UNIVERSITÉ DU QUÉBEC À MONTRÉAL

INJECTION DES CONNAISSANCES LINGUISTIQUES ET VISUELLES DANS LES MÉCANISMES D'ATTENTION POUR UN ENCODAGE PROFOND DU SENS DES MOTS : VERS L'INTERPRÉTABILITÉ ET L'EXPLICABILITÉ DES GRANDS MODÈLES DE LANGUE.

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR

TOUFIK MECHOUMA

DÉCEMBRE 2025

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, le Pr Ismail Biskri, pour son encadrement tout au long de cette thèse. Je souhaite également rendre hommage au très regretté Pr Jean-Guy Meunier, dont la générosité scientifique et humaine a profondément marqué mon parcours. Mes remerciements vont aussi au Pr Serge Robert, qui a accepté avec bienveillance de me diriger en fin de parcours, une marque de confiance que j'estime particulièrement. Je n'oublie pas les enseignants qui m'ont accompagné durant mes années de scolarité au DIC, ainsi que tout le personnel du département, avec une pensée spéciale pour Mylène Dagenais, dont le soutien a été précieux.

DÉDICACE

Je dédie cette thèse à mes très chers parents, à mon fils bien-aimé, Racim, source inépuisable de joie et d'inspiration, dont la présence lumineuse m'a guidé à chaque étape de ce parcours. à ma petite et grande famille ainsi que à tous mes ami(e)s.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

3-LHTN Three-level Hierarchical Transformer Network.

ALBERT A Lite BERT.

BERT Bidirectional Encoder Representations from Transformers.

BLIP Bootstrapping Language-Image Pretraining.

BoW Bag Of Words.

CLIP Contrastive Language-Image Pretraining.

Context2Vec Context to Vector.

DALL-E Deep Autoregressive Language and Latent Embeddings.

DeBERTa Decoding-enhanced BERT with Disentangled Attention.

DistilBERT Distilled BERT.

Doc2Vec Document to Vector.

DocNADE Document Neural Autoregressive Distribution Estimator.

DPM Dependency Parsing Mask.

ELMo Embeddings from Language Models.

FFNN Feed Forward Neural Network.

GPT Generative Pre-trained Transformer.

HAN Hierarchical Attention Networks.

HAND Hierarchical Attention Network for Multi-Scale Document.

HSA-RNN Hierarchical Structure-Adaptive RNN for Video Summarization.

LDA Latent Dirichlet Allocation.

lingBERT Linguistic Bidirectional Encoder Representations from Transformers.

LLMs Large Language Models.

LSA Latent Semantic Analysis.

LXMERT Learning Cross-Modality Encoder Representations from Transformers.

MHAM Multi-Head Attention Mechanism.

MLM Masked Language Modeling.

NLP Natural Language Processing.

NSP Next Sentence Prediction.

OWL Web Ontology Language.

RDF Resource Description Framework.

RDFS Resource Description Framework Schema.

RNN Recurrent Neural Network.

RoBERTa Robustly Optimized BERT Approach.

SCABERT Syntaxe Constrainte Aware Bidirectional Encoder Representations from Transformers.

SGB Syntactic Knowledge via Graph Attention with BERT.

SGBC Syntactic Knowledge via Graph Attention with BERT - Concatenation.

SGBD Syntactic Knowledge via Graph Attention with BERT - Decoder-Guided Syntax.

SpanBERT Span-based BERT.

SVD Singular Value Decomposition.

Syntax-BERT Syntax-Enhanced Bidirectional Encoder Representations from Transformers.

T5 Text to Text Transfer Transformer.

TALN Traitement Automatique du Langage Naturel.

TinyBERT Tiny Bidirectional Encoder Representations from Transformer.

VLGBERT Visual and Linguistic Bidirectional Encoder Representations from Transformers.

Word2Vec Word to Vector.

RÉSUMÉ

Cette thèse explore l'injection de connaissances linguistiques et visuelles dans les mécanismes d'attention, en particulier le modèle BERT «*Bidirectional Encoder Representations from Transformers*». Les travaux de recherche menés dans ce cadre visent à intégrer les connaissances linguistiques et visuelles afin d'améliorer l'encodage du sens des mots. Quatre articles principaux sont présentés. Le premier article propose un renforcement du mécanisme d'attention basé sur le produit scalaire utilisé par BERT, en y intégrant un masque de dépendance syntaxique. Cette approche permet de capturer les relations structurelles entre les mots, améliorant ainsi la représentation contextuelle. Le deuxième article introduit le modèle lingBERT, qui intègre des connaissances linguistiques dans l'attention via une stratégie hybride de masquage. Cette méthode combine la technique classique de masquage avec celle masquant les mots ayant des dépendances linguistiques, puis les prédit par la suite, afin d'améliorer la compréhension linguistique du modèle. Le troisième article explore l'utilisation des multiplicateurs de Lagrange dans les mécanismes d'attention, afin d'intégrer des dépendances syntaxiques via une optimisation basée sur des contraintes. L'article présente le modèle SCABERT «*Syntaxe-Constraint-Aware Bidirectional Encoder Representations from Transformers*». Celui-ci, oriente le processus d'apprentissage et permet une meilleure compréhension des relations linguistiques. Enfin, le quatrième article propose VLG-BERT « Visual and Linguistic Bidirectional Encoder Representations from Transformers » un modèle intégrant des représentations visuelles latentes multimodales dans les «*embeddings de mots*». L'approche permet d'initialiser les vecteurs de mots par leurs représentations visuelles latentes. Ce cadre vise à capturer des significations profondes en combinant des informations de différentes modalités, ce qui permet d'enrichir les représentations sémantiques et d'améliorer les performances sur des tâches variées. Dans l'ensemble, cette thèse met en évidence l'importance de l'intégration des connaissances linguistiques et visuelles pour optimiser les mécanismes d'attention, ouvrant ainsi la voie à de nouvelles perspectives en termes d'interprétabilité et d'explicabilité pour les grands modèles de langue. Bien que le titre de cette thèse évoque une orientation vers l'interprétabilité des grands modèles de langue, il est important de préciser que cette recherche ne traite pas directement les problématiques d'interprétabilité et d'explicabilité en tant que telles. L'objectif principal est de proposer un encodage plus profond du sens des mots par l'injection de connaissances linguistiques et visuelles dans les mécanismes d'attention. Néanmoins, cette approche ouvre des perspectives intéressantes pour des travaux futurs en interprétabilité, en rendant les processus d'attention potentiellement plus compréhensibles et plus alignés avec des connaissances structurées.

Remarque : Ceci est une remarque importante.
Cette thèse par articles s'inscrit dans une approche interdisciplinaire propre à l'informatique cognitive. Elle s'adresse donc à un lectorat familier à la fois avec les concepts fondamentaux de l'apprentissage automatique et du traitement automatique du langage naturel, en particulier les grands modèles de langue et les concepts cognitifs sous-jacents. Certains termes techniques largement utilisés dans la littérature sont ainsi mentionnés sans développement didactique approfondi.

CHAPITRE 1
INTRODUCTION GÉNÉRALE

1.1 Le contexte : le sens des mots en sciences cognitives et en informatique

Le sens constitue l'une des problématiques les plus complexes et les plus controversées en intelligence artificielle Rich et Knight (2009). En effet, l'étude du sens requiert une prudence particulière, car ce dernier fait l'objet d'études dans de multiples disciplines, telles que la logique, la linguistique, la sémiotique, la philosophie, etc. Chacune de ces disciplines a en effet ses propres définitions et arguments Thérien (1989). Bien qu'il existe un alignement sémantique entre ces disciplines, celui-ci se présente souvent dans un vocabulaire propre à chacune d'entre elles Rastier (1996). La question du «*sens du sens*» est une problématique de longue date qui demeure d'actualité en sciences cognitives Pylyshyn (1984); Jordan *et al.* (2021). L'analyse du sens, dans la langue en particulier, est à l'origine de nombreuses approches théoriques. Celles-ci peuvent être catégorisées en fonction de la manière dont elles perçoivent le sens Aitchison (1987). D'un côté, la première catégorie s'intéresse au décodage individuel du sens. Elle fait référence à l'idée que le sens des mots, des phrases ou des expressions est principalement le produit de la cognition individuelle. Elle repose sur l'hypothèse que chaque individu attribue un sens à l'aide de ses mécanismes cognitifs et de ses représentations mentales. Dans cette perspective, le sens est souvent perçu comme objectif et indépendant des interactions sociales. De l'autre côté, la deuxième catégorie étudie l'attribution sociale du sens. Elle se concentre davantage sur la langue comme un phénomène social, et postule que le sens des symboles émerge à travers des interactions sociales et des conventions collectives Saussure (1916); Barthes (1972). En d'autres termes, le sens ne se limite pas à un décodage individuel : il émerge également des pratiques socioculturelles partagées. Cette double nature du sens cognitive et socioculturelle a initialement été explorée dans le champ des sciences cognitives, mais elle a aussi profondément influencé le domaine de l'informatique, notamment dans le développement de modèles visant à représenter le sens des mots et des énoncés. Ces modèles illustrent comment des systèmes artificiels peuvent encoder le sens dans des contextes particuliers.

Dans cette introduction, nous explorons les principales théories du sens issues des sciences cognitives, ainsi que les approches informatiques qui ont tenté de formaliser et de modéliser l'encodage de la langue. Trois grandes catégories d'approches sont distinguées : les approches symboliques classiques, les approches statistiques, et les approches connexionnistes. Les approches symboliques classiques reposent sur l'idée que

la connaissance, y compris le sens des mots, peut être représentée à l'aide de structures symboliques explicites et manipulables par des règles logiques. Comme l'expliquent Newell et Simon (1976), la cognition humaine elle-même peut être vue comme une forme de manipulation symbolique. Des formalismes tels que les réseaux sémantiques Quillian (1968), les frames Minsky (1974), ont grandement influencé les débuts de l'intelligence artificielle Russell et Norvig (2010), présentent des limites notables, notamment leur incapacité à gérer la variabilité linguistique et contextuelle sans intervention humaine explicite. Face aux limites des approches symboliques, les approches statistiques ont émergé avec l'accroissement des données textuelles numériques et des capacités de calcul. Ces méthodes s'inscrivent dans le courant distributionnaliste, inspiré des travaux de Harris (1954); Firth (1957), selon lesquels «*You shall know a word by the company it keeps*». Les premiers modèles de type «*n-grammes*» Shannon (1948), les matrices de co-occurrence Landauer et Dumais (1997), ou GloVe Pennington *et al.* (2014), traduisent cette approche. Elles permettent de détecter des similarités sémantiques latentes en se basant sur des modèles d'utilisation, sans modélisation explicite du sens. Enfin, les approches connexionnistes, principalement incarnées par les réseaux de neurones artificiels, encodent le sens des mots à travers des architectures neuronales Mikolov *et al.* (2013). Aujourd'hui, l'essor des réseaux profonds, notamment des transformeurs Vaswani *et al.* (2017), a permis une contextualisation dynamique du sens dans des modèles comme BERT Devlin *et al.* (2019) ou GPT Brown *et al.* (2020a). Ces modèles apprennent des représentations hautement performantes pour une variété de tâches linguistiques, mais soulèvent aussi des enjeux critiques liés à l'opacité, à la généralisation, et à la compréhension réelle du langage Bender et Koller (2020).

Dans les sections suivantes, nous discuterons plus en détail de ces trois familles de modèles, en présentant leurs fondements théoriques, leurs mécanismes d'apprentissage, leurs points forts et leurs limites pour saisir la richesse et la complexité du sens humain.

1.1.1 En sciences cognitives

Les sciences cognitives ont étudié en profondeur le sens des mots, leur représentation et leur traitement par les êtres humains. Ces avancées sont le fruit des progrès technologiques et des découvertes en neurosciences. Ces théories ont évolué, passant de «*modèles symboliques classiques*» rigides à des approches plus dynamiques, distribuées et nuancées. Cette transition reflète l'évolution historique des théories du sens, en fonction des contextes philosophiques, linguistiques, cognitifs et sociétaux. Fodor (1975); Harnad (1990); Barsalou (1999); Chomsky (2000); Pulvermüller (2013). Dans le cadre de cette thèse, qui vise à approfon-

dir l'encodage du sens par l'injection conjointe de connaissances linguistiques et visuelles, il est nécessaire d'exposer brièvement les grands courants qui ont marqué la réflexion sur le sens. La notion de sens a en effet fait l'objet de nombreuses interprétations dans l'histoire de la pensée, en philosophie, en linguistique, en sémiotique et en sciences cognitives. Chaque courant a proposé une manière particulière de concevoir ce que signifie «*avoir du sens*», en fonction de ses présupposés épistémologiques, de ses outils conceptuels et de ses objectifs théoriques. Ainsi, sans nous inscrire pleinement dans chacun de ces cadres, nous proposons un survol de quelques grandes théories du nominalisme et du réalisme médiévaux, au structuralisme, en passant par le pragmatisme, la sémantique formelle, la sémantique distributionnelle et le cognitivisme. Ce parcours a pour objectif de montrer la diversité des conceptions du sens, leur évolution dans le temps et leur influence notre compréhension actuelle du langage. C'est dans ce contexte que s'inscrit la perspective adoptée dans cette thèse qui s'appuie plus spécifiquement sur le modèle sémiotique de «*Charles Sanders Peirce*», appartenant au courant pragmatiste. Ce modèle, fondé sur la triade «*signe*, *objet*, *interprétant*», offre un cadre théorique particulièrement fécond pour penser l'articulation entre représentations symboliques, perceptives et conceptuelles.

1.1.1.1 Le nominalisme et le réalisme

La question du sens des mots tire son origine de «*la philosophie médiévale du Moyen Âge*» Marenbon (2007). Des penseurs réalistes comme «*Platon*» et «*Aristote*» soutiennent que le sens n'est pas seulement une construction linguistique ou une convention humaine, mais qu'il s'incarne réellement et objectivement dans le monde Mohr (1981); Sokolowski (1964). Les adhérents de ce courant, appelé «*réalisme*», admettent que les mots que nous utilisons pour communiquer, interagir et décrire le monde ont des entités indépendantes de notre perception et des concepts généraux dits «*universaux*» Armstrong (1989). Le «*réalisme*» part du postulat que le monde est fondamentalement différent de nos représentations de celui-ci. De plus, le «*réalisme*» considère que l'existence du monde précède l'émergence de nos représentations Panaccio (2004). En réalisme, les mots sont considérés comme des ponts reliant l'esprit humain à des vérités existantes dans le monde. Par exemple : «*Je vois bien l'arbre*, *mais je ne vois pas l'arbrité*» Mohr (1981). Les «*réalistes*» soutiennent que le mot «*arbre*» ne dénote pas seulement un ensemble de plantes que nous appelons arbres, il fait plutôt référence à une réalité universelle qui est «*l'arbrité*», comme étant une essence véridique partagée par tous les arbres Cross (2005). Cette conception réaliste du langage naturel considère celui-ci comme étant une «*fonctionnalité ontologique*» qui véhicule et révèle des vérités intrinsèques du monde Sokolowski (1964). Contrairement au réalisme, le nominalisme trouve que les concepts généraux,

ou «*universaux*», que nous décrivons à l'aide de mots, ne sont que des termes, des étiquettes ou encore des noms que nous donnons à des catégories d'entités similaires Marenbon (2007). Les nominalistes, à l'image de «*Guillaume D'Ockham*», estiment que les mots et les concepts généraux ne peuvent être que des outils permettant de décrire et d'organiser le monde. Ils affirment que les «*universaux*» n'ont pas de référent dans le monde Panaccio (2004). Le débat mené par ces deux courants vise à clarifier la relation entre les concepts linguistiques et la réalité du monde Armstrong (1989). Les répercussions de ce débat sur l'intelligence artificielle, se manifestent dans l'interrogation portant sur les catégories sémantiques, afin de savoir si celles-ci ont une existence réelle dans le monde ou s'il s'agit de conventions humaines van Inwagen (2004). Dans le domaine du «*traitement automatique du langage naturel TALN*», le nominalisme peut se manifester dans des modèles contextualisés comme «*BERT*» et «*GPT*» Devlin *et al.* (2019); Brown *et al.* (2020a). Ces modèles n'encodent le sens d'un mot que dans un contexte spécifique, excluant ainsi l'idée d'un sens universel. Toutefois, il existe des modèles d'apprentissage qui marient la langue à la vision, rapprochant ainsi l'encodage du sens des mots à une perspective réaliste qui vise à apprendre des représentations latentes correspondant à des «*universaux*» Rahman *et al.* (2020); Ramesh *et al.* (2022).

1.1.1.2 Le structuralisme

Le «*structuralisme*» est apparu entre les XIX^{e} et XX^{e} siècles. Il a été initié par le linguiste «*Ferdinand de Saussure*». Ce dernier met l'accent non seulement sur les éléments linguistiques pris individuellement, mais aussi sur les relations entre ces éléments. Il a notamment introduit les termes clés du structuralisme, comme le «*signifiant*» pour désigner la forme matérielle dénotant le mot « *arbre*». Ce dernier peut être le son ou encore la chaîne de lettres qui le forme. Le «*signifié*» désigne quant à lui la représentation que l'on obtient du mot «*arbre*». Selon le structuralisme, la relation entre ces deux notions n'est qu'une convention sociale. Cette dernière ne repose sur aucune logique régissant l'association d'une idée particulière, qu'il s'agisse d'une forme sonore ou graphique. Exemple : il n'y a aucune raison intrinsèque pour que l'idée d'un arbre soit exprimée par le mot «*arbre*». «*Saussure*» perçoit la langue comme un système de signes linguistiques. Selon lui, l'analyse de la langue repose sur deux types de relations au sein de ce système. Les relations «*syntagmatiques*», qui sont des combinaisons d'unités, et les relations«*paradigmatiques*», qui sont des combinaisons verticales d'unités. Le premier type porte sur l'ordre de mots dans une séquence, tandis que le deuxième étudie les substitutions «*paradigmatiques*» possibles entre des unités linguistiques occupant une position similaire dans une phrase. Les modèles modernes de traitement de la langue reposent sur les représentations vectorielles de mots appelées «*embeddings*». Ils associent une séquence de caractères

le «*signifiant*» à une représentation numérique permettant ainsi l'encodage du sens des mots dans un espace vectoriel qui pourrait être perçu comme étant le «*signifié*» d'un point de vue structuraliste. En outre, les mécanismes d'attention permettent un apprentissage dynamique du sens d'un mot en fonction de son contexte linguistique. Cette capacité flexible est nuancée à encoder le sens d'un mot renforce l'idée «*saussurienne*» que le sens est relationnel et dépendant des autres termes du système linguistique Saussure (1916).

1.1.1.3 Le pragmatisme

À l'instar des théories précédentes, le pragmatisme a ses propres définitions et ses particularités qui expliquent l'origine du sens des mots. Selon ce courant de pensée, le sens des mots est le produit d'une expérience ayant des répercussions issues de leur pratique. Cette expérience, qui est indispensable, constitue un aspect très important du pragmatisme, car elle contribue à la construction du contexte Peirce (1878); Dewey (1938). «*Charles Sandres Peirce*» est l'un des fondateurs de ce courant à la fin du XIXe siècle. Pour lui, les mots sont des éléments et des outils indispensables pour agir, résoudre un problème ou atteindre un objectif Peirce (1958). Les mots servent de guide pour appréhender le monde et mener à bien une action ou vivre une expérience. Le pragmatisme met en évidence la dynamique du sens. Autrement dit, le sens est en évolution constante avec le contexte Peirce (1878); Dewey (1938). À titre d'exemple, le mot «*cheval*» a toujours désigné un animal de compagnie. Cependant, avec l'évolution de l'informatique, ce terme peut désormais faire référence à un programme malveillant s'il est associé à la ville de «*Troie*». «*Peirce*» conçoit sa triade sémiotique à l'aide du «*représentamen*», de l'«*objet*» et de l'«*interprétant*». À la différence du «*signifiant*» dans l'unité duale linguistique chez «*Saussure*», «*Peirce*» recense trois types de «*représentamen*». Le premier est appelé «*indice*», celui-ci est souvent lié à une relation causale par exemple, le sang résultant d'une blessure. Le deuxième est appelé «*icône*». Il se caractérise par sa ressemblance avec l'«*objet*» auquel il se rapporte. Le troisième le «*symbole*» exprime une convention partagée entre des individus, comme le code de la route Peirce (1878); Dewey (1938); Ogden et Richards (1923). En sémiotique, la «*triade*» de «*Peirce*» est considéré comme un moyen d'accéder au sens Ogden et Richards (1923). Les modèles multimodaux modernes du traitement automatique du langage naturel s'alignent relativement avec cette «*triade*». Des modèles récents comme «*CLIP et DALL-E*» intègrent des données textuelles et visuelles en même temps afin d'apprendre à encoder le de mots. Ceci renforce les idées «*peirciennes*» en reliant la langue à des expériences du monde réel. De son côté, «*John Dewey*» met en avant le lien entre langue, pensée et action dans des contextes sociaux et éducatifs. Selon lui, la langue sert principalement à résoudre des problèmes

pratiques et à organiser l'activité humaine Dewey (1916). Cette perspective permet d'explorer l'encodage du sens des mots dénotant des actions et permet de tester ces modèles non seulement sur leurs capacités à prédire ou à générer du texte, mais aussi sur leur aptitude à guider et à mener à bien des actions pratiques, comme dans le domaine de la « *robotique*».

1.1.1.4 La sémantique formelle (philosophie analytique, empirisme logique)

Contrairement aux courants précédents, la «*sémantique formelle*» aborde le sens des mots sous différents angles, en s'appuyant sur la logique et sur des théories relevant de la «*philosophie analytique*». «*Gottlob Frege*», philosophe, mathématicien et surtout logicien distingue dans sa «*théorie du sens et de la référence*», le référent d'un mot et son sens Frege (1892). Selon cette théorie, dire que « *Donald Trump*» est l'un des ex-présidents des «*États-Unis*» revient à dire que «*Donald Trump*» est un homme d'affaires. Pour «*Frege*», il est important de distinguer le sens d'un mot de la représentation mentale qu'il peut évoquer. Selon lui, le sens est une entité objective et commune à tous et qui détermine la référence du mot dans le monde. En revanche, la représentation mentale est une image subjective, propre à chaque individu et susceptible de varier d'une personne à l'autre. Le sens ne correspond donc pas à la manière dont une personne imagine un concept, mais à un mode de présentation stable et partagé par les locuteurs, qui permet la communication et la compréhension mutuelle. «*Rudolf Carnap*» introduit ensuite les notions d'«*extension*» et d'«*intension*» Carnap (1947). Il définit l'extension comme l'ensemble de toutes les entités dont le mot fait référence dans le monde. Exemple : le mot «*plante*» désigne toutes les plantes sur terre. En revanche, l'intension désigne l'ensemble des propriétés d'une entité faisant partie de l'extension. Ces mots décrivent des propriétés structurelles ou fonctionnelles permettant d'identifier un objet ou une entité appartenant à cette catégorie. Par exemple, l'intension du mot «*plante*», est un organisme vivant qui effectue la photosynthèse, possède des racines, une tige et des feuilles, et appartient au règne végétal. La théorie de «*Carnap*» peut, dans un certains cas, être transposée aux modèles du traitement automatique du langage naturel. L'extension se manifeste notamment dans le processus d'apprentissage des «*entités renommées*». Les phrases utilisées pour l'entraînement contiennent des mots décrivant les propriétés d'un objet, d'une entité ou d'un concept. Ces phrases pourraient constituer une intension, dont le but est d'apprendre une représentation vectorielle qui correspond à une extension spécifique dite «*entités renommées*». De la même manière, l'intension apparaît dans la prédiction du mot suivant, comme c'est le cas avec «*GPT*» Brown *et al.* (2020a). Après le mot «*cheval*», «*GPT*» prédit le mot court, qui est une propriété du cheval. Par ailleurs, au début du XIXe siècle, les logiciens ont considéré la logique formelle comme un outil rigoureux permettant aux

sciences empiriques de développer des connaissances Russell et Whitehead (1913). Grâce aux travaux de «*Frege et Russell*», repris ensuite par le «*Cercle de Vienne*» Carnap *et al*. (1929), la logique est devenue un moyen de découverte scientifique. Elle a également été utilisée comme moyen de modélisation de la langue Goodman (1954). Les empiristes logiques ont utilisé des systèmes logiques formels capables de représenter des propositions et des énoncés à l'aide de symboles mathématiques et de règles syntaxiques précises, afin de modéliser certains aspects de la langue. Cette modélisation logique visait à désambiguïser la langue à l'aide de formulations rigoureuses et exactes, c'est-à-dire à formaliser la signification des mots et des phrases à l'aide de la logique symbolique. Dans un premier temps, les logiciens ont utilisé la logique des propositions pour modéliser la langue. Cette dernière repose sur des propositions interconnectées par un ensemble de connecteurs logiques qui définissent son système formel et ses règles d'inférence.
Par exemple, considérons l'énoncé : «*Il fait soleil et il fait froid.*»
On peut le décomposer en deux propositions atomiques P : «*Il fait soleil.*» et Q : «*Il fait froid.*»
La formulation logique correspondante en logique propositionnelle est : $P \wedge Q$ «*La logique propositionnelle*» a ensuite été enrichie pour donner naissance à «*la logique des prédicats*» qui permet d'exprimer des relations entre objets du monde. Contrairement à « *La logique propositionnelle* » , elle introduit des «*prédicats*», des «*variables*» et des «*quantificateurs*» pour modéliser des énoncés plus complexes Hintikka (1962); Prior (1957).

Par exemple, l'énoncé : «*Tous les corbeaux sont noirs*» peut être formalisé ainsi :
$H(x)$: « x est un corbeau »
$M(x)$: « x est noir »
Formulation : $\forall x\big(H(x) \rightarrow M(x)\big)$.

