Radzik. A solidarity value for n-person transferable utility games. *International Journal of Game Theory*, 23 :43–48, 1994.

Conor O'Sullivan and Joeran Beel. Predicting the outcome of judicial decisions made by the european court of human rights. *arXiv :1912.10819*, 2019.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi : 10.3115/v1/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise : Randomized input sampling for explanation of black-box models. *arXiv :1806.07421*, 2018.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long : Attention with linear biases enables input length extrapolation. *arXiv :2108.12409*, 2021.

Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv :1911.02972*, 2020.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Tadeusz Radzik and Theo Driessen. On a family of values for tu-games generalizing the shapley value. *Mathematical Social Sciences*, 65 :105–111, 2013.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv :1911.05507*, 2019.

Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. Explanation methods in deep learning : Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36. Springer, 2018.

Y.L.J.A. Remmits. Finding the topics of case law : Latent dirichlet allocation on supreme court decisions. 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you ?" : Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *arXiv :2003.05997*, 2020.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5) :206–215, 2019.

Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The least square prenucleolus and the least square nucleolus. two values for tu games based on the excess vector. *International Journal of Game Theory*, 25 :113–34, 1996.

Luis M. Ruiz, Federico Valenciano, and Jose M. Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24 :109–130, 1998.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. *arXiv :1910.01108*, 2020.

Jaromir Savelka, Hannes Westermann, Karim Benyekhlef, Charlotte S. Alexander, Jayla C. Grant, David Restrepo Amariles, Rajaa El Hamdani, Sébastien Meeùs, Aurore Troussel, Michał Araszkiewicz, Kevin D. Ashley, Alexandra Ashley, Karl Branting, Mattia Falduti, Matthias Grabmair, Jakub Harašta, Tereza Novotná, Elizabeth Tippett, and Shiwanni Johnson. Lex rosetta : Transfer of predictive models across languages, jurisdictions, and legal domains. ICAIL '21, page 129–138, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi : 10.1145/3462757.3466149.

Jürgen Schmidhuber. Deep learning in neural networks : An overview. *CoRR*, abs/1404.7828, 2014.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam : Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Mehmet Fatih Sert, Engin Yıldırım, and İrfan Haşlak. Using artificial intelligence to predict decisions of the turkish constitutional court. *Social Science Computer Review*, 2021.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28) :307–317, 1953.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi : 10.18653/v1/N18-2074.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention : Attention with linear complexities. *arXiv :1812.01243*, 2020.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1) :60, Jul 2019. ISSN 2196-1115. doi : 10.1186/s40537-019-0197-0.

Avanti Shrikumar, Jocelin Su, and Anshul Kundaje. Computationally efficient measures of internal neuron importance. *arXiv :1807.09946*, 2018.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv :1704.02685*, 2019.

Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv :1712.01815*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Roos Slingerland, Alexander Boer, and Radboud Winkels. Analysing the impact of legal change through case classification. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 121–1230, 2018.

Jerrold Soh, How Khang Lim, and Ian Ernst Chai. Legal area classification : A comparative study of text classifiers on Singapore Supreme Court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 67–77, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi : 10.18653/v1/W19-2208.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1) :1929–1958, 2014.

Benjamin Strickson and Beatriz De La Iglesia. Legal judgement prediction for uk courts. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*, ICISS 2020, page 204–209, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377256. doi : 10.1145/3388176.3388183.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer : Enhanced transformer with rotary position embedding. *arXiv :2104.09864*, 2021.

Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Predicting the law area and decisions of french supreme court cases. *CoRR*, abs/1708.01681, 2017.

Yi Sun and Mukund Sundararajan. Axiomatic attribution for multilinear functions. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 177–178, 2011.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR, 13–18 Jul 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Gildas Tagny Ngompe. *Méthodes D'Analyse Sémantique De Corpus De Décisions Jurisprudentielles*. Theses, IMT - MINES ALES - IMT - Mines Alès Ecole Mines - Télécom, January 2020.

Gildas Tagny Ngompé, Sébastien Harispe, Guillaume Zambrano, Jacky Montmain, and Stéphane Mussard. Reconnaissance de sections et d'entités dans les décisions de justice : application des modèles probabilistes hmm et crf. In

Extraction et Gestion des Connaissances - EGC 2017. , Revue des Nouvelles Technologies de l'Information, Grenoble, France, January 2017.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers : A survey. *arXiv :2009.06732*, 2020.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer : Rethinking self-attention in transformer models. *arXiv :2005.00743*, 2021.

Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *arXiv :1806.02847*, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv :1807.03748*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi : 10.1145/1390156.1390294.

A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Rupali Wagh and Deepa Anand. Legal document similarity : a multi-criteria decision-making perspective. peerj computer science. *PeerJ in Computer Science*, 2020.

Bernhard Waltl, Johannes Muhr, Ingo Glaser, Elena Scepankova Georg Bonczek, and Florian Matthes. Classifying legal norms with active machine learning. In *Proc. of the 30st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 11– 20, 2017.

Lulu Wan, George Papageorgiou, Michael Seddon, and Mirko Bernardoni. Long-length legal document classification. *arXiv :1912.06905*, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue : A multi-task benchmark and analysis platform for natural language understanding. *arXiv :1804.07461*, 2019.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value : A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05) :7285–7292, Apr 2020a. ISSN 2159-5399.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer : Self-attention with linear complexity. *arXiv :2006.04768*, 2020b.

Fusheng Wei, Han Qin, Shi Ye, and Haozhen Zhao. Empirical study of deep learning for text classification in legal document review. *CoRR*, abs/1904.01723, 2019.

Jason W. Wei and Kai Zou. EDA : easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

Xiao Yan, Jinfeng Li, Xinyan Dai, Hongzhi Chen, and James Cheng. Norm-ranging lsh for maximum inner product search. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet : Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. Interpretable charge predictions for criminal cases : Learning to generate court views from fact descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, pages 1854–1864, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi : 10.18653/v1/N18-1168.

Peyton Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 29 :65–72, 1985.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird : Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc., 2020.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *arXiv :1311.2901*, 2013.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. Poolingformer : Long document modeling with pooling attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12437–12446. PMLR, 18–24 Jul 2021.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus : Pre-training with extracted gap-sentences for abstractive summarization. *arXiv :1912.08777*, 2020.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help ? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 159–168, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi : 10.1145/3462757.3466088.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi : 10.18653/v1/D18-1390.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies : Towards story-like visual explanations by watching movies and reading books, 2015.