1.1.1.5 La sémantique distributionnelle (le distributionalisme)

Le « *distributionnalisme* » est un courant de pensée apparu aux «*États-Unis*». Il se caractérise par une accentuation du contexte grammatical et syntaxique de la langue, sans se préoccuper du sens intrinsèque des mots. Il se concentre sur l'étude de l'ordre des mots et des règles qui régissent une langue, sans s'intéresser à la dimension sémantique profonde. Le distributionnalisme est une approche empirique, car il repose sur l'observation directe d'unités linguistiques mesurables, telles que les mots, les phonèmes et les phrases. Il rejette donc les notions de sens et de concept, qu'il juge trop abstraites. L'analyse distributionnelle em-

pirique est donc une «*analyse inductive*», car elle consiste à faire des observations permettant d'induire des règles décrivant le comportement syntaxique de la langue. Plusieurs ouvrages mentionnent que le problème du contexte renvoie à l'hypothèse distributionnelle de «*Zellig Harris (1954)*». Cette dernière postule que les mots qui apparaissent dans des contextes similaires ont des propriétés linguistiques similaires. Harris (1954). Elle a ensuite été généralisée par «*John Rupert Firth*» en 1957. Ce dernier considère que le sens d'un mot est déterminé par son contexte lexical. Firth (1957). Bien que la sémantique distributionnelle trouve son origine dans les travaux de «*Harris et Firth*», ce terme n'a été adopté qu'à partir des années 1980 et 1990. Ce terme a donné lieu à des approches linguistiques et computationnelles qui utilisent des modèles vectoriels pour représenter le sens des mots. On la retrouve dans des modèles purement statistiques qui calculent les co-occurrences entre les mots et leurs contextes à l'aide de matrices représentant les relations distributionnelles. Turney et Pantel (2010). On la retrouve également dans d'autres modèles neuronaux fondés sur l'apprentissage automatique. La sémantique distributionnelle fournit une base théorique et pratique pour représenter le sens des mots en fonction de leur contexte. Elle constitue le fondement théorique de plusieurs modèles de plongement lexical, tels que «*Word2Vec*, *GloVe et FastText*» Mikolov *et al.* (2013); Pennington *et al.* (2014); Bojanowski *et al.* (2017).

1.1.1.6 La sémantique cognitive (le cognitivisme)

Le «*courant cognitiviste*» propose, quant à lui, une vision complètement différente de celle de la «*sémantique distributionnaliste*». Le «*Cognitivisme*» s'intéresse en effet à la pensée humaine et aux processus cognitifs, qu'il considère comme un paradigme fondamental pour la compréhension et la production de la langue Newell et Simon (1972). L'une des distinctions du cognitivisme est que le sens d'un mot dépend non seulement du contexte, mais aussi des connaissances, des expériences et des représentations que les individus lui attribuent. Les travaux «*d'Allen Newell et Herbert Simon*» ont marqué les origines du cognitivisme symbolique. À ses débuts, le cognitivisme avait un caractère symbolique et comparait l'esprit à un ordinateur. Il postule que la pensée humaine n'est qu'un traitement d'informations symboliques, semblables à celles traitées par un programme informatique. «*Jerry Fodor*», l'un des piliers de la théorie de la «*modularité de l'esprit*», considère la cognition comme étant une représentation symbolique permettant le traitement des concepts abstraits. Dans son ouvrage intitulé «*The Language of Thought*», il introduit un langage symbolique mental structuré, qu'il appelle «*Mentalese*» et qu'il considère comme la forme de la pensée humaine Fodor (1975). «*Roger Schank*», un autre pilier du modèle cognitif symbolique appliqué à la compréhension de la langue, soutient quant à lui que la compréhension humaine repose sur la manipu-

lation de structures symboliques représentant le savoir et l'expérience humaine. Ses travaux portent sur la manière dont les êtres humains interprètent le monde à l'aide de symboles. Cette conception symbolique de la pensée humaine a toutefois été critiquée par plusieurs cogniticiens, notamment «*Marvin Minsky*», qui souligne que l'on parle d'une intelligence artificielle, néanmoins incapable d'apprentissage perceptif, d'organisation de la mémoire ou encore de raisonnement critique humain Schank et Abelson (1977); Minsky (1986).

De son côté, «*John Searle*», philosophe du langage et de l'esprit, argumente, à partir de son expérience de la «*Chambre chinoise*», que l'intelligence artificielle ne peut qu'être qu'une intelligence faible Searle (1980). Il justifie cette position en affirmant que la simplie manipulation de symboles par une machine ne suffit pas à créer une véritable compréhension ou une véritable conscience. Il préconise d'associer la cognition symbolique à une dimension plus profonde que la simple manipulation de symboles. Ces critiques ont motivé l'introduction du terme «*concept*». Il est en effet presque impossible d'aborder le sens des mots sans évoquer les concepts. Ces derniers sont considérés comme les briques de base de la construction du sens Lakoff (1987); Fodor (1998); Evans et Frankish (2009). Toutefois, la définition d'un concept varie d'une communauté à une autre. Dans la sémantique formelle, par exemple, le terme «*concept*» désigne une dénotation symbolique de la pensée. En logique des propositions et des prédicats, par exemple, il est indispensable de représenter les concepts qu'ils soient du monde réel ou abstraits à l'aide d'un modèle sémantique formel pour pouvoir les manipuler. De même, les ontologies nécessitent la représentation des concepts à l'aide de symboles. D'autre part, dans la sémantique cognitive, le terme «*concept*» fait l'objet d'un dialogue interdisciplinaire en sciences cognitives abordant des notions telles que la conceptualisation et la catégorisation. «*Ray Jackendoff*» estime que demander ce qu'est un «*concept*» à un psychologue, un philosophe ou un linguiste revient à interroger un physicien sur ce qu'est la masse : une réponse isolée ne peut être fournie Jackendoff (1983).

Pour sa part, «*Jess Prinz*», pense que sans les concepts, les pensées ne peuvent pas exister, et que par conséquent, la langue n'a rien à exprimer Prinz (2002). L'anthropologue «*Benjamin Lee Whorf*» et son élève «*Edward Sapir*» limitent la pensée et la connaissance du monde à la maîtrise de la langue. Selon eux, la langue est la clé de voûte de l'interprétation et de la représentation mentale du monde Whorf (1940); Sapir (1985). Cette hypothèse a toutefois été confrontée au problème de la variation des représentations et des catégorisations du monde selon les langues, problème introduit en linguistique par «*Wilhelm von Humboldt*» Whorf (1940); Sapir (1985). De son côté, «*Searle*» écrit que les concepts, au sens philosophique sont

exprimables par la langue. Néanmoins, ils ne se limitent pas aux lexèmes Searle (2006). «*Stevan Harnad*» propose l'«*ancrage symbolique*» comme approche pour attribuer un sens aux mots. Il conditionne l'acquisition du sens à la capacité d'attribuer des référents aux mots, ainsi qu'à la nécessité de la conscience Harnad (1990). En psychologie, on distingue souvent les concepts concrets et abstraits. Un concept concret possède des référents perceptibles, comme les fleurs, les arbres ou les chats. Un concept abstrait, en revanche, ne satisfait pas cette définition. En sciences cognitives, il y a deux courants principaux concernant les concepts. Le premier courant «*Fodor, Jackendoff, Pinker, Pylyshyn*» considère les concepts comme des entités abstraites et complètement séparées du système sensorimoteur. Il s'agit d'une vision amodale de la cognition Barsalou *et al.* (2008). Le second courant ancre les concepts dans les états cérébraux perceptifs et intéroceptifs. Il assume une continuité entre le système sensorimoteur et les concepts. Les principaux représentants de ce courant sont «*Barsalou, Lackoff, Damasio et Evans*» Barsalou *et al.* (2008); Petito *et al.* (2000); Damasio et Damasio (1994); Lakoff (1987). Le cogniticien «*Peter Gärdenfors*» avance que nos mots expriment nos concepts Gärdenfors (2019), tandis que «*Paul Chilton*» explore l'aspect spatio-temporel du sens linguistique, en proposant une vision conceptuelle et géométrique du sens Chilton (2013). Certains chercheurs, notamment en psychologie cognitive, privilégient l'usage du terme «*catégorie*». «*Ludwig Wittgenstein*» considère la catégorisation comme un acte d'interprétation : reconnaître un objet X revient à l'identifier par son appartenance à une catégorie Y plutôt que par ses seules propriétés intrinsèques Wittgenstein (2001). En 1973, «*Eleanor Rosch*» propose la théorie des prototypes. Elle avance que certains membres d'une catégorie sont plus centraux que d'autres. Cette théorie s'oppose à «*la conception aristotélicienne classique*» qui définit une catégorie par des conditions nécessaires et suffisantes. Elle définit la catégorie comme un ensemble de cas typiques «*prototypes*» et leurs variantes. Cette théorie a toutefois été remise en question par l'analyse des catégories lexicales, qui remet en cause la centralité de certains concepts. Par exemple, le verbe penser ne peut être considéré comme plus ou moins central. La notion de prototype est associée à «*Wittgenstein*», qui, avec sa théorie de la ressemblance familiale, montre que les gens regroupent les concepts selon plusieurs caractéristiques partagées, plutôt qu'une seule Wittgenstein (2001). «*Gärdenfors*», avec sa théorie des espaces conceptuels multidimensionnels, il tente d'expliquer les prototypes par la convexité de ses espaces. Il définit une catégorie selon une distance conceptuelle : si A et B appartiennent à une catégorie, et que C se situe à une distance intermédiaire entre les deux, alors C appartient aussi à cette catégorie Gärdenfors (2000).

1.1.2 En informatique

En «*traitement automatique du langage naturel TALN*», on distingue trois approches permettant de traiter le sens des mots et d'automatiser la langue à l'aide d'une machine. La première catégorie est celle des «*approches symboliques*», qui nécessitent la maîtrise d'un savoir-faire exprimé a priori sous forme de règles ou de formalismes. Ces approches sont généralement rigides, coûteuses et subjectives. La deuxième catégorie est celle des «*approches statistiques*» basées sur des modèles statistiques non supervisés. Elles utilisent la co-occurrence comme principale métrique d'observation et d'analyse. Ces techniques permettent d'extraire des informations et de construire des représentations des textes sans avoir besoin de règles explicites. La troisième catégorie est celle des «*approches connexionnistes ou neuronales*». Ces dernières interagissent directement avec les données. Elles sont orientées données et ne nécessitent pas de savoir-faire exprimé a priori.

1.1.2.1 Approches symboliques en traitement automatique du langage naturel

Le symbolisme est considéré comme l'une des formes de l'intelligence artificielle classique. La communauté scientifique utilise également l'expression «*approches orientées règles*» ou «*rules-based*» en anglais. Le symbolisme vise à encoder les concepts et les événements appartenant à notre environnement à l'aide de symboles. L'objectif de cet encodage est de pouvoir représenter et raisonner sur ces derniers à l'aide de modèles logiques. À l'instar de toute autre discipline de l'intelligence artificielle, le traitement automatique du langage naturel est également influencé par le symbolisme classique. La logique formelle est à la base de tous les modèles computationnels qui l'adoptent comme fondement. Elle est utilisée dans différentes tâches de traitement automatique du langage naturel, comme l'analyse sémantique permettant de déterminer le sens des phrases et les relations entre leurs mots, ou encore le raisonnement automatique sur des textes, la traduction automatique et les tâches conversationnelles pour répondre à des questions en langue naturelle. Russell et Norvig (2010). En 1956, «*Richard H. Richens*», de l'université de Cambridge, utilisait le calcul logique propositionnel pour traduire des textes en langue naturelle. Il a proposé une méthode fondée sur le calcul logique pour analyser et traduire les structures linguistiques. Selon «*Richens*», la logique propositionnelle pouvait être utilisée pour représenter les relations entre les mots et les concepts dans une phrase, permettant ainsi une traduction plus rigoureuse et systématique entre les langues. Richens (1956). en 1960, «*Victor H. Yngve*» proposait pour sa part son premier modèle de génération automatique de phrases bien structurées, s'inspirant de la «*grammaire générative de Noam Chomsky*». Ce modèle repose sur une formalisation grammaticale des langues naturelles, s'appuyant sur des concepts de «*grammaires*

génératives» pour produire des structures syntaxiques correctes. Le travail de «*Yngve inspira Robert F. Simmons et Sheldon Klein*» qui appliqua la logique des prédicats aux réseaux sémantiques en 1963. «*Simmons et Klein*» avaient pour objectif la représentation formelle des relations sémantiques entre concepts. Ils utilisaient la logique des prédicats pour structurer l'information dans des réseaux sémantiques. Ces réseaux consistaient en des graphes où les nœuds représentaient des concepts ou des entités de la langue Simmons (1963). En plus des travaux précédents, ceux de «*Yehoshua Bar-Hillel*» ont eu un impact significatif sur le développement de la «*linguistique computationnelle*». Il a été l'un des premiers à explorer les liens entre la «*logique modale*» et la linguistique. Ses recherches visaient à formaliser des concepts linguistiques complexes à l'aide d'outils logiques ouvrant ainsi la voie à des analyses plus rigoureuses de la langue. Il fut également l'un des premiers à tenter de concevoir des systèmes capables de traduire automatiquement d'une langue à une autre Bar-Hillel (1964). En 1966, «*ELIZA*» a été développé par «*Joseph Weizenbaum*» dans le but de simuler une conversation à l'aide de techniques rudimentaires de traitement du langage naturel permettant de reformuler les phrases en posant des questions Weizenbaum (1966). «*Cordell Green*», une figure marquante de l'informatique est particulièrement connue pour ses travaux sur l'utilisation de la logique des prédicats pour la langue. Il fut l'un des premiers chercheurs à utiliser cette logique pour formaliser la langue à des fins de traduction automatique. Il a explicité la représentation des relations sémantiques entre les mots et les phrases à l'aide de la logique des prédicats. Il a également développé des systèmes capables d'interpréter des commandes en langue naturelle grâce à un raisonnement logique Green (1969). C'est en 1969 que la théorie de la dépendance conceptuelle a vu le jour, grâce aux travaux du psychologue et cogniticien «*Schank*». Inspirée par les travaux du linguiste américain «*Sydney Lamb*», cette théorie a été l'un des premiers modèles de compréhension de la langue. L'objectif de cette théorie était de séparer les mots d'une phrase de son sens. En d'autres termes, la théorie stipule que deux phrases sémantiquement équivalentes doivent avoir la même représentation du sens Lamb (1966); Schank (1969). Les chercheurs logico-symbolistes ne se sont toutefois pas arrêtés à la théorie de la dépendance conceptuelle. D'autres modèles orientés, cadre et graphe ont été proposés. Parmi ceux-ci, les modèles de cadres «*frames*» ont été introduits par «*Marvin Minsky*» dans sa communication «*A framework for representing knowledge*» de 1974. Ces modèles furent parmi les premières tentatives de formalisation de la description du savoir dans des dispositifs intelligents. Un cadre est une structure de données utilisée pour représenter des conditions ou des scènes dans un processus d'information. C'est aussi une organisation hiérarchique décrivant un concept ou un événement grâce à des attributs. Ces éléments, appelés, «*slots*», peuvent contenir une valeur, un code ou des liaisons vers d'autres cadres, exprimant ainsi le temps, le lieu, les participants, les objets impliqués, voire les actions elles-mêmes. Dans le traitement automatique du langage naturel, la connais-

sance de l'environnement d'un mot ou d'une expression est essentielle pour comprendre toute sa signification Minsky (1974). En 1976, les «*graphes conceptuels*» ont été proposés par «*John Sowa*». Ils'agissait de l'un des premiers formalismes de représentation de la connaissance. Initialement fondés sur la logique du premier ordre, ces graphes utilisaient une forme diagrammatique comme formalisme de représentation. Ce modèle se distingue des approches strictement syntaxiques et logiques par un cadre plus intuitif et flexible. Les travaux de «*John Sowa*» ont influencé de nombreux domaines, notamment linguistique computationnelle, l'intelligence artificielle, la représentation des connaissances et la modélisation des réseaux sémantiques Sowa (1976). Quatre ans plus tard, une extension des réseaux sémantiques a été initiée par deux universités néerlandaises «*Groningen et Twente*» qui ont été baptisée «*graphes de connaissances*». Ceux-ci se distinguent des réseaux sémantiques par la restriction des types de relations possibles entre les nœuds. L'objectif était de permettre des opérations algébriques sur les graphes James (1992). L'avènement du «*Web sémantique*» a influencé le domaine du «*TALN*» en introduisant un nouveau formalisme de représentation des données linguistiques. Le Web sémantique est apparu sur la scène grâce aux recherches menées par «*Tim Berners-Lee*» Berners-Lee *et al.* (2001). Il s'agit d'une description formelle des concepts et des relations. À l'aide de standards tels que «*RDF, RDFS et OWL*», le Web sémantique modélise des connaissances et établit des relations explicites entre concepts, enrichissant ainsi les ressources utilisées dans le «*TALN*», telles que les «*ontologies*» lexicales et les bases de données sémantiques. Le standard «*Resource Description Framework RDF*» est un modèle conceptuel de données basé sur des triplets, destiné à décrire les ressources Web et leurs métadonnées Klyne et Carroll (2004). Le standard «*Resource Description Framework Schema RDFS*» est une extension du «*RDF*» introduisant la notion de classe et de hiérarchie entre les classes Tous *et al.* (2011). «*Ontology Web Language OWL*» incarne un paradigme révolutionnaire dans l'ingénierie des connaissances qui est les «*ontologies*» Gruber (2009). Différentes définitions ont été proposées pour les ontologies. «*Robert Neches*» et ses collègues estiment qu'une ontologie est une définition des termes et des relations de base constituant le vocabulaire d'un domaine, ainsi que des règles permettant d'étendre ce vocabulaire Neches *et al.* (1991). «*Tom Gruber*», quant à lui, définit une ontologie comme une description explicite de concepts et de relations, destinée à un agent ou à une communauté d'agents Gruber (1993). Pour «*Sowa*», l'intérêt des ontologies réside dans l'étude des catégories d'objets qui existent ou peuvent exister dans un certain domaine Sowa (1995). Le Web sémantique a ouvert la porte à une contextualisation riche et à une interopérabilité fluide avec des données issues de plusieurs domaines. Plusieurs outils de traitement automatique du langage naturel ont émergé grâce au Web sémantique, notamment «*WordNet*», une ontologie lexicale contenant une base structurée de synonymes, d'antonymes, d'hyponymes et de relations sémantiques. Elle est utilisée dans des tâches telles que la désambiguïsation

lexicale et l'extraction d'informations Miller (1995). «*OntoNotes*», une ontologie combinant les annotations sémantiques, syntaxiques et discursives. Elle est utilisée pour le marquage sémantique et l'amélioration des systèmes TALN multilingues Hovy *et al.* (2006). «*FrameNet*», une ontologie basée sur des cadres sémantiques, utile pour l'encodage du sens des phrases Fillmore *et al.* (2006). «*DBpedia*», une ontologie multilingue pour l'extraction et la structuration des connaissances à partir de «*Wikipédia*» Auer *et al.* (2007). En outre, l'interopérabilité des ressources sémantiques améliore les performances des systèmes multilingues et rend possible le développement d'agents conversationnels plus intelligents et contextuellement pertinents, révolutionnant ainsi l'application du TALN dans de nombreux secteurs.

1.1.2.2 Approches statistiques en traitement automatique du langage naturel

Les approches statistiques en traitement automatique du langage naturel ont évolué avec le temps, intégrant des modèles probabilistes, des méthodes de réduction de la dimensionnalité, ainsi que des techniques d'apprentissage non supervisé afin de mieux comprendre les relations entre les mots et les documents. Dans cette section, nous allons explorer quelques modèles statistiques qui ont influencé l'état de l'art.

1. Modèles vectoriels catégoriques : les premiers modèles de représentation des mots étaient des modèles vectoriels catégoriques. Le «*one-hot encoding*» fut l'une des premières représentations catégoriques de mots. Il repose sur une représentation vectorielle binaire dans laquelle l'indice correspondant au mot, prend la valeur 1, tandis que toutes les autres cases contiennent des zéros. Le one-hot encoding a ensuite été étendu à une représentation fondée sur les sacs de mots «*bag-of-words*». Cette méthode consiste à représenter un document ou une phrase par une matrice indiquant le nombre d'occurrences de chaque mot Harris (1954); Salton (1971). Les modèles de représentation catégorique ont ensuite évolué pour donner naissance aux modèles à base de pondérations.

2. Term Frequency-Inverse Document Frequency : Contrairement aux modèles basés sur le poids, ils ne se limitent pas au nombre d'occurrences d'un terme. Ils prennent en compte la fréquence d'apparition d'un mot par rapport à la taille du document, à l'image des modèles «*Term Frequency TF*» et «*Term Frequency-Inverse Document Frequency (TF-IDF)*». Le TF-IDF a été introduit en 1970 comme une amélioration de la représentation des textes de la méthode «*bag-of-words BoW*». Il permet de pondérer l'importance des mots dans un document en tenant compte de leur fréquence et de leur rareté dans le corpus, ce qui améliore la précision dans des tâches comme la recherche d'information Jones (1972). La fréquence du terme (TF) mesure la fréquence d'apparition d'un mot dans un document donné. C'est un indicateur de l'importance

d'un mot dans un document spécifique. L'inverse de la fréquence du document (IDF) est une mesure qui réduit l'importance des mots fréquents dans tout le corpus, car ils sont généralement peu informatifs. L'idée est qu'un mot courant dans tous les documents n'apporte pas d'information. spécifique Jones (1972). Le score TF-IDF pour un mot dans un document est le produit de TF et d'IDF :

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{1.1}$$

Où : TF (Term Frequency) :

— TF(t, d) représente la fréquence d'apparition du terme t dans le document d.

$$\text{TF}(t, d) = \frac{\text{Nombre de fois que le terme } t \text{ apparaît dans le document } d}{\text{Nombre total de termes dans le document } d}$$

IDF (Inverse Document Frequency) :

— IDF(t) mesure l'importance du terme t dans l'ensemble du corpus.

$$\text{IDF}(t) = \log\left(\frac{\text{Nombre total de documents dans le corpus}}{\text{Nombre de documents contenant le terme } t}\right)$$

TF-IDF :

— TF-IDF(t, d) combine les deux mesures pour donner une mesure de l'importance du terme t dans le document d.

Bien entendu, les modèles à base de pondérations sont construits sur le modèle des sacs de mots « *bag-of-words* ». Par conséquent, ils partagent la même limitation : l'incapacité à capturer l'ordre des mots dans un document. Néanmoins, ils offrent de bonnes performances au niveau lexical Salton *et al.* (1975).

3. L'analyse sémantique latente : « *La sémantique latente* » est une autre forme de modèle vectoriel permettant d'interpréter les relations entre un terme et un document dans «*un espace conceptuel latent*». Elle s'appuie sur la décomposition en valeurs singulières «*(Singular Value Decomposition, SVD)*» pour factoriser la matrice de co-occurrence. L'analyse sémantique latente «*Latent Semantic Analysis LSA*» est une technique de réduction de dimensionnalité utilisée pour extraire des relations sémantiques entre les mots et les documents. Cette approche a été introduite en 1990 pour pallier les limitations du modèle «*BoW*» et du modèle «*TF-IDF*», car elle permet de réduire la dimensionnalité à l'aide de la décomposition SVD. Théoriquement, elle repose sur l'hypothèse distributionnelle, selon laquelle les termes apparaissant dans des contextes similaires ont des significations similaires. L'Analyse sémantique latente consiste à décomposer la matrice terme-document, notée A, en trois matrices à l'aide de la décomposition en valeurs singulières. Cette factorisation permet de révéler des relations latentes entre les termes et les documents, réduisant ainsi la dimensionnalité de l'espace vectoriel tout en préservant la structure sémantique sous-jacente.

Le modèle LSA peut être formulé comme suit :

$$A \approx USV^T \tag{1.2}$$

Où :

- A est la matrice terme-document de taille $m \times n$, avec m étant le nombre de termes et n le nombre de documents.
- U est une matrice de taille $m \times k$ qui représente les relations entre les termes et les dimensions latentes, où k est le nombre de dimensions latentes choisies.
- S est une matrice diagonale de taille $k \times k$ qui contient les valeurs singulières indiquant l'importance de chaque dimension latente.
- V^T est la matrice transposée de V, de taille $k \times n$, représentant les relations entre les documents et les dimensions latentes.

Cette décomposition permet de « *réduire la dimensionnalité* » tout en capturant les structures sémantiques cachées dans les données textuelles.

4. Latent Dirichlet Allocation : «*Latent Dirichlet Allocation LDA*» est un autre modèle appartenant à la catégorie des approches statistiques. Il a été proposé en 2003 par « *Blei, Ng et Jordan* ». Il s'agit d'un modèle génératif de texte fondé sur des distributions de probabilité. «*LDA*» est utilisé pour modéliser des sujets dans un corpus de textes. Cette approche probabiliste permet d'identifier des thèmes latents au sein d'un corpus, en supposant que chaque document est une combinaison de plusieurs sujets, et que chaque sujet est caractérisé par une distribution de probabilité sur les mots. Blei *et al.* (2003).

Notations LDA :

- M : Nombre total de documents.
- N_d : Nombre de mots dans le document d.
- K : Nombre de sujets (topics).
- β_k : Distribution des mots pour le sujet k, un vecteur de taille V (vocabulaire).
- θ_d : Distribution des sujets pour le document d, un vecteur de taille K.
- $z_{d,n}$: Sujet assigné au n-ième mot dans le document d.
- $w_{d,n}$: n-ième mot dans le document d.
- α : Paramètre hyperparamètre de la distribution de Dirichlet sur θ_d.
- η : Paramètre hyperparamètre de la distribution de Dirichlet sur β_k.

Le processus génératif de LDA peut être formalisé comme suit :

1. Pour chaque sujet k, tirer une distribution des mots :

$$\beta_k \sim \text{Dirichlet}(\eta)$$

2. Pour chaque document d :
   — Tirer une distribution des sujets :

$$\theta_d \sim \text{Dirichlet}(\alpha)$$

   — Pour chaque mot $w_{d,n}$ dans le document d :
      (a) Tirer un sujet :

$$z_{d,n} \sim \text{Categorical}(\theta_d)$$

      (b) Tirer un mot :

$$w_{d,n} \sim \text{Categorical}(\beta_{z_{d,n}})$$

Probabilité jointe complète :

$$P(w, z, \theta, \beta \mid \alpha, \eta) = \prod_{k=1}^{K} P(\beta_k \mid \eta) \prod_{d=1}^{M} P(\theta_d \mid \alpha) \prod_{n=1}^{N_d} P(z_{d,n} \mid \theta_d) P(w_{d,n} \mid \beta_{z_{d,n}}) \tag{1.3}$$

L'algorithme fonctionne par «*inférence bayésienne*», en affectant les mots dans les documents à des sujets de manière itérative jusqu'à ce que le modèle converge. Chaque mot dans un document est affecté à un sujet latent selon une probabilité conditionnelle. Le modèle apprend ensuite les distributions de sujets et de mots à partir des données textuelles Blei *et al.* (2003).

5. Vecteurs Globaux pour la Représentation des Mots : «*GloVe*» est une technique statistique inspirée du modèle de l'analyse sémantique latente. L'auteur de GloVe critique le contexte local de «*Word2Vec*» et propose un contexte global considérant le mot par rapport au reste du document. Le processus d'apprentissage du modèle GloVe est basé sur l'hypothèse que le logarithme de la probabilité de co-occurrence d'un mot dans la matrice de co-occurrence est égal au produit scalaire de la ligne et de la colonne qui lui correspondent. $\log(X_{ij}) = w_i \cdot c_j + b_i + b_j$ où X_{ij} est la fréquence de co-occurrence, w_i et c_j sont les vecteurs des mots centraux et contextuels, et b_i, b_j sont des biais scalaires. «*La factorisation de matrice*» utilisée par GloVe consiste en plusieurs itérations ayant pour objectif la décomposition en valeurs singulières la matrice de co-occurrence afin d'émerger les deux matrices correspondant aux mots centraux et ceux du contexte Pennington *et al.* (2014). Contrairement à une décomposition classique en valeurs singulières , GloVe utilise un modèle d'apprentissage supervisé où une fonction de coût spécifique est minimisée :

$$J = \sum_{i,j} f(X_{ij}) \left(\log(X_{ij}) - w_i \cdot c_j - b_i - b_j\right)^2 \tag{1.4}$$

La fonction $f(X_{ij})$ pèse les erreurs en fonction de la fréquence de co-occurrence, favorisant un ajustement précis pour les paires fréquentes tout en limitant l'impact des valeurs rares.

- X_{ij} : Fréquence de co-occurrence entre les mots i et j.
- $\log(X_{ij})$: Transformation logarithmique de la fréquence pour éviter des variations extrêmes.
- $w_i \cdot c_j$: Produit scalaire entre le vecteur de mot w_i et le vecteur de contexte c_j, qui représente la similarité sémantique entre les deux mots.
- b_i, b_j : Biais associés aux mots et aux contextes pour stabiliser l'apprentissage.
- $f(X_{ij})$: Fonction d'importance qui pondère les paires en fonction de leur fréquence pour éviter que des paires trop rares ou trop fréquentes dominent l'optimisation.

Les approches statistiques exploitent les propriétés quantitatives du texte pour représenter les mots et les documents. La méthode TF-IDF pondère les mots en fonction de leur fréquence et de leur rareté dans un corpus, mais ne permet pas de prendre en compte la sémantique. La méthode LDA modélise les documents comme des distributions de sujets et permet d'extraire des thématiques sous-jacentes grâce à une approche probabiliste. Le LSA, basé sur la décomposition en valeurs singulières, réduit la dimensionnalité des représentations textuelles et révèle des relations latentes entre les mots, mais avec une perte d'interprétabilité. Enfin, GloVe utilise les cooccurrences globales des mots pour générer des vecteurs qui capturent les relations sémantiques et analogiques, offrant ainsi une représentation plus riche de la langue. Bien que ces méthodes aient marqué une avancée significative, elles restent limitées par rapport aux modèles neuronaux modernes, qui utilisent des architectures plus complexes pour mieux modéliser le sens et le contexte de la langue.

1.1.2.3 Approches neuronales en traitement du langage naturel

L'intelligence artificielle et ses sous-disciplines connaissent un engouement croissant, notamment depuis les résultats impressionnants obtenus par les réseaux de neurones profonds. Le traitement automatique du langage naturel, qui est une sous-discipline de l'intelligence artificielle, a ainsi franchi une étape importante grâce à ces modèles. L'intérêt croissant pour le traitement automatique du langage naturel s'explique par le large éventail d'applications dans l'industrie, notamment dans les domaines de la traduction automatique, de la classification de textes, du résumé automatique, de la reconnaissance des entités nommées, de l'analyse des sentiments et des agents conversationnels. Dans cette section, nous allons explorer quelques modèles neuronaux qui ont influencé l'état de l'art dans ce domaine.

1. Word2Vec : La première catégorie utilise principalement des réseaux de neurones. Elle a été étudiée entre 2001 et 2003 par Sun (2000). Cependant, les résultats obtenus n'étaient pas aussi prometteurs que ceux des autres techniques de modélisation de la langue Bengio *et al.* (2003). Des années plus tard, l'auteur de Mikolov *et al.* (2010) explora l'application des «*réseaux neuronaux récurrents RNR*» à la modélisation du langage. Il finit par proposer la fameuse technique «*Word2Vec*» avec ses deux modèles «*CBoW*» et «*SkipGram*» Mikolov *et al.* (2013). Word2Vec utilise une technique de fenêtrage local pour déterminer le mot central et les mots voisins qui représentent le contexte de ce mot. Il utilise un simple perceptron pour apprendre à représenter les mots dans deux matrices formées à partir des poids du perceptron. La première contient les mots centraux et la deuxième, les mots du contexte. Le modèle SkipGram prédit les mots du contexte à partir du mot central, tandis que CBoW prédit le mot central à partir des mots du contexte. En revanche, la deuxième catégorie repose sur des techniques statistiques pour l'apprentissage de la représentation des mots. Mikolov *et al.* (2013).

Modèle Skip-gram Le modèle Skip-gram maximise la probabilité de prédire le contexte C autour d'un mot cible w_t. La fonction objective est donnée par :

$$\max \prod_{t=1}^{T} \prod_{c \in C_t} P(w_c | w_t) \tag{1.5}$$

où la probabilité conditionnelle est définie comme :

$$P(w_c | w_t) = \frac{\exp(v'_{w_c} \cdot v_{w_t})}{\sum_{w \in V} \exp(v'_w \cdot v_{w_t})} \tag{1.6}$$

où : - v_{w_t} est le vecteur d'entrée du mot cible w_t, - v'_{w_c} est le vecteur de sortie du mot de contexte w_c, - V est le vocabulaire.

Modèle CBOW (Continuous Bag of Words)

Le modèle CBOW prédit un mot cible w_t à partir des mots de contexte C environnants. La fonction objective est :

$$\max \prod_{t=1}^{T} P(w_t | C_t) \tag{1.7}$$

où la probabilité conditionnelle est donnée par :

$$P(w_t|C_t) = \frac{\exp(v'_{w_t} \cdot \frac{1}{|C_t|} \sum_{c \in C_t} v_{w_c})}{\sum_{w \in V} \exp(v'_w \cdot \frac{1}{|C_t|} \sum_{c \in C_t} v_{w_c})} \tag{1.8}$$

où : - v_{w_c} sont les vecteurs d'entrée des mots de contexte, - v'_{w_t} est le vecteur de sortie du mot cible.

2. Doc2Vec : «*Doc2Vec*» est une extension de Word2Vec. Il s'agit d'un modèle qui permet de représenter les documents sous forme de vecteurs. Alors que Word2Vec est utilisé pour encoder le sens des mots, Doc2Vec permet d'encoder l'ensemble d'un document. Il génère un vecteur unique pour chaque document, entraîné parallèlement aux vecteurs des mots. Ce vecteur capture les informations contextuelles qui permettent de distinguer ce document des autres. Doc2Vec fonctionne sur deux architectures principales. La première est appelée « mémoire distribuée ». Cette architecture est similaire à Word2Vec, mais elle ajoute un vecteur d'identification du document qui permet de prédire les mots en fonction du contexte. La deuxième architecture, dite «*sac de mots distribué*», est toutefois plus proche du modèle «*Skip-Gram*» de «*Word2Vec*». Dans cette architecture, le modèle apprend à prédire les mots à partir du vecteur du document, sans tenir compte de l'ordre des mots. Le et Mikolov (2014). Le modèle de la mémoire distribuée prédit chaque mot w_t en fonction des mots précédents et du vecteur de document $\mathbf{d}_i$:

$$P(w_t|w_{t-1}, w_{t-2}, \ldots, w_1, \mathbf{d}_i) = \frac{\exp(\mathbf{v}_{w_t}^\top (\sum_{j=1}^{t-1} \mathbf{v}_{w_j} + \mathbf{d}_i))}{\sum_{w \in V} \exp(\mathbf{v}_w^\top (\sum_{j=1}^{t-1} \mathbf{v}_{w_j} + \mathbf{d}_i))} \tag{1.9}$$

où :

- $\mathbf{v}_w$ est le vecteur de représentation du mot w.
- $\mathbf{d}_i$ est le vecteur de représentation du document d_i.
- V est le vocabulaire total.

Le modèle de sac de mots distribué prédit chaque mot indépendamment de son ordre à partir du vecteur de document $\mathbf{d}_i$:

$$P(w_t|\mathbf{d}_i) = \frac{\exp(\mathbf{v}_{w_t}^\top \mathbf{d}_i)}{\sum_{w \in V} \exp(\mathbf{v}_w^\top \mathbf{d}_i)} \tag{1.10}$$

où :

- $\mathbf{v}_{w_t}$ est le vecteur de représentation du mot w_t.

— $\mathbf{d}_i$ est le vecteur de représentation du document d_i.

— V est le vocabulaire total.

3. Context2Vec : Melamud *et al.* (2016) propose «*Context2Vec*», un autre modèle neuronal, fondé sur une critique du contexte local proposé par Word2Vec. Contrairement au modèle Word2Vec, qui génère une seule représentation fixe pour chaque mot, Context2Vec génère une représentation différente pour chaque occurrence d'un mot, selon le contexte. L'auteur utilise les «*Long Short-Term Memory LSTM*» comme moyen de mieux capter le contexte d'un mot. Context2Vec a pour objectif le plongement des mots cibles ainsi que de leur contexte situé dans la phrase dans le même espace réduit, afin d'expliciter les dépendances entre les mots cibles et le contexte. Il utilise un encodage bidirectionnel du contexte grâce aux LSTM. Il vise à obtenir une représentation contextuelle de chaque mot, basée sur sa proximité avec d'autres mots dans une fenêtre contextuelle Melamud *et al.* (2016). Soit w_t le mot cible et $w_{c1}, w_{c2}, \ldots, w_{cn}$ les mots contextuels dans la fenêtre de taille n. La représentation d'un mot dans le contexte est donnée par $\mathbf{e}_t = \text{Embeddings}(w_t)$ et $\mathbf{e}_c = \text{Embeddings}(w_{c1}, w_{c2}, \ldots, w_{cn})$. L'encodeur est un réseau récurrent RNN qui prend la séquence de mots contextuels et génère une représentation du mot cible w_t : $\mathbf{h}_t = RNN(\mathbf{e}_c)$. Après avoir obtenu la représentation $\mathbf{h}_t$, la probabilité du mot cible est prédite à l'aide d'une couche fully connected suivie d'une fonction softmax : $\mathbf{y}_t = \text{Softmax}(W\mathbf{h}_t + b)$

La fonction de perte est ensuite calculée en utilisant l'entropie croisée entre la prédiction $\mathbf{y}_t$ et le mot réel w_t :

$$\mathcal{L} = -\sum_{i=1}^{V} \mathbf{y}_t[i] \log(\hat{\mathbf{y}}_t[i]) \tag{1.11}$$

4. ELMo : En 2018, le modèle «*ELMo*» est apparu à la suite des travaux de Peters *et al.* (2018). Les auteurs essayaient, à travers ELMo, de prendre en compte l'aspect syntaxique et sémantique du mot dans l'encodage contextuel. ELMo se compose de deux couches neuronales constituées de plusieurs unités LSTM. Les unités LSTM sont interconnectées dans les deux sens, permettant ainsi un encodage bidirectionnel du contexte. Chaque couche neuronale génère ce qu'on appelle des vecteurs intermédiaires, qui seront ajoutés aux vecteurs originaux des mots pour donner la représentation finale. ELMo a permis de franchir un pas important en matière de représentation contextuelle des mots en utilisant des «*LSTM bidirectionnels*», mais il a été rapidement éclipsé par des modèles basés sur le transformeur, comme «*BERT*», qui offrent des performances bien supérieures grâce à leur capacité à gérer des dépendances longues et à leur parallélisation plus efficace. L'architecture d'ELMo repose sur un modèle de langue bidirectionnel avec une architecture de réseau neuronal récurrent. Soit $w_1, w_2, \ldots, w_T$ la séquence de mots d'entrée de longueur T, où chaque w_i repré-

sente un mot dans le vocabulaire. L'idée principale d'ELMo est de produire une représentation contextuelle de chaque mot en fonction de ses voisins dans la phrase. Cette représentation est obtenue par l'utilisation de deux modèles de langue : un modèle de langue direct, de gauche à droite, et un modèle de langue inverse, de droite à gauche Peters *et al.* (2018).

Soit $\vec{h}_t^{\text{fw}}$ et $\vec{h}_t^{\text{bw}}$ les sorties des couches cachées du modèle de langue direct et inverse respectivement pour le mot w_t.

a) Modèle de langue direct :

$$\vec{h}_t^{\text{fw}} = \text{RNN}_{\text{fw}}(w_1, w_2, \ldots, w_t) \tag{1.12}$$

b) Modèle de langue inverse :

$$\vec{h}_t^{\text{bw}} = \text{RNN}_{\text{bw}}(w_T, w_{T-1}, \ldots, w_{t+1}) \tag{1.13}$$

Les sorties des deux RNNs sont combinées pour donner une représentation contextuelle bidirectionnelle du mot w_t : $\vec{h}_t = [\vec{h}_t^{\text{fw}}; \vec{h}_t^{\text{bw}}]$ où $[\cdot;\cdot]$ représente la concaténation des vecteurs.

c) Représentation finale de chaque mot : La représentation contextuelle d'ELMo pour un mot donné est une combinaison linéaire des représentations à différentes couches du réseau RNN. Soit $\vec{h}_t^l$ la sortie de la l-ème couche de l'RNN, la représentation finale du mot w_t dans ELMo est donnée par $\mathbf{E}_t = \sum_{l=1}^{L} \gamma_l \cdot \vec{h}_t^l$ où L est le nombre total de couches dans le réseau, et γ_l est un poids apprenant associé à la l-ème couche, qui est optimisé lors de l'entraînement.

4. BERT : La publication d'ELMo a coïncidé avec l'arrivée d'un autre modèle, «*BERT*», qui ne se contente pas d'assurer l'encodage bidirectionnel, mais aussi l'encodage positionnel des mots. BERT est conçu sur la base des transformeurs. Il repose sur un modèle fondé sur des mécanismes d'attention permettant de traiter efficacement les dépendances à longue portée dans un texte. Par rapport aux modèles antérieurs tels que Word2Vec ou GloVe, le principal apport de BERT réside dans sa capacité à générer des représentations contextuelles riches. Il est disponible en deux versions. La version de base contient 12 encodeurs ayant chacun 12 têtes d'attention . La deuxième version, dite large, contient 24 encodeurs ayant chacun 16 têtes d'attention. BERT s'appuie sur deux stratégies d'apprentissage principales : la première consiste en la modélisation masquée du langage, évoquant la prédiction d'un mot au sein d'une phrase, et la deuxième est la prédiction de la phrase suivante. Plusieurs autres modèles ont émergé à la suite de BERT, notamment SciBERT Beltagy et Cohn (2020), BioELECTRA pour la normalisation conceptuelle médicale Kanaka-

rajan *et al.* (2021), ainsi que «*BNE*» et «*BioSyn*» pour l'encodage nominatif biomédical Sung *et al.* (2020). L'application de BERT à d'autres langues a donné naissance à «*AraBERT*» pour l'arabe Antoun *et al.* (2020), «*CamemBERT*» pour le français Martin *et al.* (2019). BERT a connu des améliorations avec «*RoBERTa*» Liu *et al.* (2019), conçu selon la stratégie de masquage, avec un apprentissage basé sur des «*mini-batches*» et un taux d'apprentissage plus élevé. «*DistilBERT*» introduit la technique de distillation, et «*XLNet*» la prédiction aléatoire avec la technique de permutation, permettant un meilleur encodage bidirectionnel Devlin *et al.* (2019); Sanh *et al.* (2019); Yang *et al.* (2019). L'architecture de BERT repose sur les transformeurs, qui sont composés de couches d'attention et de réseaux «*feedforward*». L'attention est calculée comme suit. :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.14}$$

où :

- $Q \in \mathbb{R}^{n \times d_k}$ est la matrice des requêtes ,
- $K \in \mathbb{R}^{n \times d_k}$ est la matrice des clés ,
- $V \in \mathbb{R}^{n \times d_v}$ est la matrice des valeurs,
- d_k est la dimension des clés.

Les matrices Q, K, et V sont obtenues à partir de l'entrée E par multiplication avec des matrices de poids apprises. Dans BERT, l'attention est effectuée sur plusieurs têtes, chacune ayant ses propres matrices de poids. L'attention multi-tête est calculée comme suit :

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)W^O \tag{1.15}$$

où chaque tête head_i est calculée comme :

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{1.16}$$

et W^O est une matrice de poids de sortie.

L'encodeur BERT ajoute également des informations de position dans l'entrée en utilisant des encodages

positionnels, car l'architecture transformeur n'a pas de structure séquentielle implicite. «*L'encodage positionnel*» $P \in \mathbb{R}^{n \times d}$ est ajouté à l'entrée E : $E_{\text{final}} = E + P$. L'objectif de BERT est de prédire les mots masqués à partir des représentations contextuelles apprises. Le modèle est entraîné avec une fonction de perte de type entropie croisée :

$$\mathcal{L} = -\sum_{i \in \mathcal{M}} \log P(x_i | x_{\text{masked}}) \tag{1.17}$$

où $\mathcal{M}$ est l'ensemble des positions des mots masqués, et $P(x_i|x_{\text{masked}})$ est la probabilité prédite du mot x_i à la position i, donnée par le modèle.

5. GPT : «*Generative Pre-trained transformer*» est un modèle de langue «*auto-régressif*» basé sur l'architecture transformeur. Contrairement à BERT, qui est bidirectionnel, GPT génère des séquences de mots en se basant uniquement sur les tokens précédents, en prédisant le mot suivant dans la séquence. L'entrée de «*GPT*» est une séquence de tokens, où chaque token est transformé en un vecteur d'embedding. Pour une séquence de tokens $x_1, x_2, \ldots, x_n$, l'entrée du modèle devient une matrice d'embeddings $E \in \mathbb{R}^{n \times d}$, où d est la dimension de l'embedding et n est la longueur de la séquence. L'entrée est également complétée par des encodages positionnels pour introduire l'ordre des tokens dans la séquence. Ces encodages positionnels sont ajoutés à la matrice d'embedding : $E_{\text{final}} = E + P$, où $P \in \mathbb{R}^{n \times d}$ représente les encodages positionnels, et E_{final} est la matrice d'entrée enrichie par les informations de position. L'architecture de GPT repose sur les transformeurs. Contrairement à BERT, GPT utilise uniquement des couches d'auto-attention causales auto-régressives Radford *et al.* (2018, 2019); Brown *et al.* (2020b). La principale différence réside dans la manière dont l'attention est calculée, en utilisant uniquement les tokens précédents pour prédire le token suivant. Le mécanisme d'attention causale dans GPT est donné par :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.18}$$

Cependant, pour garantir que l'attention soit causale (autrement dit, qu'un token ne puisse pas "voir" les tokens futurs), on applique un masquage sur les positions futures dans la matrice des scores d'attention. Cela se fait en remplaçant les valeurs au-dessus de la diagonale de la matrice QK^T par $-\infty$ (ou un très grand nombre négatif) afin d'empêcher l'attention vers les tokens futurs :

$$\text{Masked\_Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \tag{1.19}$$

où M est une matrice de masquage telle que $M_{ij} = -\infty$ si $j > i$, et $M_{ij} = 0$ sinon.

Comme dans BERT, GPT utilise l'attention multi-tête pour permettre au modèle de se concentrer sur différentes parties de la séquence d'entrée. L'attention multi-tête est calculée comme suit :

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right) W^O \tag{1.20}$$

où chaque tête head_i est calculée comme :

$$\text{head}_i = \text{Masked\_Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{1.21}$$

et W^O est une matrice de poids de sortie. Après l'attention, la sortie passe par un réseau feedforward à deux couches. Le réseau feedforward est défini comme suit : $FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$. où $W_1 \in \mathbb{R}^{d \times d_{\text{ff}}}$ et $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d}$ sont des matrices de poids, et b_1 et b_2 sont des biais. L'objectif de GPT est de prédire les tokens suivants dans une séquence donnée, ce qui en fait un modèle auto-régressif. Pour une séquence d'entrée de n tokens, le modèle prédit le token suivant x_{i+1} à partir des i tokens précédents $x_1, \ldots, x_i$.

La fonction de perte de GPT est la log-vraisemblance négative des tokens prédits :

$$\mathcal{L} = -\sum_{i=1}^{n-1} \log P(x_{i+1}|x_1, \ldots, x_i) \tag{1.22}$$

où $P(x_{i+1}|x_1, \ldots, x_i)$ est la probabilité prédite du token suivant x_{i+1}, conditionnée sur les tokens précédents.

1.1.3 La triade peircienne comme grille de lecture des modèles proposés

Les modèles proposés dans les chapitres 3 et 5 peuvent être considérés comme une machine de «*sémiosis*» au sens de «*Charles Sanders Peirce*». La structure syntaxique, représentée par la matrice d'adjacence des dépendances syntaxique, joue le rôle de «*représentamen*». Cette structure agit comme un signe porteur de signification, dont l'«*objet*» est la relation «*sémantico-syntaxique*» réelle entre les mots. Le mécanisme d'attention permettant d'introduire la matrice d'adjacence sous la forme d' «*attention mask*» comme c'est le cas dans le chapitre 3 ou encore le mécanisme d'attention contraint par des «*multiplicateurs de Lagrange*» dans les chapitres 5 et 6, produit l'interprétant. Ce dernier se présente dans les représentations vectorielles des mots qui intègrent et respectent cette structure syntaxique. Ainsi, «*SCABERT*», mentionné dans le chapitre 5, incarne un processus de sémiosis computationnelle, où la signification linguistique n'est pas simplement extraite passivement, mais activement construite et validée par un dialogue dynamique entre la forme syntaxique et le sens représenté sous forme vectorielle. Ce rapprochement entre une architecture profonde de réseaux de neurones et la théorie peircienne du signe montre comment la «*linguistique computationnelle*» peut dépasser la simple corrélation statistique pour intégrer une dimension structurante et interprétative fondamentale de la signification. Dans lingBERT présenté dans le chapitre 4, le représentamen émerge à travers des phrases dans lesquelles les mots masqués sont sélectionnés selon leurs relations syntaxiques, contrairement au masquage aléatoire utilisé dans «*BERT*». Ces phrases masquées, structurées autour de dépendances syntaxiques qui constituent des unités signifiantes dont l'objet est la relation sémantico-syntaxique réelle entre les mots. Le processus de prédiction des mots masqués, en tenant compte de ces dépendances, oriente l'apprentissage du modèle vers une compréhension plus structurée et motivée du langage. Ainsi, l'«*interprétant*» prend la forme de vecteurs contextualisés, qui encodent non seulement le sens lexical des mots, mais aussi leur rôle syntaxique et leur contribution au sens global de la phrase. Ce mécanisme d'attention syntaxiquement guidée favorise une sémiosis plus fine, alignée avec la structure profonde de la langue . Dans le chapitre 6 qui aborde le modèle VLG-BERT, dans ce dernier, le représentamen peut être vu comme la forme complexe et multimodale des représentations internes du modèle. On y trouve notamment, La matrice d'adjacence syntaxique qui encode les dépendances entre mots, qui structure explicitement la forme linguistique, ainsi que les vecteurs d'embedding initialisés à partir des représentations latentes visuelles extraites du modèle «*Vision Transformer ViT*». Ce dernier donne une dimension perceptuelle à la forme linguistique. Ce représentamen est donc une fusion entre une forme linguistique syntaxique et une forme perceptuelle, visuelle, incarnant la forme du signe qui sera interprétée. Dans «*VLG-BERT*» l'objet est double et multimodal. D'une part, il s'agit des relations syntaxiques

réelles entre les mots dans la phrase ; ces relations sont codifiées dans la matrice d'adjacence syntaxique qui sert de vérité terrain. D'autre part, il s'agit des entités et concepts réels auxquels les mots concrets renvoient dans le monde. Ils capturés par les représentations visuelles latentes préapprises sur «*ImageNet*». Ainsi, l'objet n'est pas seulement un concept abstrait ou un mot isolé, mais une relation linguistique située dans un contexte perceptif réel, riche et hiérarchisé grâce à «*WordNet*». L'interprétant dans VLG-BERT correspond quant à lui à la représentation contextuelle et sémantique intégrée que le modèle construit et affine à travers son mécanisme d'attention. L'optimisation via les multiplicateurs de Lagrange garantit que les représentations produites respectent la structure syntaxique, et donc que l'interprétation respecte la forme linguistique. L'intégration des embeddings «*embeddings*» permet une interprétation ancrée dans le sens réel des mots, favorisant ainsi une compréhension multimodale plus riche. L'interprétant est ainsi la compréhension «*incarnée*» que le modèle construit une interprétation dynamique et contextuelle du signe qui intègre à la fois la structure linguistique et la signification perceptive.

1.2 Opacité des grands modèle de langue

Avec l'essor des modèles de type «*boîte noire*» comme les réseaux de neurones profonds, la question de leur interprétabilité «*comprendre leur fonctionnement*» et de leur explicabilité «*obtenir des justifications claires de leurs décisions*» est devenue centrale dans les domaines du traitement du langage naturel et de la vision par ordinateur. Cette tendance est motivée par la nécessité croissante de mettre en place des systèmes d'intelligence artificielle transparents et fiables en particulier dans les contextes sensibles tels que «*santé*, *justice*, *finance*» Lipton (2017). Pour gagner la confiance des utilisateurs de ces secteurs sensibles, ces derniers doivent comprendre les décisions prises par ces modèles. En cas d'erreur, comment la détecter et la corriger si le modèle est opaque ? En outre, les modèles profonds peuvent amplifier les biais «*racistes*, *sexistes*, *etc.*», ce qui rend une auditabilité nécessaire. Dans le contexte des grands modèles de langue, les représentations vectorielles de la langue sont difficiles à interpréter humainement. Les mécanismes d'attention indiquent ce que le modèle regarde, mais pas toujours le «*pourquoi* Arrieta *et al.* (2019); Rudin *et al.* (2021). Les grands modèles génératifs inventent des réponses qui paraissent plausibles mais qui sont en réalité fausses, et ce sans justification interne. Ce phénomène est appelé «*hallucination*» Ji *et al.* (2023); Rudin *et al.* (2021). «*Gradient-weighted Class Activation Mapping Grad-CAM*» est un outil qui utilise une technique d'explicabilité des modèles d'apprentissage profond, notamment pour les réseaux de neurones convolutifs (CNN) Selvaraju *et al.* (2019). Il met en évidence les zones d'une image influençant la décision, mais sans logique sémantique claire ! Un changement infime, appelé «*adversarial attack*» peut tromper le modèle et

révéler sa fragilité interprétative Chakraborty *et al.* (2018). Il existe un paradoxe remarquable entre la performance et l'interprétabilité. Par exemple, les modèles les plus performants comme les «*transformeurs*» sont souvent les moins interprétables. Dans ce qui suit, nous allons expliciter les notions d'interprétabilité et d'explicabilité.

1.2.1 L'interprétabilité

L'«*interprétabilité*» consiste en l'habilité à appréhender le fonctionnement d'un modèle neuronal profond. L'interprétabilité est une problématique qui s'adresse à la communauté scientifique de ce domaine. Elle est une condition nécessaire, mais non suffisante pour l'explicabilité. Un modèle interprétable n'est pas nécessairement explicable pour un non-expert Doshi-Velez et Kim (2017); Gilpin *et al.* (2019); Arrieta *et al.* (2019). L'interprétabilité passe généralement par la simplification de ses composants ou en les reliant à des structures connues telles que les «*arbres de décision*, *règles*». L'interprétabilité s'attaque à plusieurs problèmes, notamment la complexité architecturale «*attention multi-têtes*, *milliards de paramètres*», l'opacité des représentations internes «*embeddings*, *mécanismes d'attention*», les biais et la robustesse «*impact des données d'entraînement*» Lipton (2017); Rudin *et al.* (2021). Plusieurs courants influents dans le domaine de l'interprétabilité sont à noter. Le premier soutient la méthode «*post-hoc*» qui consiste en une analyse à posteriori permettant la visualisation des attentions, l'analyse de neurones et l'utilisation de classificateurs de sondage «*probing classifiers*», etc Jain et Wallace (2019); Dalvi *et al.* (2018); Tenney *et al.* (2019a). Un deuxième courant repose sur l'identification des sous-réseaux responsables de comportements, ce que l'on appelle l'approche par intervention. Cette technique consiste à supprimer des composants, tels que les «*têtes d'attention*, *couches*» pour mesurer leur impact comme dans les «*ablation studies*» ou bien à modifier les entrées pour observer les changements de sortie, à la manière des «*contrefactuels*» Kovaleva *et al.* (2019); Wexler *et al.* (2019). Une autre communauté s'attaque à l'interprétabilité via des méthodes symboliques et formelles, qui consistent à approximer le modèle par des règles logiques comme la «*distillation en arbres de décision*», ou à identifier des sous-réseaux responsables de comportements Hinton *et al.* (2015); Olah *et al.* (2020).

1.2.2 L'explicabilité

L'«*explicabilité*» vise à exposer les raisons d'une décision. Elle s'adresse aux utilisateurs et consommateurs finaux. L'explicabilité produit des justifications pour les décisions précises. Il s'agit donc d'une ingénierie des explications. Un modèle explicable peut utiliser des approximations sans qu'il soit nécessaire d'en com-

prendre le fonctionnement en profondeur Doshi-Velez et Kim (2017); Guidotti *et al.* (2018). Afin d'améliorer l'explicabilité, les chercheurs déploient des techniques d'explication à posteriori qui se basent sur la visualisation. Le modèle «*Local interpretable model-agnostic explanations LIME*» est un modèle de visualisation qui approxime localement un modèle complexe par un modèle simple, dont le but est de générer une explication simple et rapide d'une prédiction individuelle Ribeiro *et al.* (2016a). L'explication se fait avec un affichage des caractéristiques «*features*» les plus influentes pour une prédiction donnée. SHAP est un autre modèle de visualisation, basé sur la théorie des jeux «*valeur de Shapley*». SHAP attribue la contribution de chaque caractéristique à la prédiction. Il permet ainsi une analyse rigoureuse des contributions des caractéristiques Lundberg et Lee (2017). L'explicabilité peut toutefois se faire par le biais d'une analyse des caractéristiques pour déterminer l'importance des mots ou de segments dans un texte par exemple. Dans certains contextes, l'explicabilité doit prendre la forme d'un modèle génératif de texte pour expliquer ses sorties. Par exemple, je classe ce courriel comme indésirable, car il contient les mots «*offre exclusive*» et «*urgence*» Arras *et al.* (2017); Hendricks *et al.* (2016). L'explication permet de résoudre des problèmes comme l'hallucination pour les grands modèles de langue, qui représente l'un des principaux défis de l'explicabilité pour ce type de modèle. Elle peut se manifester lorsque l'explication d'un faux énoncé générée par un grand modèle de langue Mehrabi *et al.* (2022); Hao *et al.* (2025). L'autre défi, tout aussi ardu, est celui des biais dont souffrent les modèles neuronaux profonds. Afin de mettre en pratique l'importance de ce genre de problème, nous prenons à titre d'exemple un modèle de recrutement. Ce dernier peut désavantager un groupe sans aucune raison claire, ce qui est très problématique Raghavan *et al.* (2020).

1.2.3 Positionnement des travaux

Cette section vise à positionner les travaux de recherche de la présente thèse au regard des notions d'interprétabilité et d'explicabilité, deux concepts clés dans l'évaluation et la confiance envers les modèles d'apprentissage profond. Elle tente de répondre à la question suivante : «*En quoi les modèles lingBERT, SCABERT et VLG-BERT ouvrent-ils de nouvelles perspectives en matière d'interprétabilité et d'explicabilité pour les chercheurs spécialisés dans ces domaines ?*».

Le «*masquage aléatoire*» des mots utilisé dans «*BERT*», ainsi que l'initialisation aléatoire des embeddings en début d'entraînement, augmentent l'opacité des grands modèles de langue, en particulier BERT. Les trois variantes proposées dans le cadre de cette thèse, «*lingBERT, SCABERT et VLG-BERT*», contribuent à la réduction de cette opacité en s'attaquant spécifiquement à la stratégie de masquage aléatoire de BERT.

Dans un premier temps, LingBERT prépare le corpus d'entraînement en introduisant un masquage hybride combinant une sélection aléatoire et une sélection basée sur les dépendances linguistiques entre les mots. SCABERT, quant à lui, supervise l'entraînement de BERT en intégrant une matrice de dépendances linguistiques au niveau de la couche de prédiction. Cette matrice agit comme une étant une vérité de terrain «*ground truth*». Elle guide le modèle vers une meilleure prise en compte de la structure syntaxique. Ainsi SCABERT fournit une interprétation directe du comportement du modèle, quels mots influencent lesquels et selon quelle relation syntaxique. En plus, le recours à une formulation par multiplicateurs de Lagrange, SCABERT offre une lecture mathématique précise des contraintes, ce qui augmente la traçabilité du processus d'apprentissage. Avec SCABERT, l'explication d'une prédiction peut désormais s'appuyer sur la structure syntaxique reconstruite par le modèle, pour dire que ce mot a influencé cette décision, car il est relié syntaxiquement au mot prédictif. Cela facilite la génération d'explications formelles ou même verbales, pour dire que cette phrase est classée ainsi, car le sujet est modifié par un adjectif négatif fort. Enfin, VLG-BERT, une extension de SCABERT, va encore plus loin en s'attaquant non seulement au masquage aléatoire, mais aussi à l'initialisation aléatoire des «*embeddings*», une pratique courante dans la plupart des grands modèles actuels. VLG-BERT propose une initialisation sémantiquement fondée. Elle est basée sur des représentations latentes de mots concrets. Ces représentations sont dérivées de modèles neuronaux de vision et agissent comme des «*labels*» conceptuels associés aux mots, apportant ainsi une dimension sémantique plus ancrée dans la réalité perceptive. Dans VLG-BERT, les représentations sémantiques des mots sont désormais ancrées dans des concepts visuels partagés avec les êtres humains, ce qui permettrait d'interpréter une activation neuronale comme étant liée à un concept visuel concret. Cette méthode offre une traçabilité sémantique latente interprétable, rendant les représentations internes plus humaines et plus auditables. L'ancrage visuel permet également d'identifier des corrélations perceptives aux activations du modèle, ce qui facilite la compréhension par des non-experts. VLG-BERT ouvre donc la voie à une interprétabilité cognitive et multimodale. En introduisant des concepts visuels dans le processus d'apprentissage, VLG-BERT permet d'analyser visuellement une image pour déterminer si un mot est associé à ce type d'image, ce qui explique pourquoi il a été interprété ainsi. Cela permet de créer une base pour des explications génératives multimodales, dans lesquelles une décision textuelle peut être reliée à un univers sensoriel.

1.3 Problématiques

1.3.1 Volet cognitif

Malgré les résultats impressionnants des grands modèles de langue, ceux-ci restent très opaques du point de vue des sciences cognitives. Contrairement à un être humain qui construit du sens en ancrant la langue dans son expérience du monde. Ces modèles fonctionnent sans perception, sans interaction physique et sans véritable intentionnalité. Ils ne possèdent pas la capacité de compréhension du monde proprement dite. Ils manipulent des corrélations symboliques et statistiques sur des milliards de mots à l'aide de modèles neuronaux, sans avoir un réel accès au sens des mots. En ce sens, la théorie de Peirce justifie le recours à des approches syntaxiques et multimodales pour rapprocher les modèles du fonctionnement sémiosique humain, et ainsi améliorer l'encodage du sens, l'explicabilité et la cohérence sémantique des «*LLMs*». Dans cette optique, il devient pertinent de revenir aux fondements de la sémiotique, en particulier à la théorie du signe développée par «*Charles Sanders Peirce*». Pour «*Peirce*», la signification n'est pas une relation figée, mais un processus dynamique et évolutif dans lequel chaque signe appelle une interprétation. Cette dernière à son tour, devient un signe pour un nouvel interprétant, dans une chaîne potentiellement infinie d'interprétations. Cette conception est particulièrement féconde pour repenser les limites des LLMs. En effet, intégrer l'approche «*percienne*» à la modélisation du langage implique de reconnaître que le sens ne peut être réduit à un simple encodage symbolique ou vectoriel. Il s'agit plutôt d'un processus interprétatif, médié par des structures perceptuelles et contextuelles. Cela ouvre la voie à une modélisation plus riche du langage naturel, dans laquelle le rôle de l'interprétant peut être opérationnalisé par des mécanismes attentionnels guidés, des modules de traitement multimodal, ou encore des structures hiérarchiques interprétatives qui imitent la dynamique du sens. Ce cadre théorique permet ainsi de jeter les bases d'un pont rigoureux entre la cognition et l'architecture neuronale profonde. Les modèles proposés visent à simuler un processus interprétatif. «*La question principale est donc la suivante : comment intégrer des connaissances linguistiques, telles que la syntaxe, ainsi que des connaissances du monde physique, dans les grands modèles de langue afin de les rapprocher du modèle humain ?*» «*Comment peut-on ouvrir une voie sur l'interprétabilité des grands modèles de langue ?*» «*En quoi les modèles proposés, inspirés de la triade sémiotique de* «Peirce», *permettent-ils une modélisation plus fidèle du sens ?*»

1.3.2 Volet informatique

Les grands modèles de langue ont démontré d'excellentes performances dans plusieurs tâches de traitement automatique du langage naturel. Cependant, ces modèles, qui reposent principalement sur l'attention et les «*embeddings*» de mots, restent limités par leur dépendance exclusive au texte, sans prise en compte des connaissances linguistiques et visuelles. Le «*masquage aléatoire*» des mots utilisés dans «*BERT*», ainsi que l'initialisation aléatoire des «*embeddings*» en début d'entraînement, ne permettent pas une meilleure intelligibilité de BERT. L'une des principales difficultés réside dans le manque d'explicabilité et d'interprétabilité des mécanismes d'attention, souvent considérés comme des boîtes noires. «*Comment injecter efficacement des connaissances linguistiques et visuelles dans ces modèles afin d'améliorer leurs performances sur les différentes tâches de traitement automatique du langage naturel ?*» De plus, «*comment concevoir des stratégies d'injection de connaissances permettant non seulement d'améliorer les performances du modèle, mais aussi de le rendre explicable et interprétable ?*»

1.4 Hypothèses

1.4.1 Volet cognitif

- Hypothèse 1 : L'injection de connaissances linguistiques et visuelles dans les grands modèles de langue permet de mieux structurer le représentamen. Cette structuration rapproche les modèles des mécanismes de signification chez l'être humain, conformément à la théorie sémiotique de Peirce et aux théories cognitives qui soulignent l'importance de la multimodalité dans la constitution du sens.
- Hypothèse 2 : Adopter une perspective peircéenne permet de formaliser l'interaction entre la langue, la perception et la cognition en tant que processus de sémiose. Cette approche offre un cadre théorique solide pour justifier l'intégration des connaissances linguistiques et visuelles dans les grands modèles de langage, en mettant en avant la nature triadique du sens. Elle oriente ainsi le développement des modèles vers une intelligence artificielle plus sémiotique et plus cognitive.

1.4.2 Volet informatique

- Hypothèse 1 : L'injection de connaissances syntaxiques sous forme de dépendances linguistiques dans les mécanismes d'attention des modèles de langage améliore le processus d'encodage du sens des mots, ce qui se traduit par de meilleures performances dans les tâches de traitement automa-

tique du langage naturel.

— Hypothèse 2 : L'intégration de représentations visuelles latentes dans l'apprentissage des représentations numériques des mots permet d'améliorer le processus d'encodage du sens. Cette approche permet d'ancrer la sémantique physique du monde dans la langue, ce qui renforce la robustesse et la capacité de généralisation des modèles de langue.

1.5 Plan de la thèse

Cette thèse est structurée en six chapitres. Le premier chapitre introduit la question du sens en général, et plus particulièrement celle du sens des mots, sous l'angle des sciences cognitives et de l'informatique. La présente thèse s'inscrit dans le cadre d'un doctorat en informatique cognitive, ce qui rend nécessaire une présentation des différentes théories abordant la notion de sens, avant d'aborder les principales approches de son encodage en informatique.

Dans ce même chapitre, nous proposons une première lecture selon une perspective peircienne, qui constitue le cadre théorique et cognitif de la contribution informatique. Cette approche permet également d'aborder les notions d'interprétabilité et d'explicabilité, et de situer les travaux de recherche menés dans cette thèse. Le chapitre se conclut par la formulation des problématiques et des hypothèses qui guideront les deux volets de la recherche, à savoir le volet informatique et le volet cognitif.

Le deuxième chapitre présente l'état de l'art des mécanismes d'attention appliqués au traitement du texte et de l'image. Il s'intéresse aux modèles visant à associer des connaissances linguistiques et visuelles aux mécanismes d'attention, en soulignant l'absence quasi systématique de liens explicites avec la sémiotique, ces approches (sémiotiques) restant rares dans la littérature.

Les quatre chapitres suivants détaillent les contributions et les approches proposées pour répondre aux problématiques soulevées. Enfin, la thèse se termine par une conclusion générale qui synthétise les contributions apportées et ouvre des perspectives pour de futurs travaux.

CHAPITRE 2

L'ÉTAT DE L'ART

2.1 Les mécanismes d'attention en langage naturel

Les mécanismes d'attention constituent une avancée déterminante dans le domaine de l'apprentissage profond, particulièrement dans la modélisation des dépendances contextuelles au sein de séquences complexes. Introduit initialement par Bahdanau *et al.* (2016) pour le traitement du langage naturel dans la traduction automatique, le mécanisme d'attention permet aux modèles d'accorder un poids différencié aux différentes parties de l'entrée, en fonction de leur pertinence contextuelle. Cette capacité a transformé l'architecture des modèles séquentiels, permettant une gestion fine de l'information sans dépendre exclusivement de la position ou de l'ordre dans la séquence. Avec l'introduction du transformeur Vaswani *et al.* (2017), l'attention a pris un rôle central dans le traitement des séquences, en permettant un accès parallèle à toutes les positions d'entrée, ce qui a considérablement amélioré l'efficacité et la performance des modèles. Rapidement, le mécanisme a été adapté à d'autres modalités comme la vision Dosovitskiy *et al.* (2021a) ou l'audio, et généralisé au traitement multimodal Rahman *et al.* (2020), rendant possible l'injection croisée de connaissances linguistiques et visuelles au sein d'un espace de représentation partagé Rahman *et al.* (2020); Dai *et al.* (2023); Ramesh *et al.* (2022). Plusieurs variantes de l'attention ont ainsi émergé. L'attention globale prend en compte les relations entre tous les éléments d'une séquence, tandis que l'attention locale restreint le calcul à une fenêtre contextuelle, comme dans Luong (2015). L'attention causal est utilisée notamment dans les modèles de génération pour empêcher l'accès à des tokens futurs Brown *et al.* (2020a). L'attention auto-régressive, quant à elle, permet une prédiction séquentielle ordonnée Katharopoulos *et al.* (2020). L'attention hiérarchique, introduite par Yang *et al.* (2016a), structure l'information à différents niveaux mots, phrases et documents, capturant ainsi des dépendances à long terme. Enfin, l'attention croisée, largement utilisée dans les modèles multimodaux comme ViLBERT Lu *et al.* (2019) ou UNITER Chen *et al.* (2020), permet d'aligner et de fusionner efficacement des représentations issues de différentes modalités. Dans ce qui suit, nous présentons les différents types de mécanismes d'attention employés dans l'état de l'art, en mettant en lumière leur rôle dans l'intégration et l'encodage profond du sens à partir de données linguistiques, visuelles, ou hybrides.

2.2 L'attention locale

L'attention locale consiste en la restriction de la fenêtre d'attention de chaque mot à un nombre limité de mots voisins au lieu d'analyser toute la séquence. Cela signifie que chaque mot ne peut prêter attention qu'à un sous-ensemble de mots dans son voisinage. Chaque mot ne regarde que les mots dans une fenêtre définie autour de lui. Les poids d'attention sont calculés uniquement pour ce sous-ensemble de mots. Cela

réduit le coût de calcul et de mémoire tout en maintenant un contexte pertinent. L'attention est calculée sur une fenêtre restreinte, réduisant ainsi la complexité à $O(n \times k)$, où k représente la taille de la fenêtre. Toutefois, si la fenêtre est trop petite, les mots éloignés ne peuvent pas interagir entre eux. Par conséquent, le choix de la fenêtre peut limiter la compréhension du modèle. Longformer est un modèle transformeur qui utilise une attention locale pour traiter de longues séquences de texte, permettant de réduire la complexité de l'attention à $O(n \times k)$, où k est la taille de la fenêtre locale Sheynin *et al.* (2021). Longformer applique une attention locale glissante à chaque token, et pour certains tokens, il applique aussi une attention globale pour capturer les relations à longue portée. Dans Longformer chaque token n'interagit qu'avec ses voisins dans une fenêtre glissante. Un nombre restreint de tokens, tels que $[CLS]$ ou $[SEP]$,bénéficie d'une attention globale. Longformer est conçu pour le traitement de documents très longs, comme des articles scientifiques ou des livres. Le résumé de texte et les questions-réponses sur des documents longs Beltagy *et al.* (2020). Linformer est un autre modèle qui réduit la complexité de l'attention dans les transformateurs en utilisant une attention locale linéaire, ce qui permet de traiter efficacement des séquences longues sans faire exploser le coût de calcul. Ce modèle repose sur l'idée que l'attention dans les transformeurs peut être bien approximée par une attention low-rank (faible rang), permettant de remplacer la matrice d'attention dense par une approximation. Ce modèle applique une réduction de rang linéaire pour l'attention, ce qui réduit la mémoire et les calculs. L'attention est locale et approximative, mais suffisamment précise pour traiter de longues séquences Wang *et al.* (2020). Performer est un autre modèle de transformateur qui utilise un mécanisme d'attention approximative pour rendre l'attention plus rapide et moins gourmande en mémoire. L'idée principale est de reformuler l'attention classique en une version qui utilise des techniques de kernels pour approximer les produits scalaires, ce qui permet de traiter de longues séquences avec moins de ressources. Il utilise une approximation des noyaux pour calculer l'attention. Performer réduit la complexité de l'attention à $O(n \log n)$, ce qui est beaucoup plus efficace pour les longues séquences Choromanski *et al.* (2020). Le Sparse transformeur est une autre variante du transformeur standard qui applique une attention locale en réduisant le nombre de calculs d'attention. Ce modèle permet de travailler sur de longues séquences tout en utilisant une attention éparse, où seuls certains tokens interagissent. Dans ce modèle l'attention est éparse et localisée à des fenêtres fixes et les calculs sont donc beaucoup plus rapides et moins gourmands en mémoire que ceux de l'attention dense classique Child *et al.* (2019). Les modèles qui utilisent l'attention locale sont particulièrement adaptés pour traiter des séquences longues, où l'attention globale traditionnelle serait trop coûteuse en termes de calcul et de mémoire. Ces modèles permettent d'optimiser l'efficacité tout en maintenant de bonnes performances sur des tâches de traitement de texte, en particulier lorsque les séquences sont longues ou quand les ressources de calcul sont limitées.

2.3 L'attention globale

L'attention globale est un mécanisme où chaque token (mot) dans une séquence peut prêter attention à tous les autres tokens de la séquence. Cela signifie que chaque mot peut établir une relation avec tous les autres mots, peu importe leur position. L'attention global-local dans le cadre de l'architecture ETC « Extended transformeur Construction » est un mécanisme conçu pour traiter efficacement les longues séquences en divisant l'attention en parties restreintes et non restreintes. ETC utilise deux types d'entrées distincts. La première est l'entrée globale, un petit ensemble de tokens auxiliaires, qui ont une attention illimitée et servent de passerelle d'information. Le deuxième type est l'entrée longue qui contient la séquence principale des tokens, à l'instar d'un Transformeur classique, mais avec une attention restreinte. Ce mécanisme est particulièrement utile pour le traitement de documents longs, car il permet de gérer des séquences de grande taille sans exploser la mémoire et le temps de calcul. Il est utilisé pour réduire la complexité en limitant l'attention long-to-long, ETC diminue la complexité quadratique $\mathcal{O}(N^2)$ d'un Transformeur classique. Les tokens globaux, qui ont une attention illimitée, permettent aux tokens longs de communiquer indirectement entre eux. Les tokens globaux agissent comme des résumés contextuels, améliorant le traitement de textes longs Vaswani *et al.* (2017); Ainslie *et al.* (2020).

- Global-to-Global : Les tokens globaux peuvent s'attendre entre eux sans restriction.
- Global-to-Long : Les tokens globaux peuvent voir tous les tokens de la séquence longue.
- Long-to-Global : Les tokens de la séquence longue peuvent voir tous les tokens globaux.
- Long-to-Long : Les tokens de la séquence longue n'ont qu'une attention restreinte à un rayon fixe.

Le mécanisme standard de l'attention globale est défini par les étapes suivantes :

a) Calcul des scores d'attention : pour chaque mot dans la séquence de sortie, on calcule un score d'attention qui évalue l'importance des mots dans la séquence d'entrée. Ce score est généralement calculé comme le produit scalaire entre un vecteur de requête (Q) associé à l'élément de sortie et un vecteur de clé (K) associé à l'élément d'entrée.

$$\text{score} = \frac{QK^T}{\sqrt{d_k}} \tag{2.1}$$

où Q est le vecteur de requête, K est le vecteur de clé et d_k est la dimension des vecteurs.

b) Application d'une fonction de pondération : Les scores obtenus sont ensuite passés par une fonction de softmax pour les transformeurs en probabilités. Ces probabilités servent à pondérer les valeurs (V) associées aux mots de l'entrée.

$$\text{attention weights} = \text{softmax}(\text{score}) \tag{2.2}$$

b) Calcul de la sortie : les valeurs pondérées (V) sont ensuite agrégées pour produire la sortie de l'attention pour un mot donné. Cela permet à chaque élément de sortie de "voir" toute la séquence d'entrée, en se concentrant sur les mots jugés les plus pertinents.

$$\text{output} = \sum(\text{attention weights}) \times V \tag{2.3}$$

Il existe plusieurs modèles s'appuyant sur l'attention globale, qui permettent de capturer les dépendances à long terme dans le texte. RoBERTa, par exemple, utilise la même architecture de modèle que BERT. Cependant, il bénéficie d'un volume de données d'entraînement plus important et d'un temps d'entraînement plus long, ce qui lui permet d'améliorer ses performances. L'une des innovations majeures de RoBERTa est l'utilisation du masquage dynamique. Ce modèle est souvent qualifié de modèle à attention globale en raison de l'utilisation de l'attention multi-tête dans l'architecture des transformeurs, permettant à chaque mot de la séquence d'accorder une importance à tous les autres, quelle que soit leur position Liu *et al.* (2019). D'autre part, ALBERT introduit plusieurs améliorations en matière d'efficacité par rapport à BERT, en se concentrant sur la réduction de la taille du modèle tout en maintenant des performances comparables. Comme RoBERTa, ALBERT utilise le masquage dynamique, dans lequel les jetons à masquer sont sélectionnés aléatoirement et peuvent varier à chaque epoch. ALBERT est également un modèle à attention globale pour les mêmes raisons que RoBERTa. Bien qu'il partage l'architecture de base de BERT, certains ajustements le rendent plus léger, tout en conservant la capacité d'attention globale Lan *et al.* (2020). De même, DeBERTa adopte des stratégies de masquage dynamique similaires à celles de RoBERTa, où le schéma de masquage change au cours de l'entraînement. Cela permet d'éviter une sur-adaptation à des positions masquées spécifiques. Comme BERT, RoBERTa et ALBERT, DeBERTa repose sur une architecture de type transformeur avec un mécanisme d'auto-attention. Chaque mot peut ainsi prêter attention à tous les autres mots de la séquence, indépendamment de leur position relative. L'attention est donc dite globale, car elle ne se limite pas à une fenêtre locale autour de chaque mot, mais considère toute la séquence He *et al.* (2021). En contraste, SpanBERT applique un masquage basé sur les spans (plages contiguës de tokens). Au lieu de masquer des jetons individuellement, SpanBERT masque des séquences voisines de jetons. Autrement dit, il sélectionne des plages entières de texte à masquer plutôt que des jetons aléatoires. SpanBERT maintient

un mécanisme d'attention globale, mais propose une approche différente de masquage, centrée sur des groupes de mots. Cela permet de mieux capturer les relations sémantiques entre spans, tout en exploitant l'attention globale. TinyBERT est une version allégée de BERT, obtenue par distillation des connaissances. Il est conçu pour réduire la taille et le coût computationnel du modèle, tout en conservant des performances acceptables par rapport à BERT-base ou BERT-large. Il conserve le mécanisme d'attention globale, mais avec un nombre réduit de couches et de paramètres, ce qui le rend plus rapide et plus léger Jiao *et al.* (2020). Sur le même principe, DistilBERT est une version simplifiée de BERT, entraînée par distillation des connaissances, avec 50% de paramètres en moins et une vitesse d'exécution accrue de 60%. DistilBERT apprend à partir des sorties d'un modèle plus grand (le professeur), ce qui permet à ce modèle plus petit (l'élève) de conserver des performances proches tout en étant plus efficace. Il conserve lui aussi le mécanisme d'attention globale, bien qu'avec une architecture plus compacte. L'attention globale constitue un mécanisme fondamental qui permet aux modèles de type transformeur de pondérer et d'intégrer de manière flexible les informations provenant de différentes parties d'une séquence. Toutefois, les défis liés à sa complexité computationnelle ont motivé des recherches visant à rendre ce mécanisme plus efficace pour le traitement de séquences longues. Par exemple, dans BERT, la modélisation du langage masqué repose sur un masquage statique appliqué durant le pré-entraînement. En revanche, RoBERTa applique un nouveau masque à chaque epoch, ce qui signifie que les jetons à masquer sont choisis différemment à chaque passage d'un exemple d'apprentissage Vaswani *et al.* (2017); Ainslie *et al.* (2020).

2.4 L'attention auto-régressive

L'attention auto-régressive est un mécanisme très puissant. Il opte pour la causalité dans son paradigme de génération de mots. Les mots générés dépendent des mots déjà générés, formant ainsi une chaîne causale. L'attention auto-régressive est fondée sur l'idée principale des transformateurs proposée par Vaswani et al. L'un des modèles les plus performants est celui introduit par OpenAi appelé GPT. Le modèle mathématique de l'attention auto-régressive dans GPT repose sur le mécanisme de l'attention à partir de vecteurs de requêtes, de clés et de valeurs, avec un masquage pour préserver la causalité. Pour chaque position t dans la séquence, nous calculons les scores d'attention en utilisant les produits scalaires entre la requête Q_t et les clés K_j. Ces scores sont ensuite normalisés par la racine de la dimension des clés d_k.

$$\text{score}(t, j) = \frac{Q_t \cdot K_j}{\sqrt{d_k}} \tag{2.4}$$

Cependant, pour garantir la causalité, un masque de causalité est appliqué sur les scores d'attention afin de faire en sorte que la position t ne puisse pas regarder les positions futures $j > t$. Ainsi, les scores pour $j > t$ sont modifiés de la manière suivante :

$$\text{score}(t, j) = \begin{cases} \frac{Q_t \cdot K_j}{\sqrt{d_k}} & \text{si} \quad j \leq t \\ -\infty & \text{si} \quad j > t \end{cases} \tag{2.5}$$

Le masque causal empêche chaque position t de regarder les positions futures. Ainsi, lorsque la fonction softmax est appliquée pour normaliser les scores d'attention. Les termes pour $j > t$ deviennent nuls, c'est-à-dire que les poids associés à ces éléments sont égaux à zéro.

La normalisation des scores par la fonction softmax donne les poids d'attention $\alpha_{t,j}$:

$$\alpha_{t,j} = \frac{\exp(\text{score}(t, j))}{\sum_{k=1}^{t} \exp(\text{score}(t, k))} \quad \text{pour} \quad j \leq t \tag{2.6}$$

et pour $j > t$, $\alpha_{t,j} = 0$.

Une fois les poids d'attention $\alpha_{t,j}$ calculés, la sortie y_t à chaque position t est une somme pondérée des valeurs V_j :

$$y_t = \sum_{j=1}^{t} \alpha_{t,j} V_j \tag{2.7}$$

Ce mécanisme d'attention causale permet à chaque élément de la séquence de générer une sortie qui dépend uniquement des éléments précédents, excluant toute information future. Ainsi, le modèle respecte strictement l'ordre temporel de la génération, condition nécessaire pour des tâches comme la modélisation de langue ou la génération de texte. Le modèle GPT est constitué de plusieurs couches empilées de transformeurs auto-régressifs. À chaque couche, l'attention est recalculée à partir des représentations générées par la couche précédente, en maintenant le masquage causal. Cette architecture en profondeur permet au modèle de capturer des relations complexes entre tokens, même à longue distance dans la séquence, en combinant progressivement des informations contextuelles plus riches à chaque niveau.

$$\text{score}(t, j) = \frac{Q_t \cdot K_j}{\sqrt{d_k}} \quad \text{pour} \quad j \leq t \tag{2.8}$$

et

$$\text{score}(t, j) = -\infty \quad \text{pour} \quad j > t \tag{2.9}$$

Le masque causal garantit que les éléments futurs ne sont pas accessibles à la position actuelle.

Application de softmax :

$$\alpha_{t,j} = \frac{\exp(\text{score}(t, j))}{\sum_{k=1}^{t} \exp(\text{score}(t, k))} \quad \text{pour} \quad j \leq t \tag{2.10}$$

Le Calcul de la sortie :

$$y_t = \sum_{j=1}^{t} \alpha_{t,j} V_j \tag{2.11}$$

Parmi les modèles fondés sur l'attention auto-régressive, CTRL a Conditional Transformer Language Model for Controllable Generation. Il se distingue par sa capacité à générer du texte de manière contrôlée. En plus du texte d'entrée appelé prompt ce modèle prend également en compte un ou plusieurs codes de contrôle, qui orientent le style, le registre ou le domaine du contenu généré. Ces codes sont des tokens spéciaux insérés au début du prompt, permettant de conditionner la génération en fonction d'un ensemble prédéfini de catégories stylistiques ou thématiques. Le modèle adapte ainsi sa production tout en maintenant la cohérence avec les contraintes imposées par ces codes de contrôle Keskar *et al.* (2019). Un autre modèle important est Transformer-XL, qui s'inscrit dans la lignée des modèles GPT, mais introduit un mécanisme de mémoire récurrente pour améliorer la modélisation des dépendances à long terme. Le texte est divisé en segments successifs de tokens, chacun pouvant couvrir plusieurs phrases ou documents. Contrairement aux modèles standards qui traitent chaque segment indépendamment, Transformer-XL concatène les états cachés du segment précédent à ceux du segment courant afin de calculer les scores d'attention. Ce mécanisme permet au modèle de capturer une continuité contextuelle entre les segments, prolongeant ainsi efficacement la fenêtre d'attention au-delà de la longueur fixe des séquences Dai *et al.* (2019).

2.5 L'attention hiérarchique

L'attention hiérarchique est un concept qui complète l'attention globale en traitant les relations au sein de niveaux d'abstraction hiérarchiques. Elle est particulièrement utile pour les tâches où les données ont une

structure imbriquée, comme les textes longs ou les images. Dans le texte, l'attention peut être appliquée d'abord aux mots, puis aux phrases, enfin aux paragraphes. Chaque niveau d'attention peut capter des relations plus larges entre les éléments à un niveau supérieur, tout en se concentrant sur des détails à des niveaux inférieurs. Tandis que l'attention globale permet de capturer des relations exhaustives en prenant en compte tous les éléments de la séquence, ce qui peut être coûteux en termes de ressources computationnelles pour de grandes séquences l'attention hiérarchique introduit une manière plus structurée et computationnelle efficace de traiter ces relations. L'attention hiérarchique, en revanche, permet une gestion plus structurée des relations en divisant la séquence en plusieurs niveaux. L'attention globale a une complexité $O(n^2)$, alors que l'attention hiérarchique peut diviser cette complexité en fonction du niveau, réduisant ainsi la charge computationnelle tout en préservant une modélisation des relations à long terme.

Le modèle mathématique général peut être formulé comme suit :

a) Représentation des éléments : soit $D = \{x_1, x_2, \ldots, x_N\}$ un document ou une séquence, où x_i est la représentation vectorielle de l'élément i (qui peut être un mot, une phrase ou un paragraphe), et N est le nombre total d'éléments dans la séquence. L'attention locale modélise les relations entre des éléments proches (par exemple, entre mots au sein d'une même phrase). Cette relation locale est modélisée via une fonction d'attention qui agrège les informations des éléments voisins dans un voisinage local $\mathcal{N}_i$ autour de x_i. L'attention locale pour chaque élément x_i peut être définie comme suit :

$$h_i^{(l)} = \text{Attention}_{\text{local}}(x_i, \{x_j\}_{j \in \mathcal{N}_i}) \tag{2.12}$$

où $\mathcal{N}_i$ représente l'ensemble des voisins de x_i dans un graphe local, souvent construit selon des relations de proximité ou de similarité. Cette étape capture les interactions locales entre les éléments.

b) Attentions globales : l'attention globale permet de capturer les relations entre des éléments distants dans le document, par exemple entre des phrases ou des paragraphes. Cette attention est calculée sur les représentations locales $\{h_i^{(l)}\}$ obtenues dans l'étape précédente, afin de pondérer l'importance relative de chaque élément dans le document. L'attention globale est définie comme suit :

$$h_i^{(g)} = \text{Attention}_{\text{global}}(h_i^{(l)}, \{h_j^{(l)}\}_{j \in \mathcal{N}_i}) \tag{2.13}$$

où $\mathcal{N}_i$ représente cette fois les voisins au niveau global (par exemple, les autres phrases ou paragraphes). Cette attention permet de capturer les relations à plus grande échelle entre les éléments dans le document.

c) Fusion des représentations locales et globales : les représentations locales et globales sont combinées pour former une représentation agrégée du document. Cette fusion est essentielle pour capturer la structure hiérarchique des relations. La fusion des attentions locales et globales pour chaque élément x_i peut être réalisée par une combinaison pondérée des deux types d'attention :

$$h_i = \lambda_1 h_i^{(l)} + \lambda_2 h_i^{(g)} \tag{2.14}$$

où λ_1 et λ_2 sont des poids appris qui déterminent l'importance relative des attentions locales et globales.

d) Décodage : la représentation agrégée h_i de chaque élément x_i peut être utilisée pour des tâches de traitement ultérieures, telles que la classification, la synthèse de texte, ou la génération de séquences. Un décodeur, typiquement un réseau de neurones feed-forward, peut être utilisé pour obtenir les résultats souhaités. Pour une tâche de classification ou de sélection, on peut calculer un score pour chaque élément :

$$\hat{y}_i = \text{Decoder}(h_i) \tag{2.15}$$

Le modèle d'attention hiérarchique offre une méthode puissante pour traiter des documents ou des séquences complexes en capturant des relations à plusieurs niveaux. En combinant des attentions locales et globales, ce modèle permet de mieux comprendre les dépendances dans des textes longs, tout en préservant l'efficacité computationnelle. L'introduction de plusieurs niveaux d'attention permet de gérer des interactions à différentes échelles, rendant ce modèle adapté à une large gamme de tâches en traitement de texte. Le modèle HAN (*Hierarchical Attention Networks*) applique une attention au niveau des mots dans une phrase, puis au niveau des phrases dans un document. L'objectif est de capter l'importance relative des mots et des phrases pour la tâche de classification ou d'analyse Yang *et al.* (2016b). Un autre modèle possédant un mécanisme d'attention hiérarchique est DocNADE (*Document Neural Autoregressive Distribution Estimator*). Il utilise une attention hiérarchique pour modéliser les relations entre les mots et les documents, permettant de capturer des structures complexes et de générer des représentations de documents plus efficaces Lauly *et al.* (2016). Zhao *et al.* (2018) propose HSA-RNN v (*Hierarchical Structure-Adaptive RNN for*

Video Summarization), une nouvelle approche adaptative au résumé vidéo intégrant la segmentation des plans et la synthèse vidéo dans un RNN hiérarchique adaptatif à la structure. Zhao *et al.* (2024) introduit *HAND* (*Hierarchical Attention Network for Multi-Scale Document*), une nouvelle architecture de bout en bout et sans segmentation pour la reconnaissance de texte et l'analyse de mise en page simultanée. Les principaux composants du modèle incluent un encodeur convolutionnel avancé intégrant des convolutions séparables en profondeur et des convolutions octavées pour une extraction robuste des caractéristiques, un cadre de traitement adaptatif multi-échelle qui s'ajuste dynamiquement à la complexité du document, ainsi qu'un décodeur d'attention hiérarchique avec des mécanismes d'attention sparse et augmentée par mémoire. Ces composants permettent au modèle de s'adapter efficacement aux pages allant d'une seule ligne à des pages à trois colonnes tout en maintenant une efficacité computationnelle Hamdan *et al.* (2024). Le *3-LHTN* (*Three-level Hierarchical Transformer Network*) est une approche innovante pour modéliser les dépendances à long terme dans les notes cliniques, dans le but de prédire des informations au niveau du patient. Il utilise une structure hiérarchique pour apprendre de manière progressive à différentes échelles. Le premier niveau utilise un modèle BERT pré-entraîné pour encoder les mots en phrases, avec la possibilité de le fine-tuner pour des tâches spécifiques. Les deuxième et troisième niveaux sont constitués de piles d'encodeurs basés sur des transformeurs, permettant d'agréger progressivement l'information du niveau de la phrase au niveau de la note, puis de la note au niveau du patient. Une des principales améliorations de ce modèle est la capacité à traiter des séquences beaucoup plus longues que les modèles BERT traditionnels, limités à 512 tokens. Ce modèle offre une solution robuste et évolutive pour prédire les résultats des patients à partir d'un grand nombre de notes cliniques, en exploitant la puissance des transformeurs dans une configuration hiérarchique Si et Roberts (2021).

2.6 L'attention croisée

L'attention croisée est un autre type de mécanisme d'attention utilisé dans les architectures de réseaux neuronaux profonds. Elle est notamment utilisée dans les modèles transformateurs multimodaux. Elle est appropriée pour les tâches nécessitant une interaction entre deux séquences distinctes. L'attention croisée permet de concentrer l'attention sur des données hétérogènes tout en les intégrant pour produire une réponse cohérente. Cette notion est particulièrement pertinente dans le contexte des modèles qui doivent souvent traiter des données multimodales. L'attention croisée suit le même principe que l'attention classique introduite par Luong (2015) dans les modèles séquence-à-séquence, puis généralisée dans le transformeur de Vaswani *et al.* (2017). Elle repose sur le calcul des scores d'attention entre les représentations

des deux séquences. l'attention croisée est utilisée pour intégrer les informations de l'encodeur dans les représentations du décodeur. Chaque élément du décodeur sélectionne dynamiquement les informations les plus pertinentes dans l'encodeur en utilisant les poids d'attention Gheini *et al.* (2021); Chen *et al.* (2021). L'attention croisée est définie entre une séquence source $\mathbf{X} \in \mathbb{R}^{n \times d}$ et une séquence cible $\mathbf{Y} \in \mathbb{R}^{m \times d}$, où n et m sont respectivement le nombre de tokens dans chaque séquence, et d est la dimension d'embedding. Les matrices de projection des requêtes, clés et valeurs sont définies comme :

$$\mathbf{Q} = \mathbf{Y}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{2.16}$$

où $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ sont les matrices de projection. Le score d'attention est calculé par :

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \tag{2.17}$$

où $\mathbf{A} \in \mathbb{R}^{m \times n}$ est la matrice des poids d'attention. La sortie de l'attention croisée est obtenue en appliquant $\mathbf{A}$ aux valeurs :

$$\mathbf{Z} = \mathbf{A}\mathbf{V} \tag{2.18}$$

où $\mathbf{Z} \in \mathbb{R}^{m \times d_k}$ représente les représentations contextuelles de la séquence cible. Pour le mécanisme multi-tête, chaque tête effectue une attention croisée indépendante avec une dimension réduite $d_h = \frac{d}{h}$:

$$\mathbf{Z}_i = \text{Attention}(\mathbf{Y}\mathbf{W}_{Q_i}, \mathbf{X}\mathbf{W}_{K_i}, \mathbf{X}\mathbf{W}_{V_i}) \tag{2.19}$$

où $\mathbf{W}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{W}_{V_i} \in \mathbb{R}^{d \times d_h}$.

Les sorties des h têtes sont concaténées et projetées dans l'espace original :

$$\mathbf{Z} = \text{Concat}(\mathbf{Z}_1, \ldots, \mathbf{Z}_h)\mathbf{W}_O \tag{2.20}$$

où $\mathbf{W}_O \in \mathbb{R}^{d \times d}$ est la matrice de projection finale.

L'attention croisée est un mécanisme puissant qui permet aux modèles d'apprendre des représentations complexes. Elle est utilisée dans plusieurs domaines tels que la traduction automatique, vision & langue et les systèmes de dialogue. DALL-E, «*Deep autoregressive language and latent embeddings* » développé par OpenAI, utilise l'attention croisée pour générer des images à partir de descriptions textuelles. Google propose T5, « Text to Text Transfer Transformeur ». Celui-ci traite toutes les tâches de traitement de la langue sous forme de texte à texte. T5 utilise l'attention croisée pour convertir des séquences d'entrée en

séquences de sortie correspondantes Raffel *et al.* (2023). CLIP, « Contrastive Language-Image Pretraining » par OpenAI, utilise l'attention croisée pour associer des descriptions textuelles à des images correspondantes, permettant un encodage multimodal efficace Radford *et al.* (2021).

2.7 Grands modèles de langue orientés syntaxe et vision

Dans ce contexte d'injection de connaissances linguistiques et visuelles, il convient de souligner que la présente thèse s'inscrit dans une contribution en informatique cognitive résolument cadrée, visant à modéliser le processus épistémologique décrit par Peirce à travers sa triade sémiotique. À notre connaissance, il n'existe pas de travaux proposant une formalisation informatique explicite et opérationnelle de ce processus dans les systèmes computationnels contemporains, en particulier dans le cadre des modèles d'apprentissage profond. Cette absence de contributions directement fondées sur la sémiotique peircienne pour structurer les mécanismes cognitifs du traitement de l'information justifie le positionnement adopté dans cet état de l'art. Ainsi, nous analysons principalement les modèles intégrant des connaissances linguistiques et visuelles au sein des grands modèles de langue et des architectures multimodales, non pas comme des équivalents théoriques de la triade peircienne, mais comme des approches voisines permettant d'identifier les points de convergence, les limites conceptuelles et les lacunes que notre travail vise précisément à combler.

Les modèles de transformateurs, comme le BERT et ses variantes, ont permis d'enregistrer de grandes avancées dans le domaine du NLP. Ces modèles sont principalement axés sur la modélisation de la sémantique de la langue. Ils ont permis d'obtenir d'excellentes performances dans de nombreux domaines Devlin *et al.* (2019); Liu *et al.* (2019); Lan *et al.* (2020); Sanh *et al.* (2020); He *et al.* (2021). La communauté scientifique a développé de nouvelles versions de BERT en raison des inexactitudes observées dans certains résultats obtenus pour certaines tâches en aval et en raison de l'évaluation des propriétés linguistiques de la langue naturelle Htut *et al.* (2019); Wiegreffe et Pinter (2019); Clark *et al.* (2019). Certains des modèles proposés visent à injecter des connaissances linguistiques dans les modèles de transformation, tandis que d'autres tentent d'ancrer la langue par le biais de données visuelles. Les liens syntaxiques entre les mots ne sont pas seulement ce qui confère à la langue sa richesse, mais aussi ce qui donne du sens au-delà des simples corrélations entre les mots Bai *et al.* (2021). Syntax-BERT est un modèle qui permet l'ajout des connaissances syntaxiques aux modèles de transformateurs. Il s'agit d'une extension de BERT. Il introduit des informations syntaxiques explicites par le biais d'arbres syntaxiques et donne des instructions au système d'auto-

attention concernant les dépendances linguistiques telles que le parent, l'enfant ou le frère et la sœur. Cette stratégie conserve l'expertise préentraînée de BERT tout en l'associant à la structure et à l'efficacité. Cette technique permet d'améliorer ses performances dans les scénarios d'analyse de langue naturel où la clarté syntaxique est requise. Syntax-BERT est un modèle qui permet d'intégrer des arbres syntaxiques lors de la mise au point sans qu'il soit nécessaire d'effectuer un apprentissage à partir de zéro Bai *et al.* (2021); Sundararaman *et al.* (2019). Le modèle syntaxique SGB « Syntactic Knowledge via Graph Attention with BERT » est un autre modèle proposé qui adopte l'injection de connaissances syntaxiques dans les modèles de transformateurs. SGB est un modèle dédié à la traduction automatique. Il utilise explicitement la connaissance des dépendances syntaxiques via les réseaux d'attention graphique, GAT « Graph Attention Networks» et les encodeurs basés sur BERT. Le GAT traite les structures syntaxiques comme des graphes, en améliorant les représentations des jetons grâce à des relations de dépendance. Il les combine également avec les résultats des BERT par le biais de deux méthodes. La première, appelée SGBC « Syntax-Guided BERT with Concatenation », concatène les sorties du BERT et du GAT pour attirer l'attention du codeur-décodeur. La seconde est le SGBD « Syntactic Graph-BERT Decoder-Guided Syntax ». Cette approche permet d'améliorer la fluidité de la traduction Dai *et al.* (2023). Outre le modèle syntaxique des modèles de transformation, des modèles orientés vers la vision ont vu le jour. L'un de ces modèles a été développé dans le but d'ancrer la langue naturelle dans les données visuelles : VisualBERT. Il est basé sur l'architecture de BERT. VisualBERT utilise l'alignement image-texte pour ancrer la langue dans des contextes visuels. Il utilise des couches d'attention croisée pour établir une connexion entre les modalités visuelles et textuelles. Les informations visuelles sont transmises par un réseau neuronal convolutionnel afin d'extraire des enchâssements visuels, qui sont ensuite intégrés aux enchâssements textuels. Les couches d'attention multimodale assurent une influence bidirectionnelle entre les représentations du texte et de l'image au cours du processus d'encodage. VisualBERT utilise une stratégie de fusion qui réunit les jetons textuels et les caractéristiques visuelles au sein d'un transformateur unifié Li *et al.* (2019). LXMERT, qui signifie « Learning Cross-Modality Encoder Representations from transformeurs » , est un modèle multimodal. Il traite les données visuelles et textuelles. Il utilise un mécanisme d'attention croisée pour fusionner les caractéristiques de l'image et du texte. L'architecture de LXMERT est basée sur un transformateur à deux flux. Le premier flux traite les caractéristiques visuelles. Il s'agit de régions d'images telles que des objets et des parties d'objets codées par un modèle R-CNN plus rapide préalablement entraîné. Les caractéristiques visuelles encodées sont ensuite introduites dans LXMERT pour apprendre les relations contextuelles entre les régions de l'image. Le deuxième flux traite les caractéristiques textuelles. Il comprend les enchâssements de mots de BERT. Les deux flux interagissent par l'intermédiaire de l'encodeur d'attention croisée. Cette interaction permet au modèle d'apprendre les relations

entre l'image et sa description textuelle correspondante Li *et al.* (2019). La liste des modèles multimodaux est suffisamment longue pour dépasser le nombre limité de pages du présent document. Sans disséquer les détails techniques, nous mentionnons entre autres UNITER, ImageBERT, et Multimodal-BERT, qui sont des modèles basés sur des transformateurs. Ils sont conçus pour relier les données visuelles et textuelles afin d'améliorer les performances dans les tâches multimodales (Rahman *et al.*, 2020; Chen *et al.*, 2020; Qi *et al.*, 2020). UNITER « UNiversal Image-Text Representation » apprend les encastrements conjoints par préapprentissage sur divers ensembles de données image-texte, ce qui permet de réaliser des tâches telles que la recherche d'images-texte et la réponse à des questions visuelles Chen *et al.* (2020). De même, ImageBERT dépend d'un espace d'intégration partagé et d'une interaction multimodale pour aligner le texte et les images Qi *et al.* (2020). De son côté, Multimodal-BERT personnalise l'architecture de BERT pour traiter les entrées multimodales. Il est particulièrement dédié à des applications telles que la classification d'images médicales et de textes Rahman *et al.* (2020). La communauté des chercheurs s'oriente vers l'intégration de données visuelles et textuelles pour coder le sens de la langue. Ces modèles offrent un excellent moyen d'ancrer la langue en alignant les informations visuelles, telles que les images, sur le contexte textuel. Dans les sections suivantes, nous présentons VLG-BERT, un modèle multimodal qui combine la connaissance syntaxique et l'ancrage visuel pour améliorer l'apprentissage de la représentation des mots.

2.8 Conclusion

Les modèles de langue d'aujourd'hui ne se limitent pas au traitement purement symbolique des textes. Ils prennent désormais en compte les structures syntaxiques, mais aussi visuelles, afin de mieux encoder le sens des mots. L'intégration explicite de la syntaxe permet de mieux comprendre comment les modèles de langue génèrent des représentations numériques des mots. Les modèles multimodaux, qui combinent texte et image, apportent une nouvelle dimension à l'encodage du sens des mots. L'arrimage d'images et de mots permet à ces modèles de dépasser la simple représentation textuelle et d'ancrer le monde réel dans la langue. Cela permet d'offrir davantage d'explicabilité et une vision plus complète de la manière dont les grands modèles de langue encodent le sens des mots.

CHAPITRE 3

RENFORCEMENT DE BERT AVEC UN MASQUE D'ATTENTION BASÉ SUR LE PARSEUR DE DÉPENDANCES SYNTAXIQUES

.

3.1 Détails de l'article

REINFORCEMENT OF BERT WITH DEPENDENCY-PARSING BASED ATTENTION MASK

Toufik Mechouma, Ismail Biskri and Jean-Guy Meunier

14th International Conference, ICCCI 2022, Proceedings. Communications in Computer and Information Science 1653, Springer 2022, ISBN 978-3-031-16209-1

3.2 Résumé

Cet article présente un nouveau masque d'attention basé sur l'analyse des dépendances syntaxiques DPM, afin de renforcer la fonction attentionnelle du modèle BERT. Ce masque ne remplace pas le masque de remplissage traditionnel, il a pour seul rôle d'inhiber l'attention portée aux tokens de remplissage. Parallèlement, le masque DPM utilise les graphes de dépendances syntaxiques pour générer une matrice d'adjacence qui encode les relations grammaticales entre les mots d'une phrase. À la différence du masque de remplissage, le DPM n'élimine pas des positions, mais agit comme un filtre structurel modulant l'attention. Il permet ainsi d'aiguiser l'attention sur des paires de mots syntaxiquement pertinentes. L'association de ces deux masques au mécanisme de self-attention permet d'injecter plus finement les relations linguistiques sans modifier la logique d'entraînement du modèle. Seuls les tokens réellement masqués pour la tâche MLM sont prédits. Les positions de padding ou celles filtrées par le DPM ne sont pas concernées par la prédiction. En intégrant cette connaissance syntaxique dans le mécanisme d'attention, le modèle acquiert des représentations plus structurées et plus riches sur le plan sémantique, ce qui améliore ses performances. Ce travail marque ainsi une première étape vers une meilleure synergie entre la structure linguistique explicite et les capacités d'apprentissage profond des grands modèles de langage, comme BERT.

3.3 Abstract

This paper introduces a novel attention mask based on syntactic dependency analysis, called the Dependency-based Attention Mask DPM, designed to enhance BERT's attention mechanism. This mask does not replace the traditional padding mask, which solely serves to inhibit attention toward padding tokens. In parallel, the DPM leverages syntactic dependency graphs to generate an adjacency matrix that encodes grammatical relations between words in a sentence. Unlike the padding mask, the DPM does not eliminate positions but acts as a structural filter that modulates attention. It injects syntactic knowledge with the attention me-

chanism. This sharpens the focus of attention on syntactically relevant word pairs. Combining both masks in the self-attention mechanism enables finer injection of syntactic knowledge without altering the model's training logic. Only the tokens actually masked for the MLM task are predicted. Padding positions or those filtered by the DPM are not involved in prediction. By integrating syntactic knowledge into the attention mechanism, the model acquires more structured and semantically enriched representations, leading to improved performance. This work thus represents a first step toward a better synergy between explicit linguistic structure and the deep learning capabilities of large language models like BERT.

3.4 Introduction

Long short term memory network, was a staple in deep learning Graves (2012). Although its impressive results, it has its downsides Sak *et al.* (2014). LSTM suffers from sequential processing, and poor information preservation Sak *et al.* (2014); H et S (1997). Transformers try to remedy to the previous LSTM inconveniences. They accomplish a bidirectional attention learning based on an all-to-all comparison. Transformers use a Dot-Product attention mechanism Vaswani *et al.* (2017). They are also used in the Bidirectional Encoder Representations from Transformers BERT architecture Luong (2015); Devlin *et al.* (2019). They use two learning strategies to teach BERT to represent words. Clark *et al.* (2019); Peters *et al.* (2018). In MLM, 15% of the tokens in the training dataset are masked. These masked tokens are then predicted by BERT. The second strategy is called Next Sentence Prediction (NSP). Unlike the first strategy, it learns the sentence representation. It predicts whether sentence B follows sentence A. Thus, token embeddings are learned throughout the MLM and NSP learning processes. BERT is built on a set of encoders. Each encoder is equipped with a multi-head attention mechanism MHAM. The MHAM performs parallel computing of the dot-product attention mechanism to learn the relationships between words. MHAM's output then goes through a feed-forward neural network (FFNN). The FFNN provides the learned contextualised representation. Residual connections are designed at the MHAM and FFNN outputs to add previous input data to the outputs, preserving information and avoiding signal vanishing. Furthermore, normalisation is performed at both levels Sak *et al.* (2014). The encoder stack achieves feature extraction. These features can then be used to fine-tune BERT for downstream tasks such as text classification, summarisation and translation.

3.5 Transformers

Transformers are considered to be an alternative solution to LSTMs Clark *et al.* (2019). They essentially comprise two main components. The first component is known as the 'encoder'. Each encoder has two

main units : a self-attention mechanism and a feed-forward neural network. The self-attention mechanism receives input encodings from the previous encoder and produces its own encodings. The feed-forward neural network computes the encodings from the self-attention mechanism and forwards them to the next encoder and to the second component of the transformers, the decoder Vaswani *et al.* (2017)Peters *et al.* (2018). Each decoder has three components : a self-attention mechanism ; an attention mechanism that processes encodings ; and a feed-forward neural network. The decoder's task is similar to the encoder's, but it has an additional attention mechanism that deals with the encoder's outputs. Unlike LSTMs, Transformers use parallel computation and word position encoding due to their multi-head attention blocks and position encoding algorithm, respectively.

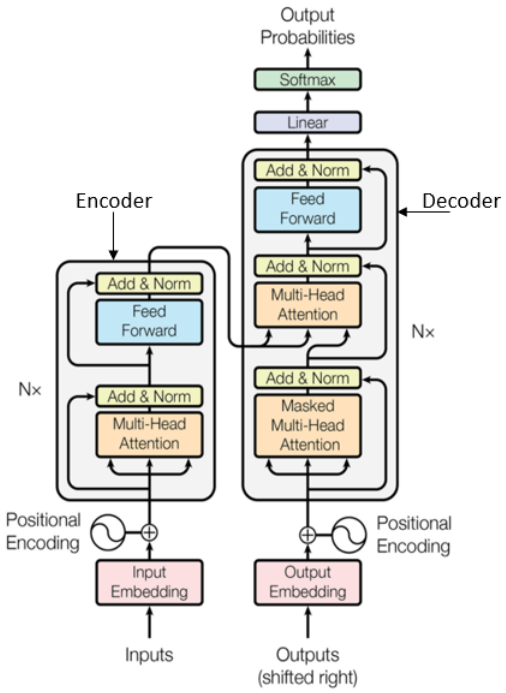

Figure 3.1 – Transformer (encoder-decoder)

3.5.1 Scaled Dot-Product Attention Mechanism

Authors in Vaswani *et al.* (2017)Luong (2015) use a dot-product attention mechanism, to learn an all-to-all attention between words, by projecting the vocabulary matrix X (embedding dimension, max sentence length) into a lower dimension Q, K and V matrices.

$$X = \begin{bmatrix} I \\ 0.63 \\ \\ . \\ . \\ 0.21 \\ 0.79 \end{bmatrix} \begin{bmatrix} Love \\ 1.25 \\ 3.65 \\ . \\ . \\ 0.44 \\ 0.01 \end{bmatrix} ..... \begin{bmatrix} Artificial \\ 5.24 \\ 3.74 \\ . \\ . \\ 0.58 \\ 1.46 \end{bmatrix} \begin{bmatrix} intelligence \\ 2.69 \\ 1.25 \\ . \\ . \\ 0.98 \\ 0.84 \end{bmatrix}$$

Figure 3.2 – Vocabulary matrix

$$Q = X \cdot W_q \tag{3.1}$$

Where W_q is a randomly initialized weight matrix, and Q is the projected query matrix

$$K = X \cdot W_k \tag{3.2}$$

Where W_k is a randomly initialized weight matrix, and K is the projected key matrix

$$V = X \cdot W_v \tag{3.3}$$

Where W_v is a randomly initialized weight matrix, and v is the projected value matrix

$$Attention(Q, K, V) = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{3.4}$$

Where K^T is the transposed key matrix, and d_k is the embedding dimension. $Q \cdot K^T$ is divided by $\sqrt{d_k}$ and followed by softmax for normalisation purpose.

For a better understanding, $Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}})$ can be considered like a filter to be applied on V, in order to compute the $Attention(Q, K, V)$. The $Attention(Q, K, V)$ is also called Scaled Dot-Product Attention. Multi-Heads attention are just a replication of h Dot-Product Attention units. Where h is a hyper-parameter that represents the number of heads per encoder, and W_q, W_k, W_v are of dimension ($d_{q,k,v}$,$d_{q,k,v}/h$)

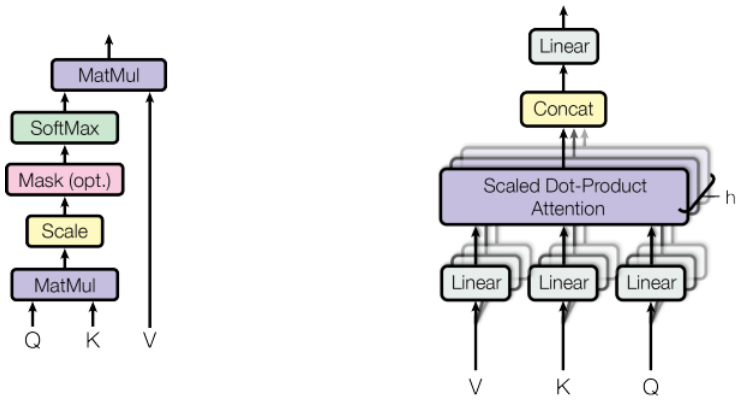

Figure 3.3 – (left) Scaled Dot-Product Attention. (right) Multi-Head attention

Note that Mask is optional as defined by the authors in Devlin *et al.* (2019).

3.5.2 Padding Mask

Since the neural network needs to have inputs that should be in similar shape and size, padding is the operation that fulfill such a requirement.

max sentence length

BERT	IS	AN	AMAZING	TOOL
I	LOVE	ARTIFICIAL	INTELLIGENCE	PAD
ME	TOO	PAD	PAD	PAD

Figure 3.4 – Padding illustration.

Padding causes problems when scaled dot-product computing is performed. The projected Q, K and V matrices contain PADs. These are considered to be like noise and need to be removed to avoid misleading results during attention computing.

SoftMax

me	to	PAD	PAD	PAD
0,25	2,73	-1e9	-1e9	-1e9
.	.	-1e9	-1e9	-1e9
.	.	-1e9	-1e9	-1e9
1,74	1,46	-1e9	-1e9	-1e9

=

me	to	PAD	PAD	PAD
0,05	0,68	0	0	0
,	.	0	0	0
.	.	0	0	0
0,25	0,17	0	0	0

Figure 3.5 – Padding Mask with SoftMax

Authors in Vaswani *et al.* (2017) add an important negative value to the corresponding PADs positions in $Q \cdot K^T$, after that, they apply a $Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}})$ to turn the negative values into zeros. The idea behind this, is to maximize the attention filter efficiency.

3.6 Proposed Mask

During BERT's implementation, we noticed that $\frac{Q \cdot K^T}{\sqrt{d_k}}$ shape is (max sentence length, max sentence length) Devlin *et al.* (2019). Thus, we wondered whether, it would be possible to add another mask, to the padding mask to improve the Scaled Dot-Product unit, and consequently, we improve the multi-head attention blocks within encoders. The proposed mask, aims to increase the quality of features extraction, by introducing a SpaCy Dependency Parsing Mask (SDPM) Honnibal et Montani (2017).

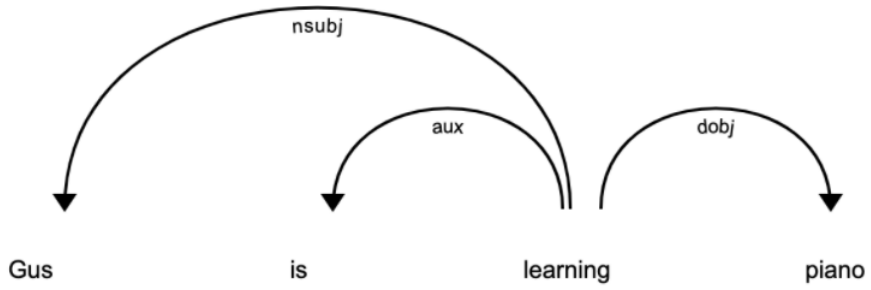

Figure 3.6 – SpaCy dependency parsing.

We first, build an adjacency matrix from the generated dependency graph. The adjacency matrix's shape is : (Max sentence length, Max sentence length).

	Gus	is	learning	piano
Gus	0	0	1	0
is	0	0	1	0
learning	0	0	0	1
Piano	0	0	0	0

Figure 3.7 – Adjacency matrix of the dependency graph.

While one value means there are direct dependencies between words, a zero value means there are no dependencies. Note that we have eliminated cases where words depend on themselves. Similarly to the padding mask, we add an important negative value to positions corresponding to zeros. Therefore, we retain the one values and add them to the attention filter.

	Gus	is	learning	piano
Gus	-1e9	-1e9	1	-1e9
is	-1e9	-1e9	1	-1e9
learning	-1e9	-1e9	-1e9	1
Piano	-1e9	-1e9	-1e9	-1e9

Figure 3.8 – Adjacency matrix after addition of an important negative value.

The adjacency matrix quantifies the semantic and syntactic relationships between words. We propose using this adjacency matrix as a second mask alongside the padding mask, as shown in equation 5.

$$Softmax((\frac{Q \cdot K^T}{\sqrt{d_k}}) + Padding_{mask} + DepParsing_{mask}) \tag{3.5}$$

	Gus	is	learning	piano	PAD	PAD
Gus	-1e9	-1e9	1	-1e9	-1e9	-1e9
is	-1e9	-1e9	1	-1e9	-1e9	-1e9
learning	-1e9	-1e9	-1e9	1	-1e9	-1e9
Piano	-1e9	-1e9	-1e9	-1e9	-1e9	-1e9
PAD	-1e9	-1e9	-1e9	-1e9	-1e9	-1e9
PAD	-1e9	-1e9	-1e9	-1e9	-1e9	-1e9

Figure 3.9 – Padding and Dependencies masks addition.

After adding both masks, we apply a softmax function to convert the negative values to zero and obtain a probability distribution. We then compute the attention as follows : : $Attention(Q, K, V) = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V$.

The proposed mask is integrated in all BERT's encoders as mentioned in the Fig.10. It takes tokens embedding vectors in input $W_1, W_2, W_3, ..., W_i$ and provide contextualized vectors $W\prime_1, W\prime_2, W\prime_3, ..., W\prime_i$

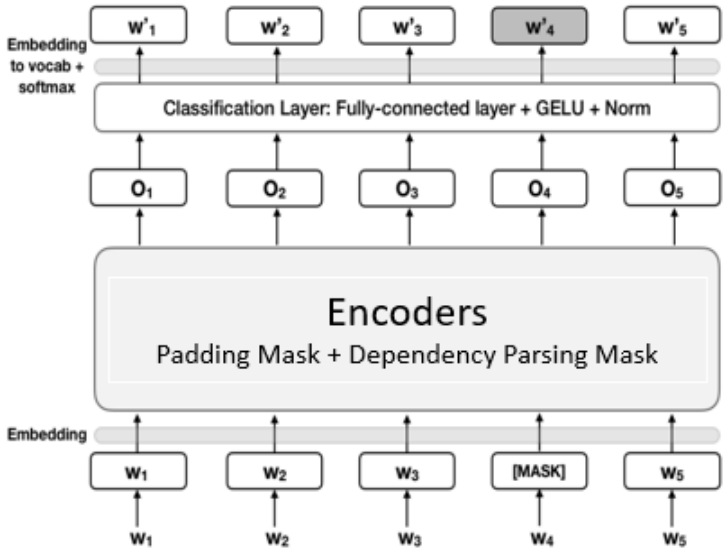

Figure 3.10 – BERT Architecture with Padding and Dependency Parsing Mask.

3.7 Experimentations

To test our model, we first implemented BERT from scratch using pytorch. We used English OpenSubtitles dataset. The dataset is available on OpenSubtitles-v2016. We performed tests on three datasets containing 100,000, 500,000 and one million sentences. To evaluate the results, we used training loss and time, with F1 scores as performance indicators. Due to hardware limitations, we performed an embedding of 50 dimensions, with a maximum sentence size of 85, rather than 768 and 512 respectively, as in BERT-base. The tests were performed on a virtual machine with an Intel(R) Xeon(R) 2.30 GHz CPU, a 46,080 KB cache size, two CPU cores and 12 GB of RAM with a CUDA GPU. The hyper-parameter values were chosen based on the hardware features and many observations. The same hyper-parameters were used for both models to enable comparison between them. The maximum sentence length is the maximum size that a sentence can be, and the batch size is used for training performance purposes. The number of segments is the number of sentences per input. The embedding dimension is the size of the vocabulary vectors. The number of encoders is the number of encoders used in the architecture of both models. The number of heads is the number of multi-head attention units per encoder in each model. The dimension of the projection matrices is represented by dim(W_q, W_k, W_v). The FFNN dimension is the dimension of the feed-forward neural network linear layer. The learning rate is used to adjust the gradient during training. Max Pred is the maximum number of tokens to be masked and predicted. Please note that, following the classical BERT strategy, only a subset of the actual content tokens are selected for prediction during MLM. Only a subset of the actual content tokens are selected for prediction during MLM, in line with the classical BERT strategy. The DPM and padding masks only affect the attention mechanism ; they do not influence which tokens are masked or predicted. In particular, PAD tokens or those filtered by the DPM are never selected for prediction. This ensures that the prediction task remains focused on meaningful linguistic content. The DPM and padding masks only affect the attention mechanism ; they do not influence which tokens are masked or predicted. In particular, PAD tokens or those filtered by DPM are never selected for prediction. This ensures that the prediction task remains focused on meaningful linguistic content. 'Nbr epochs' refers to the number of epochs required to train the models.

Table 3.1 – Dataset 1

	BERT	**BERT (DP Mask)**
Nbr of sentences	100000	100000
Hyper parameters	**BERT**	**BERT (DP Mask)**
Max sent length	85	85
batch size	10	10
nbr segments	2	2
Embedding dimension	50	50
nbr encoders	6	6
nbr heads	12	12
dim (W_q, W_k, W_v)	32	32
FFNN dim (W_o)	200	200
Learning rate	0.001	0.001
max pred	3	3
Nbr epochs	500	500
Min Loss	0.8434	$\mathbf{0.65}$
Training time (sec)	$\mathbf{179.179}$	185.991
F1-Score-mlm	0.5	$\mathbf{1}$
F1-Score-nsp	0.5	$\mathbf{0.76}$

The first test on dataset 1 shows that the performance of BERT-DPM overcomes that of BERT. We also noticed that the training time for BERT is shorter than for BERT-DPM. This is a logical result because BERT-DPM involves more computing steps than BERT.

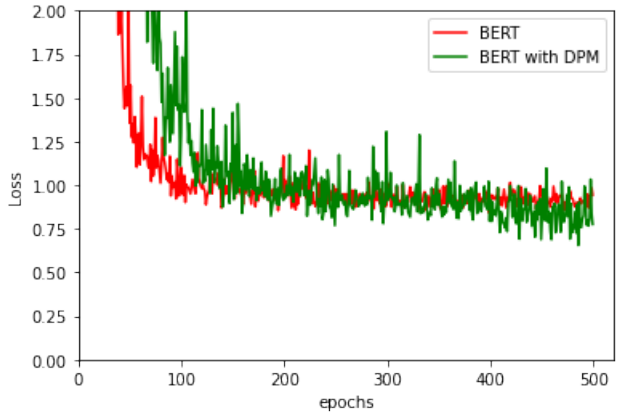

Figure 3.11 – Dataset 1 .

Table 3.2 – Dataset 2

	BERT	**BERT (DP Mask)**
Nbr of sentences	500000	500000
Hyper parameters	**BERT**	**BERT (DP Mask)**
Max sent length	85	85
batch size	10	10
nbr segments	2	2
Embedding dimension	50	50
nbr encoders	6	6
nbr heads	12	12
dim (W_q, W_k, W_v)	32	32
FFNN dim (W_o)	200	200
Learning rate	0.001	0.001
max pred	3	3
Nbr epochs	500	500
Min Loss	0.892	$\mathbf{0.428}$
Training time (sec)	$\mathbf{202.286}$	207.325
F1-Score-mlm	0.32	$\mathbf{1}$
F1-Score-nsp	0.60	$\mathbf{0.80}$

The second test on dataset 2 shows that the performance of BERT-DPM overcomes that of BERT. We also noticed that the training time for BERT is shorter than for BERT-DPM. This is a logical result because BERT-DPM involves more computing steps than BERT.

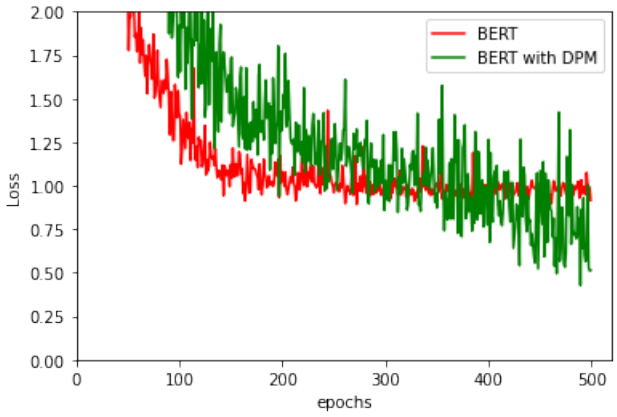

Figure 3.12 – Dataset 2 .

Table 3.3 – Dataset 3

	BERT	**BERT (DP Mask)**
Nbr of sentences	1M	1M
Hyper parameters	**BERT**	**BERT (DP Mask)**
Max sent length	85	85
batch size	10	10
nbr segments	2	2
Embedding dimension	50	50
nbr encoders	6	6
nbr heads	12	12
dim (W_q, W_k, W_v)	32	32
FFNN dim (W_o)	200	200
Learning rate	0.001	0.001
max pred	3	3
Nbr epochs	500	500
Min Loss	0.8404	$\mathbf{0.509}$
Training time (sec)	$\mathbf{208.71}$	218.521
F1-Score-mlm	0.43	$\mathbf{1}$
F1-Score-nsp	0.615	$\mathbf{0.749}$

The third test on dataset 3 shows that the performance of BERT-DPM overcomes that of BERT. We also noticed that the training time for BERT is shorter than for BERT-DPM. This is a logical result because BERT-DPM involves more computing steps than BERT.

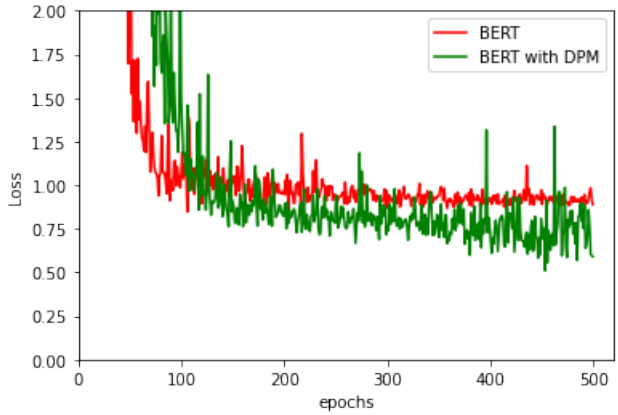

Figure 3.13 – Dataset 3 .

3.8 Conclusion

The experimental results show that adding syntactic structure to BERT's attention mechanism using a dependency parsing mask (DPM) greatly improves the model's performance. This improvement is consistent across datasets of different sizes. The DPM works alongside the traditional padding mask. While the padding mask eliminates the influence of non-informative PAD tokens by setting their attention weights to zero, the DPM introduces a structural prior over the sentence by directing attention towards pairs of tokens that share direct syntactic dependencies. It is important to note that this mask does not aim to eliminate tokens, but rather to refine the attention distribution and improve the quality of the learned contextual representations. Furthermore, the DPM is only applied within the attention computation and does not influence the selection of tokens for prediction in the MLM task. Only content tokens (not PAD or syntactically filtered tokens) are masked for prediction, which is consistent with standard BERT training. These findings support the idea that incorporating linguistic knowledge into Transformer attention mechanisms can improve model interpretability and performance, particularly in contexts where resources are limited or the syntax is complex.

3.9 Perspective

The experiments presented in this work demonstrate the effectiveness of the proposed dependency-based masking mechanism in improving Scaled Dot-Product Attention within the BERT architecture. This is achieved by integrating syntactic information into the attention mechanisms. Beyond its empirical contributions, this research provided an opportunity to rebuild the BERT architecture from scratch, offering a deeper un-

derstanding of its internal mechanisms and providing a solid foundation for future architectural innovations. A critical insight gained through this implementation is that BERT's original random masking strategy overlooks the linguistic structure inherent in natural language. The next logical step is to develop a hybrid masking strategy that combines traditional random token masking with syntax-aware dependency masking. This combined approach is expected to preserve the benefits of randomness in terms of generalisation while also incorporating linguistic priors that can enhance contextual learning. This approach aims to refine pretraining objectives and contribute to a broader research agenda. The deeper syntactic theory is integrated into neural models, the more interpretable, robust and semantically grounded NLP systems become, bringing us closer to linguistically informed language understanding.

CHAPITRE 4

LINGBERT, VERS L'INJECTION DE LA CONNAISSANCE LINGUISTIQUE DANS UN MÉCANISME D'ATTENTION BASÉ SUR UNE STRATÉGIE DE MASQUAGE HYBRIDE

.

4.1 Détails de l'article

LingBERT, Linguistic Knowledge Injection into Attention Mechanism based on a Hybrid Masking Strategy

Toufik Mechouma, Ismail Biskri and Serge Robert

23rd International Conference on machine learning and applications, ICMLA 2024,Miami, Florida, USA.
1946-0759/24/©2024 IEEE DOI 10.1109/ICMLA61862.2024.00253

4.2 Résumé

Dans cet article, nous présentons LingBERT, un modèle de langage basé sur les transformers. Nous présentons deux architectures de LingBERT basées sur une stratégie de masquage hybride. Ces deux modèles s'inspirent de BERT Base. Notre modèle introduit l'injection de connaissances linguistiques (dépendances syntaxiques) dans les mécanismes d'attention. Cependant, BERT et certaines de ses variantes utilisent un masquage aléatoire des tokens pendant l'entraînement, ce qui peut entraîner une capture inefficace des dépendances syntaxiques et sémantiques. Pour remédier à ce problème, notre méthode combine un masquage aléatoire et un masquage sélectif. Elle consiste à masquer les mots ayant des dépendances syntaxiques, tout en masquant un faible pourcentage de mots de manière aléatoire. Les tokens résultant de cette stratégie sont ensuite transmis à deux versions de lingBERT. Cette stratégie de masquage garantit que les relations linguistiques sont préservées et apprises de manière plus efficace. De plus, nous maintenons un faible taux de masquage aléatoire afin d'éviter le surapprentissage. Grâce à des expérimentations et des évaluations, notre approche permet d'améliorer significativement la capture du contexte et d'optimiser les performances dans diverses tâches de traitement du langage naturel. Elle permet également de réduire la complexité du modèle BERT. Notre approche offre également une interprétation du fonctionnement interne du modèle à chaque étape de l'apprentissage. Notre travail propose une nouvelle approche qui consiste à injecter des connaissances dans les modèles de langage basés sur les mécanismes d'attention, afin d'améliorer leurs capacités d'encodage du sens tout en les optimisant.

4.3 Abstract

In this paper, we present lingBERT, a transformer-based language model. We present two lingBERT architectures based on a hybrid masking strategy. Both models are inspired by the BERT base model. Our model incorporates linguistic knowledge (syntactic dependencies) into attention mechanisms. Models such

as BERT employ random masking of tokens during training, which can result in an inefficient capture of syntactic and semantic dependencies. To address this issue, our method employs two masking strategies. The first masks words with syntactic dependencies. The second uses a low percentage of randomly masked words. Tokens resulting from both strategies are then processed. over to the two proposed lingBERT architectures. This strategy ensures that linguistic relationships are preserved and learnt more effectively. More effectively. Additionally, we maintain a low level of randomness ratio of masked tokens to prevent overfitting and improve model generalisation model's ability to generalise. Through comprehensive experiments and our approach has been shown through extensive experimentation and evaluation to significantly improve context capture, leading to better performance across various NLP tasks. Furthermore, our approach provides insight into the inner workings of our model throughout the learning process. This work opens up a new avenue for knowledge injection into attention-mechanism-based models, thereby advancing the capabilities of language understanding systems.

4.4 Introduction

Transformer models, particularly BERT and its derivatives, have transformed natural language processing (NLP) by delivering state-of-the-art results in various tasks. These models rely on attention mechanisms, which allow them to identify long-range dependencies within text. The majority of these models use a token masking strategy. On the one hand, RoBERTa uses the same model architecture as BERT. However, its improved performance is due to its larger training data and extended training time. One of RoBERTa's major innovations is its use of dynamic masking. In BERT, masked language modelling (MLM) involves applying a static mask once during pre-training. In contrast, RoBERTa applies a new mask at every epoch, meaning the tokens to be masked are chosen differently each time a training example is encountered. Liu *et al.* (2019). Conversely, ALBERT (A Lite BERT) introduces several efficiency improvements over BERT by focusing on reducing model size while maintaining performance. Like RoBERTa, ALBERT uses dynamic masking, whereby the tokens to be masked are randomly selected and may vary during each training epoch. Lan *et al.* (2020). Similarly, DeBERTa uses dynamic masking strategies, like those in RoBERTa, where the masking pattern changes during training. This helps to prevent the model from becoming over-reliant on specific masked positions. He *et al.* (2021). By contrast, SpanBERT uses span-based masking. Rather than masking individual tokens, it masks neighbouring spans of tokens. In other words, rather than randomly selecting individual tokens to mask, it selects entire spans of text Joshi *et al.* (2020). Although, TinyBERT, DistilBERT and SciBERT retain the same static masking strategy used in BERT Sanh *et al.* (2019) Jiao *et al.* (2020) Beltagy

et Cohn (2020). Masking involves obscuring tokens in a sequence, regardless of the text's linguistic structure. Although this method is effective in generalising across a wide range of contexts, it can inadvertently reduce the model's ability to capture intricate syntactic and semantic relationships between words. This can result in an inability to grasp the deeper contextual meaning of phrases, particularly in more complex linguistic constructions. Clark *et al.* (2019) Tenney *et al.* (2019b) Alex *et al.* (2019). Some models, unlike BERT, do not use the masking strategy. For example, XLNet uses a permutation language modelling objective. This means that it considers all possible permutations of the input sequence and predicts the tokens within these orders. Yang *et al.* (2019). Another model, called ELECTRA, uses a replaced token detection approach. In this model, the generator network replaces some of the tokens with incorrect words, and the model is then trained to distinguish between the real tokens and the replaced ones Clark *et al.* (2020). Generative models similar to GPT also use a generative pre-training approach based on a transformer decoder architecture. These models are trained to predict the next token in a sequence. Brown *et al.* (2020c). Despite their success, these models have inherent limitations in how they handle syntactic and semantic dependencies during training. Mask-based models, for example, suffer from limitations in interpretability. These models mask parts of the input text in order to predict missing tokens. Conversely, unmasking-based models, such as those involving auto-regressive generation, are difficult to understand. How does any single word influence the prediction ? How can we interpret and explain the cumulative context of all preceding words ? Regardless of their success, these models might require further enhancement to handle long-term dependencies properly and avoid issues such as maintaining coherent and accurate text, since early mistakes can affect the rest of the output Clark *et al.* (2019) Tenney *et al.* (2019b) Alex *et al.* (2019) Jain et Wallace (2019). The complexity of the interactions between the tokens, coupled with the probabilistic nature of these masking- and unmasking-based models, obscures the reasoning behind their outputs and complicates efforts to interpret and explain the models' behaviour. These limitations emphasise the trade-offs between different modelling strategies and their effect on the interpretability of language models Holtzman *et al.* (2020) Tan *et al.* (2020) Ribeiro *et al.* (2016b). To address these shortcomings, we present two versions of LingBERT. Both versions use a hybrid masking strategy to inject linguistic knowledge into our models. The hybrid masking strategy has already been mentioned Zhang *et al.* (2021). Authors explore hybrid masking strategies and their impact on text classification tasks, reinforcing the benefits of combining different masking approaches Zhang *et al.* (2021). The hybrid masking strategy combines the advantages of structured and random masking. Specifically, our model prioritises masking words with syntactic dependencies to ensure these linguistically linked tokens are processed together within a particular head's attention. Additionally, incorporating a small proportion of randomly masked tokens mitigates the risk of overfitting,

thereby enhancing the model's generalisation capabilities. This approach enables the model to learn and preserve critical linguistic relationships more effectively. The hybrid nature of our masking strategy improves the model's contextual understanding and increases the interpretability of the learning process, providing clearer insights into how linguistic knowledge is represented and utilised within the model. The proposed approach is inspired by Mechouma *et al.* (2022a). The authors propose improving BERT's performance by introducing a dependency parsing mask to the multi-head attention mechanism. This mask complements the existing padding mask, which is used to filter padding positions. Experiments have demonstrated that incorporating the dependency-parsing mask improves BERT's attention filtering Mechouma *et al.* (2022a). In this paper, we first present the theoretical details of the masking strategies. We then discuss some specifics about the training dataset. We also analyse the architectures of both lingBERT versions. Finally, we conclude with an experimental evaluation of our models.

4.5 Theoretical Background

4.5.1 Hybrid Masking Strategy of Tokens

To improve the performance of natural language transformer-based models, we present a new attention mechanism that uses a hybrid masking strategy. This approach is designed to capture syntactic relationships while maintaining model robustness through controlled randomness. The hyperparameters in our model, such as the masking percentage for syntactic and random tokens, were carefully selected through a series of experiments to optimise the model's performance.

4.5.1.1 Syntactic Dependency-Based Masking

We use spaCy tool due to its post-tagging and parsing high accuracy, 95.1%, 97% respectively. The accuracy report is available on the official website. During the training process, we first perform syntactic parsing on each sentence or segment. This identifies tokens with specific syntactic dependencies without considering dependency types. These dependencies are crucial for capturing the syntactic and semantic essence of the sentence effectively. Distinctly to BERT's masking strategy, we mask 20% of syntactic-depended tokens. The masking percentage is a tunable hyper-parameter. Experiments revealed and justified the choice of 20% due to the outstanding results.

4.5.1.2 Random Masking

In the random masking strategy, 10% of tokens without syntactic dependencies are masked at random. This masking serves as our metric to avoid overfitting that may result from masking based on syntactic dependencies. The masking percentage is a separate, tunable hyperparameter that differs from that used in the first strategy. Experiments revealed that the choices of 10% for the lingBERT.v1 and 15% for lingBERT.v2 grant an outstanding performance.

4.5.1.3 Training Dataset Formatting

Both strategies required the training dataset to be pre-processed prior to the training phase. This dataset was meticulously pre-processed and formatted before being introduced to the model during runtime. Based on dependency parsing, pairs of tokens are grouped according to their syntactic relationships. Focusing on preserving these relationships, BERT employs a strategy of syntactic masking in conjunction with randomly selecting tokens from each sentence, ensuring these tokens do not overlap with those selected for masking. These tokens are replaced in the same way as the Masked Language Modeling (MLM), with a special [MASK] token in the sentence.

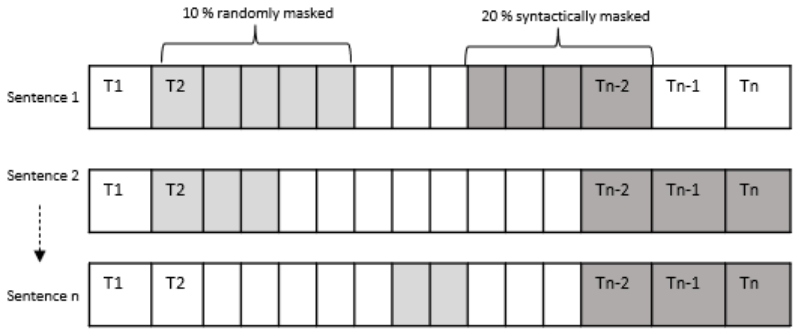

Figure 4.1 – TRAINING DATASET FORMAT FOR lingBERT V1.

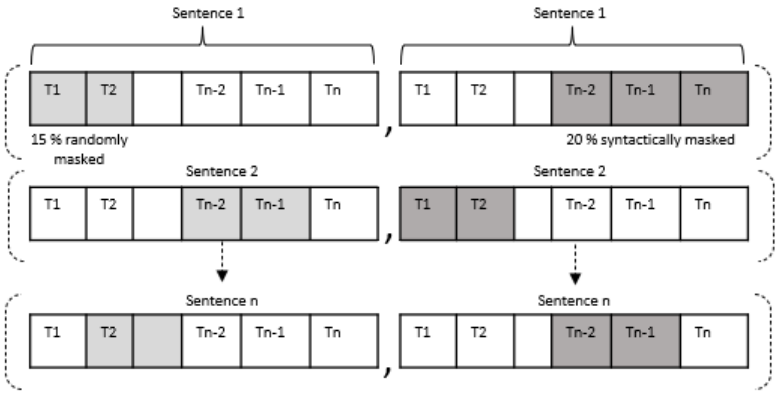

Figure 4.2 – TRAINING DATASET FORMAT FOR lingBERT v2.

4.5.2 Theoretical Foundation of The architecture

The proposed masking strategies raise a critical question about the architecture. They ask whether it is possible to integrate both types of tokens, which come from syntactic and random masking, within Heads-Attention along the encoders stack. Alternatively, separate stacks of encoders could be created, with each stack fully dedicated to learning the specific patterns and relationships of its masking strategy. Both architectures have been implemented and tested. Architectural schemas are analysed in the next section. The attention mechanism used is the same as the one proposed in Vaswani *et al.* (2017).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where :

- First point Q is the matrix of queries.
- K is the matrix of keys.
- V is the matrix of values.
- d_k is the dimensionality of the keys (or queries).

4.5.2.1 lingBERT v1

The first architecture is almost identical to the BERT Base architecture. The only difference is that the number of encoders is 7, rather than 12, and each encoder has 7 Attention-Heads rather than 12. The number of encoders was reduced in each experiment without compromising performance. The optimal hyperpa-

rameters were determined through a series of experimental trials. Another key difference is the format of the training dataset : lingBERT v1 processes sentences containing random and syntactic masked tokens simultaneously.

4.5.2.2 lingBERT v2

The second architecture consists of two stacked encoders. The first is specialised in learning representations based on random masking. It has 12 encoders, each with seven Attention-Heads. The latter learns representations using a syntactic-based masking strategy. It has 12 encoders with three Attention-Heads each. The number of encoders is reduced based on performance. There is a shared embedding matrix that is randomly initialised. This matrix is used to generate the initial embeddings for the input tokens. These embeddings are then processed in parallel through two stacks to produce two different outputs. During training, both stacks independently update their internal parameters based on their respective inputs. After passing through the stacks, the outputs are averaged to produce a combined output. This averaged output is then fed into the final layer to predict the masked tokens. The error (or loss) is calculated by comparing the predicted and actual tokens. Both stacks receive the same error signal because the final prediction is based on their combined output. As the two stacks are fed embeddings derived from the same initial embedding matrix, the gradients from the back-propagation process are used to update this shared matrix. This means that the gradients from the syntactic and random masking stacks influence how the embeddings are adjusted during training. LingBERT v2 processes pairs of sentences, each with a different selection of tokens. These are the randomly selected tokens and the syntactically based masked tokens. Similarly, in each architecture, feed-forward neural networks have a hidden size of 768. Also, residual information is added and normalised throughout each encoder. This helps preserve information and prevent gradient vanishing or exploding. The final output vectors from the two architectures are averaged and sent through a prediction layer comprising 30,000 elements. At the end of the word representation learning process, the corresponding embedding vectors are extracted from the final encoder output. These vectors are then used for testing purposes in downstream tasks.

4.6 Architectures

In the following, we describe the architecture of each model.

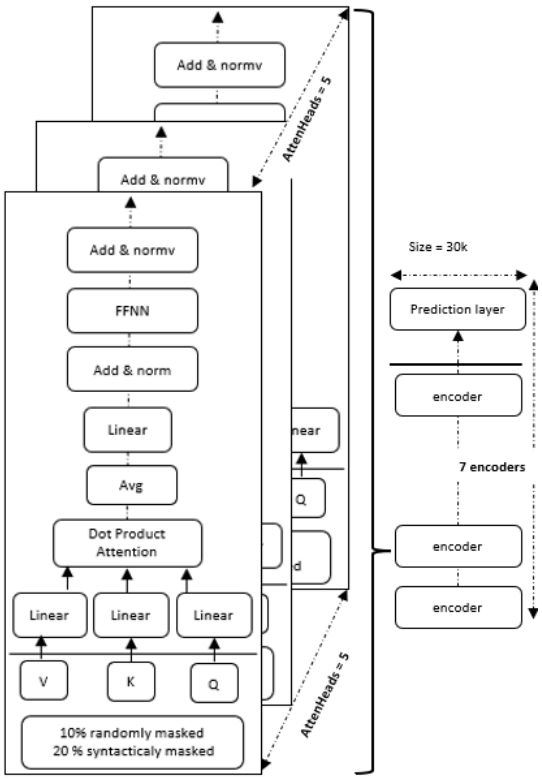

Figure 4.3 – lingBERT V1.

The lingBERT V1 architecture contains 7 encoders with 7 Attention-Heads per encoder.

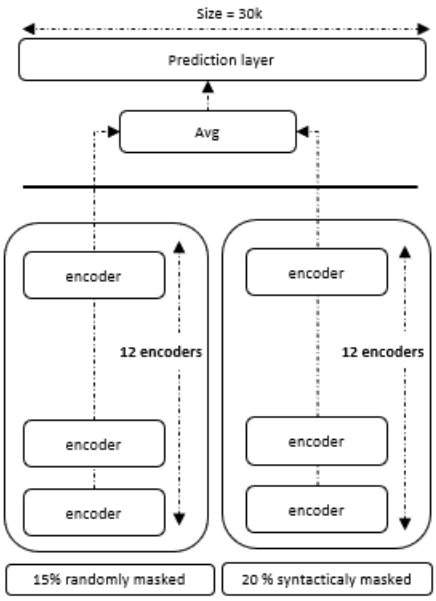

Figure 4.4 – lingBERT V2.

The lingBERT V2 architecture consists of two encoders-stackes. The former is specialized in learning representation based on random masking. It has 12 encoders, 7 Attention-Heads each. The latter learns representation from syntactic based masking strategy. It has 12 encoders, 3 Attention-Heads each.

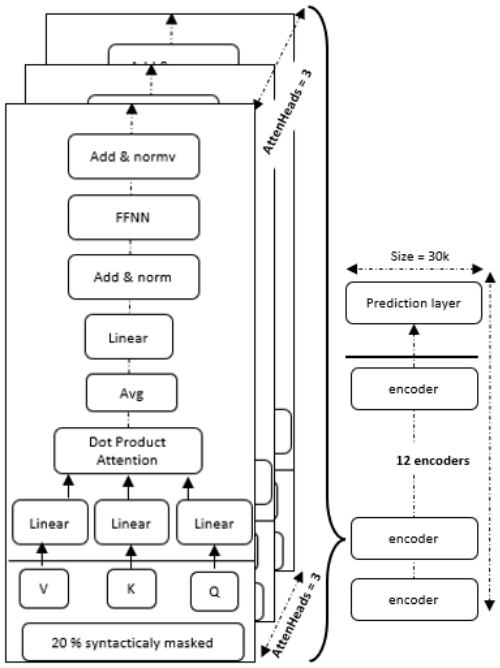

Figure 4.5 – SYNTACTIC-MASKING FOR lingBERT v2.

The Figure 4.5 represents the architecture of lingBERT v2 with 20% of syntactically masked tokens for the first encoders-stack.

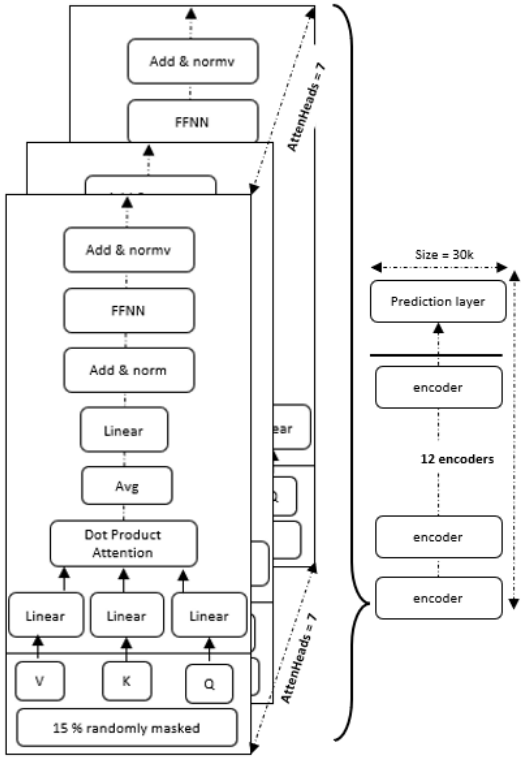

Figure 4.6 – RANDOMLY-MASKING FOR lingBERT v2.

The Figure 4.6 represents the architecture of lingBERT v2 with 15% of randomly masked tokens for the second encoders-stack.

4.7 Experiments

For evaluation and testing purposes, we used the same training dataset as BERT : the English Wikipedia dump and the Bookscorpus, which is a 16 GB Wikipedia dump of text after processing and cleaning. This represents plain text extracted from Wikipedia articles, excluding lists, tables and other non-textual content. The BookCorpus dataset is a large-scale text corpus containing over 11,000 free, unpublished books available online. To enable comparison with BERT and its derived models, we opted for a high-performance hardware configuration. Training was performed on a commercial cloud with eight NVIDIA Tesla K80 GPUs. The training dataset was formatted in almost 17 hours in accordance with the above models' requirements. To evaluate our two models, we used text classification as a downstream task for both GLUE and AG News

to evaluate the generated embeddings by lingBERT.

4.8 Findings

The bellow benchamrking tables show the performance of our two models.

Model	Training Time	Dataset	Glue Accur
lingBERT.v1	8 days	16 GB Wiki+BookCor	0.937
lingBERT.v2	11 days	16 GB Wiki+BookCor	0.944
BERT	4 days	16 GB Wiki + BookCor	0.858
RoBERTa	1-2 weeks	160 GB of text data	0.945
DistilBERT	Few days	16 GB Wiki+BookCor	0.908
ALBERT	Several days	16 GB Wiki+BookCor	0.911

Table 4.1 – Comparison of various NLP models

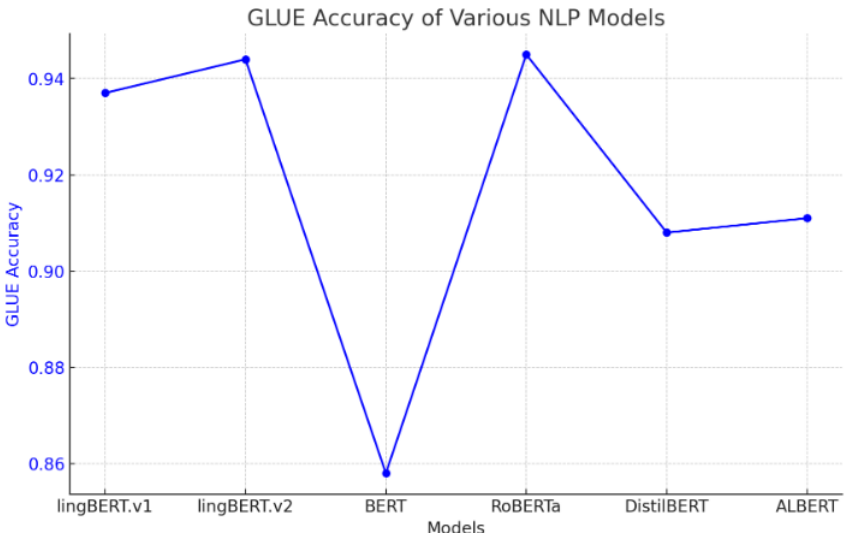

Figure 4.7 – GLUE SCORES OF THE NLP MODELS

The statistics compare NLP models, focusing on training time, dataset size, and GLUE scores. BERT, as a baseline, scores 0.858 while lingBERT versions, with longer training on the same dataset, perform better (0.937 and 0.944). RoBERTa, trained on a much larger dataset, achieves a highest score of 0.945, showing the advantages of more data. DistilBERT, designed for efficiency, scores 0.908, balancing speed and accuracy. ALBERT, reaches the score of 0.911. LingBERT v1 & v2 outperform BERT, DistilBERT and ALBERT showing the

efficiency of the proposed architectures and masking strategy.

Model	Dataset	Accuracy	F1 Score
lingBERT.v1	AG News	93.7%	92.3%
lingBERT.v2	AG News	94.4%	93.5%
BERT	AG News	93.3%	92.1%
RoBERTa	AG News	94.5%	93.5%
DistilBERT	AG News	90.08%	90.2%
ALBERT	AG News	94.2%	93.1%

Table 4.2 – Benchmarking Text Classification Accuracy and F1 Score

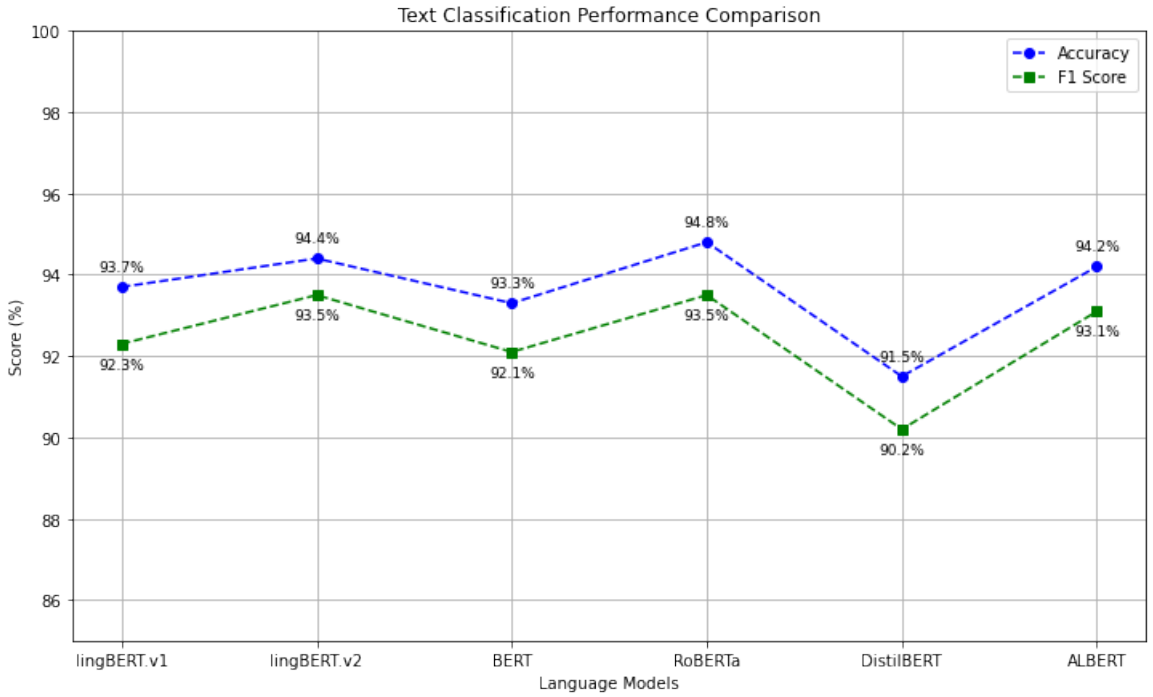

Figure 4.8 – ACCURACY AND F1 SCORE OF NLP MODELS ON AGNEWS

Overall, RoBERTa and lingBERT.v2 are the models that perform best on the AG News dataset, offering the highest accuracy and F1 scores. ALBERT also performs well, making it particularly appealing for applications where model size and efficiency are important. BERT and lingBERT.v1 provide solid baseline results, though they are outperformed by their newer counterparts. Although DistilBERT trails behind in terms of accuracy and F1 score, it offers a valuable alternative for scenarios where computational efficiency is more important

than achieving the highest possible classification performance.

4.9 Conclusion

In conclusion, the lingBERT models, both v1 and v2, offer significant advancements in NLP performance through their innovative hybrid masking strategy. LingBERT.v1 demonstrates strong performance on the AG News dataset, achieving 93.7% accuracy and a 92.3% F1 score, effectively balancing accuracy with contextual understanding. LingBERT.v2 builds upon this foundation with further enhancements, resulting in even higher accuracy (94.4%) and F1 score (93.5 %), showcasing notable improvements in capturing and leveraging syntactic dependencies. By integrating syntactic knowledge into the attention mechanisms and employing a refined masking approach, lingBERT models enhance both the interpretability and effectiveness of text classification tasks. These improvements highlight the models' ability to preserve critical linguistic relationships and outperform existing benchmarks, representing a significant step forward in the development of systems that understand language. Also, both versions of LingBERT significantly simplify the architecture compared to the BERT Base model. Another added value is the formatted training dataset, which reduces the computing complexity that would otherwise occur during training. Our results demonstrate that the proposed approach significantly improves performance on various NLP tasks, offering a promising new way to integrate linguistic knowledge into transformer models. By advancing the capabilities of attention-based systems, our work contributes to the ongoing evolution of natural language understanding technologies. Further research is needed to explore ways of injecting linguistic knowledge into large language models.

4.10 Perspective

Encouraged by the results of lingBERT, our next line of inquiry will focus on integrating linguistic knowledge more deeply into the training objectives of large language models. Although lingBERT introduced syntactic information via a hybrid masking strategy, demonstrating its ability to enhance interpretability and task performance, it still operates primarily within a statistical paradigm, whereby language structure emerges implicitly from data-driven learning. Our next work aims to move beyond this by explicitly supervising the learning process using syntactic trees as the ground truth. Specifically, we propose encoding syntactic structures as binary adjacency matrices that represent dependency relationships between words. These matrices will serve as supervisory targets for the model, enabling it to learn from both masked token prediction and structured linguistic constraints. This approach is implemented in our proposed SCA-BERT (Syntax-Constraint-Aware BERT) model, in which the model's attention mechanism directly encodes

and predicts syntactic dependencies. Syntactic constraints are enforced during training using augmented Lagrangian optimization, guiding the model to align its internal representations with known syntactic structures. This represents a shift from treating syntactic features as auxiliary inputs to making them central supervisory signals in model learning.

CHAPITRE 5

SCABERT : LA CONNAISSANCE SYNTAXIQUE COMME UNE VÉRITÉ DE TERRAIN POUR LA SUPERVISION D'UN MÉCANISME D'ATTENTION GUIDÉ PAR CONTRAINTE VIA LAGRANGE AUGMENTÉ

.

5.1 Détails de l'article

Syntax-Constraint-Aware SCABERT : Syntactic Knowledge as a Ground Truth Supervisor of Attention Mechanism via Augmented Lagrange Multipliers

Toufik Mechouma, Ismail Biskri and Serge Robert

Proceedings of Tenth International Congress on Information and Communication Technology - ICICT Feb 2025, London, UK. (accepted not yet published)

5.2 Résumé

Cet article présente une variante de BERT qui utilise une technique permettant d'injecter des connaissances linguistiques sous forme de contrainte. Il s'agit d'un nouveau modèle qui tire parti de la technique d'optimisation basée sur les multiplicateurs de Lagrange augmentés. Le modèle utilise les dépendances syntaxiques encodées dans une matrice d'adjacence correspondant à l'arbre syntaxique afin de superviser le processus d'apprentissage de la représentation des mots. Cette méthode garantit que la structure syntaxique influence les représentations des mots du modèle. L'application de l'optimisation lagrangienne augmentée permet d'imposer des contraintes au mécanisme d'attention, facilitant ainsi l'apprentissage des relations syntaxiques. Cette approche consiste à modifier l'architecture standard du BERT, notamment la couche de prédiction. L'objectif est de prédire une matrice d'adjacence qui encode les relations syntaxiques entre les mots, plutôt que d'utiliser des jetons masqués. Les résultats de nos expériences montrent que l'injection de connaissances syntaxiques permet d'améliorer les performances par rapport au BERT, notamment en ce qui concerne le temps d'apprentissage et la classification des textes d'AG News en tant que tâche en aval. En combinant la flexibilité de l'apprentissage profond avec des connaissances linguistiques structurées, nous fusionnons les approches ascendantes et descendantes. Notre modèle permet également d'améliorer l'interprétabilité et les performances des modèles de langage.

5.3 Abstract

This paper introduces Syntax-Constraint-Aware BERT (SCA-BERT), a novel variant of BERT that uses augmented Lagrange multipliers to inject syntactic knowledge into the attention mechanism. The model uses syntactic dependencies as a form of ground truth to supervise the learning of word representations, ensuring that syntactic structure influences the model's representations of words. Applying augmented Lagrangian

optimization imposes constraints on the attention mechanism, facilitating the learning of syntactic relationships. This approach augments the standard BERT architecture by modifying the prediction layer. The aim is to predict an adjacency matrix encoding words' syntactic relationships instead of masked tokens. Our experiments demonstrate that injecting syntactic knowledge improves performance in terms of training time compared to BERT, and also on AG News text classification as a downstream task. By combining the flexibility of deep learning with structured linguistic knowledge, we merge bottom-up and top-down approaches. Furthermore, Syntax-Constraint-Aware BERT improves the interpretability and performance of Transformer-based models.

5.4 Introduction

Natural Language Processing NLP has witnessed transformative advancements in recent years, primarily driven by the advent of transformer architectures such as BERT Devlin *et al.* (2019). These models leverage self-attention mechanisms to capture complex relationships between words, enabling a deeper understanding of contextual information. Despite their impressive performance across various tasks, many transformer models often lack explicit incorporation of linguistic structures, such as syntactic dependencies, which are critical for nuanced language comprehension Htut *et al.* (2019) Mechouma *et al.* (2024). Also interpretability remains a challenge for transformer models Jain et Wallace (2019). By representing these relationships as graphs, researchers have demonstrated the effectiveness of syntactic structures in improving NLP tasks such as machine translation, information extraction, and sentiment analysis Marcheggiani et Titov (2017) Li *et al.* (2023). For instance, authors in Wu *et al.* (2018) employed dependency parsing in their neural machine translation framework, highlighting that syntactic structures can significantly enhance translation quality by preserving grammatical relations across languages. Furthermore, recent studies have aimed to integrate syntactic information more deeply into transformer models. Syntax-aware Transformers, such as StructBERT Peng *et al.* (2019), have shown that incorporating syntactic trees into the attention mechanisms improves performance on downstream tasks like semantic role labeling and question answering. Similarly, authors in Bai *et al.* (2021) proposed a method to integrate syntactic knowledge by augmenting BERT with a syntactic dependency tree, leading to improved accuracy in various NLP benchmarks. Despite these advancements, most existing approaches treat syntactic information as auxiliary or secondary to the primary learning objective. They often use syntactic features as inputs rather than establishing them as direct targets for model learning. This gap presents an opportunity to explore the potential of using syntactic structures, such as the adjacency matrix of dependency graphs, as primary targets in training deep learning

models Mechouma *et al.* (2022b).

In optimization field, Augmented Lagrange Multipliers serve to incorporate constraints into objective functions, facilitating solutions to constrained problems. Augmented Lagrangian methods enhance this approach by adding penalty terms that enforce constraints more effectively during the optimization process. These methods have been employed in machine learning, where they have shown success in tasks requiring the satisfaction of multiple constraints Narasimhan *et al.* (2020). The application of Augmented Lagrangian techniques in the context of NLP remains relatively unexplored. However,learning with constraints in structured prediction tasks, suggest that these methods could be beneficial in integrating syntactic constraints into model learning Pan *et al.* (2020).

This paper presents a new architecture for replacing masked tokens in BERT with a binary adjacency matrix that represents syntactic dependencies. Treating this matrix as the target for model predictions enables the model to learn word representations that closely adhere to syntactic structures. Additionally, Augmented Lagrange multipliers are employed to introduce dynamic penalties for constraint violations during training, thereby promoting a more structured and linguistically informed learning process. By aligning the learning of word representations with syntactic dependencies, our approach seeks to enhance both the performance and interpretability of models on various NLP tasks. Our work bridges the gap between deep learning methodologies and linguistic theory by providing insights into how structured knowledge can be integrated into modern NLP systems.

5.5 Conceptual Model

The model is based on the fundamental tenets of transformer architectures and incorporates syntactic dependencies through the use of an adjacency matrix, M. This serves to encode the syntactic dependencies. During the training phase, it is employed as the target ground truth to facilitate convergence towards it. The positional encoding is kept as in BERT base, while the next sentence prediction is not integrated. This section will delineate the various layers and components that comprise the model.

5.5.1 Input Layer

The input comprises word embeddings, represented as a matrix $E \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. The model takes both tokens and position embeddings as

input to the Transformer layers.

$$E = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1d} \\ e_{21} & e_{22} & \cdots & e_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nd} \end{pmatrix}$$

Figure 5.1 – Word embedding matrix $E \in \mathbb{R}^{n \times d}$, where n is the number of words in the sentence and d is the embedding dimension.

5.5.2 Syntactic Dependencies Encoding

A binary adjacency matrix, $M \in \mathbb{R}^{n \times n}$, is incorporated into the model, to encode syntactic dependencies, where n is the number of words in a sentence. If word i has a direct dependency on word j, the corresponding entry in the matrix $\mathbf{M}$ is set to 1, indicating a dependency. Otherwise, the entry is set to 0. This matrix serves as a ground truth and a target for the model to learn during training.

$$M = \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{pmatrix}$$

Figure 5.2 – Matrix M representing syntactic dependencies between words.

5.5.3 Encoders Stack

Subsequently, we present the encoder stack, which is structured in accordance with the architectural principles of BERT Base. The encoder stack comprises a series of 12 Transformer layers, 12 attention heads, 768 hidden size, 512 maximum sentence length which perform attention-based learning over the input embeddings.

5.5.4 Prediction Layer

The input to the prediction layer is the output from the last encoder layer, denoted as matrix $H \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. To generate the syntactic dependency matrix A of shape $n \times n$, where n is the number of words in the input sentence. The model uses a fully connected (dense) layer that takes the encoded word representations H and maps them to an adjacency matrix representing the syntactic dependencies as follows.

$$A = \text{softmax}(H \cdot W) \tag{5.1}$$

Where : $H \in \mathbb{R}^{n \times d}$ is the output of the encoder stack.

$W \in \mathbb{R}^{d \times n}$ is a learnable weight matrix of the prediction layer.

$A \in \mathbb{R}^{n \times n}$ is the predicted syntactic adjacency matrix, representing the dependencies between the tokens in the input sequence. The output values $A_{ij} \in [0, 1]$ represent the strength of the syntactic dependency between the words i and j. A value close to 1 indicates a strong dependency, while a value close to 0 indicates weak or no dependency.

$$\begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1d} \\ h_{21} & h_{22} & \cdots & h_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nd} \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} & \cdots & w_{dm} \end{pmatrix} = \quad \text{softmax} \left(\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \right) = A$$

Figure 5.3 – Matrix A representing predicted attention weights.

5.5.4.1 Softmax or Sigmoid ?

in our context the question ties directly into the concepts of dependent and independent variables in the field of probability. From linguistic perspective, words are connected by syntactic dependencies, and these dependencies usually carry semantic meaning. By applying softmax, we introduce a distributional hypothesis where words with strong syntactic relationships have higher probabilities compared to unrelated words, which is closer to how humans understand the language words. In the case of sigmoid activation, we treat

the syntactic relationships between words as independent events. In other words word-pairs are processed in isolation. From computational perspective, by introducing probability distribution, softmax squashes negative values towards zero and brings probabilities to one for relevant relationships, which is beneficial when used with the Lagrangian multiplier to converge quickly to a binary adjacency matrix. One potential downside of softmax is enforcing mutual exclusivity in its outputs, which could be problematic because a word can have multiple syntactic relationships simultaneously. In our case, softmax makes more sense than sigmoid, especially when the goal is to inject syntactic knowledge in a more controlled manner by encouraging a probability distribution over syntactic dependencies.

5.6 Augmented Lagrangian Formulation

The Augmented Lagrangian method represents an extension of the classical Lagrangian approach to optimisation, particularly suited for handling constraints in problems where traditional Lagrangian multipliers may be insufficient. In the present context, the Augmented Lagrangian framework is applied to enforce syntactic dependencies during the learning of word representations in a Transformer-based model. The mathematical foundation involves modifying the objective function by incorporating a penalty term to enforce the constraint.

The choice of the Augmented Lagrangian method is driven by the non-convex nature of the underlying optimisation problem, particularly in the context of training deep learning models such as Transformers. While traditional gradient descent methods are effective for unconstrained optimisation, they often encounter difficulties in satisfying hard constraints, particularly in complex, non-convex landscapes.

$$A - M = 0 \tag{5.2}$$

where :

A is the predicted adjacency matrix $n \times n$ and

M is the target syntactic matrix $n \times n$.

n is the sentence tokens number.

The objective function is defined as $L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2$. This represents the squared Frobenius norm, which quantifies the discrepancy between the predicted and actual syntactic matrices. The Augmented Lagrangian introduces Lagrange multipliers λ and a penalty parameter μ to modify this loss function, yielding :

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top (\text{vec}(A) - \text{vec}(M)) + \frac{\mu}{2} \|\text{vec}(A) - \text{vec}(M)\|_F^2 \tag{5.3}$$

Where :

$L_{\text{task}}(A, M)$ is the previous defined objective function.

$operatornamevec()$ denotes the matrix vectorization obtained by stacking its columns into a single column vector (flattened vecto)

A, obtained by stacking its columns into a single column vector. λ are the Lagrange multipliers vector $n^2 \times 1$ that adjust dynamically to enforce the constraint.

μ is a positive scalar controlling the strength of the penalty term. It can be viewed as a form of regularization.

5.6.1 Loss Function

The prediction layer's output A is compared with the true adjacency matrix M which contains the actual syntactic dependencies using a task-specific loss function. The loss can be formulated as :

$$L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2 \tag{5.4}$$

Where : $\| \cdot \|_F^2$ is the Frobenius norm, which measures the difference between the predicted and true syntactic adjacency matrices.

5.6.2 Lagrange Multipliers

The term $\lambda^\top(A - M)$ plays crucial role in the enforcement of constraints during the optimisation process. In this context, the vector λ represents the Lagrange multipliers associated with the constraints defined in the optimisation problem. The constraints are that the learned matrix A should closely approximate the target adjacency matrix M, which encodes the syntactic dependencies between words. The denotation $\lambda^\top(A - M)$ represents the dot product between the vector λ and the matrix $A - M$. The λ vector is of length n dimension. Each entry of λ corresponds to a specific word in the sentence. This allows for individual weighting of the constraint violations associated with each word's syntactic dependencies. This configuration allows the model to ascertain the extent to which each word's representation should be modified in accordance with its relationship to other words within the sentence, thereby reflecting its significance within the context of the syntactic structure.

When λ is treated as importance weights of words, the model emphasizes the syntactic influence of each word on the overall structure. This aligns well with the goal of capturing linguistic dependencies, as the adjustments made by λ can reflect the importance of each word in maintaining syntactic relationships. The gradient updates influenced by λ can help shape the learning process, as the model adjusts the embeddings based on the weighted contributions of each word. This can lead to more effective embeddings that respect syntactic constraints more closely.

5.6.3 Constrained Learning with Penalization

The term $\frac{\mu}{2}\|A - M\|_F^2$ serves as a penalty that increases in severity when the predicted adjacency matrix A diverges from the target adjacency matrix M. This penalty discourages the model from making predictions that contravene the syntactic constraints, in a manner analogous to how regularisation techniques prevent overfitting by penalising complex models. The value of μ directly influences how strongly the constraints are enforced during training. The value of μ exerts a direct influence on the degree to which constraints are enforced during the training process. A larger μ places greater emphasis on satisfying the constraints, effectively guiding the optimisation process towards solutions that adhere closely to the required syntactic structure. This is analogous to a regularisation parameter in traditional regularisation methods such as $L2$ regularisation, where a larger value results in more stringent constraints on the model parameters.

5.6.4 Balancing Objective Function and Constraint Satisfaction

By adjusting μ, you can find a balance between minimizing the objective function $L_{\text{task}}(A, M)$ and ensuring that the predicted matrix A aligns with the constraints defined by M. In this way, μ serves a dual purpose : enhancing model performance on the primary task while also ensuring that the learned representations are constrained by the linguistic structure, similar to how regularization techniques aim to improve generalization.

5.6.5 Optimization

1. Loss Computing : at the start of each training iteration, compute the task loss

$$\frac{1}{2}\|A - M\|_F^2 \tag{5.5}$$

2. Constraint Violation Computing : determine the constraint violations function as

$$g(A) = A - M \tag{5.6}$$

3. Lagrange Multipliers Update : the Lagrange multipliers λ are updated to measure the current constraint violations

$$\lambda^{(k+1)} = \lambda^{(k)} + \mu \left(\text{vec}\left(A^{(k)}\right) - \text{vec}(M) \right) \tag{5.7}$$

   By applying the softmax function to the sum of the constraint violations, it effectively normalizes these constraint violations across the word embedding space.

4. Total Loss Computing : the total loss function is then expressed as

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top \left(\text{vec}(A) - \text{vec}(M)\right) + \frac{\mu}{2} \left\|\text{vec}(A) - \text{vec}(M)\right\|_F^2 \tag{5.8}$$

5. Total Gradient Computing : compute the gradient of the total loss with respect to A

$$\nabla_A L_A(A, \lambda, \mu) = \nabla_A L_{\text{task}}(A, M) + \nabla_A (\lambda^\top \left(\text{vec}(A) - \text{vec}(M)\right)) + \nabla_A \left(\mu \|A - M\|_F^2\right) \tag{5.9}$$

6. Gradient Descent Optimization : update A using the computed gradients

$$A \leftarrow A - \eta \nabla_A L(A, \lambda, \mu) \tag{5.10}$$

   where η is the learning rate, controlling how much A is updated in each iteration.

7. Backpropagation Computing : the gradients $\nabla_A L_A(A, \lambda, \mu)$ are computed based on the loss with respect to the output A. These gradients will indicate how changes in A affect the overall loss, providing information about how to adjust the weights in all encoder layers. Using the chain rule, the gradients of the loss with respect to the encoder weights can be calculated by tracing back through the layers of the model.

$$\nabla L_A = \nabla_A L_A + \nabla_H L_A \cdot W^T + \nabla_{W_q} L_A + \nabla_{W_k} L_A + \nabla_{W_v} L_A \tag{5.11}$$

Where : $\nabla_A L_A$ the gradient of the loss function with respect to the output matrix A.

$\nabla_H L_A$ is the gradient of the loss function with respect to the hidden states H.

W^T is the transposed weight matrix connecting H to the output matrix A.

∇W_q is the gradient of the loss L_A with respect to the weights W_q of the query projection in the self attention mechanism of the encoder.

∇W_k is the gradient of the loss L_A with respect to the weights W_k of the key projection in the self attention mechanism of the encoder.

∇W_v is the gradient of the loss L_A with respect to the weights W_v of the values projection in the self attention mechanism of the encoder.

5.7 Architecture

The proposed architecture consists of two interconnected components : the BERT Base and a Prediction Layer. The former is BERT Base follows the standard Transformer architecture, which operates without any constraints and leverages gradient descent optimization and the latter is the modified prediction layer that introduces a novel constraint-based optimization mechanism using Augmented Lagrangian Optimization.

5.8 Experiments

In order to evaluate and test the model, the same datasets that were used for BERT were employed : the English Wikipedia dump and BookCorpus. Following processing and cleaning, the Wikipedia dump yielded 16 GB of plain text, excluding non-textual elements such as tables and lists. In contrast, BookCorpus provides access to a substantial corpus of over 11,000 free, unpublished books sourced from the internet. To ensure a meaningful comparison with BERT and its derived models, we selected a robust hardware configuration. The training was conducted on a commercial cloud platform utilising 8 NVIDIA Tesla K80 GPUs. Preparing

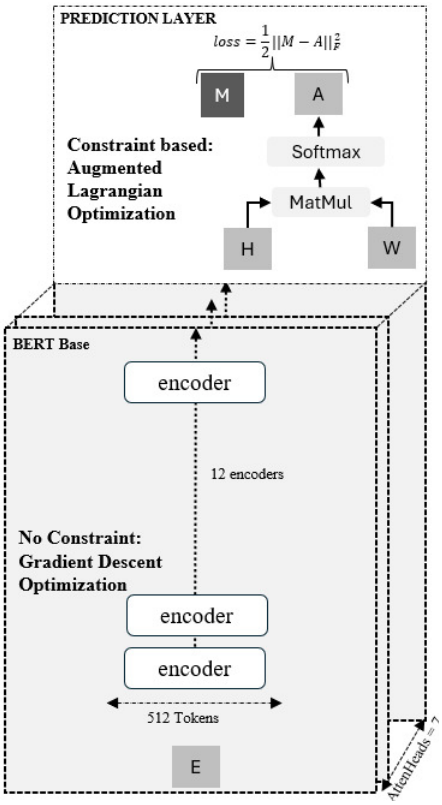

Figure 5.4 – Proposed Architecture

the training data required approximately 17 hours, in line with the specifications of these models. For model evaluation, we concentrated on text classification task. The AG News is used to focus on categorizing news articles into predefined categories to assess the performance and embeddings produced by our model.

Model	**Training Time**	**Dataset**
SCABERT	6 days	16 GB Wiki+BookCor
BERT	4 days	16 GB Wiki + BookCor

Table 5.1 – Comparison of various NLP models

Given that BERT was trained in approximately 4 days using 16 TPUs, while SCABERT took 6 days using only 8 GPUs, the longer training time for SCABERT is expected. This reflects the effeciency of the constraint based optimization with augmented lagrangian.

Metric	**BERT Base**	**SCABERT**
Precision (Class 0)	0.9539	- 0.9728
Recall (Class 0)	0.9584	- 0.9722
F1-Score (Class 0)	0.9562	- 0.9741
Precision (Class 1)	0.9884	- 0.9891
Recall (Class 1)	0.9879	- 0.9883
F1-Score (Class 1)	0.9882	- 0.9895
Precision (Class 2)	0.9251	- 0.9476
Recall (Class 2)	0.9095	- 0.9298
F1-Score (Class 2)	0.9172	- 0.9322
Precision (Class 3)	0.9127	- 0.9348
Recall (Class 3)	0.9242	- 0.9442
F1-Score (Class 3)	0.9184	- 0.9305
Accuracy	0.9450	- 0.9632

Table 5.2 – Comparison of SCABERT and BERT Base performance on AG News

The table presents a comparative analysis of performance metrics between SCABERT and BERT Base on the AG News dataset. The metrics evaluated include Precision, Recall, F1-Score, and Accuracy for four distinct classes. Overall, the results indicate that while BERT base performs admirably in certain metrics, especially

for Class 1, SCABERT generally outperforms it across all metrics and classes in this dataset.

5.9 Conclusion

In summary, SCABERT's key advantages over BERT stem from its innovative integration of syntactic dependencies as ground truth via adjacency matrices, which supervise the attention mechanism directly. This syntactically informed approach enhances contextual understanding and leads to faster convergence and improved training efficiency thanks to the combination of gradient descent and augmented Lagrange multipliers for constraint-based optimisation. By aligning word representations with syntactic structures, SCABERT achieves superior performance in NLP classification tasks, offering deeper linguistic insight and greater interpretability than traditional BERT models.

5.10 Perspective

While SCABERT is a significant milestone in leveraging syntactic structures as ground truth to supervise attention mechanisms, it remains rooted in the symbolic-linguistic domain. The innovation of SCABERT lies in its ability to align internal representations with syntactic dependencies, thereby improving the interpretability and efficiency of language understanding. However, meaning in human language extends beyond syntax. It is also rooted in perception, experience and multimodal grounding. To further push the boundaries of semantic representation, our proposed next model, VLG-BERT (Visually and Linguistically Grounded BERT), integrates syntactic and perceptual knowledge to enhance language modelling. Like SCABERT, VLG-BERT uses syntactic supervision, but it also incorporates visual latent representations obtained from pre-trained vision models. It achieves this by grounding a curated vocabulary of around 10,000 concrete tokens drawn from ImageNet labels and expanded through WordNet semantic relations. VLG-BERT introduces a lookup-based initialisation of embeddings informed by real-world perception rather than relying on random initialisation. This multimodal fusion provides a stronger foundation for meaning encoding and aims to make large language models more cognitively plausible. Not only does VLG-BERT capture linguistic structure, it also aligns its representations with how humans understand language in relation to the world they perceive. Its architecture reflects interdisciplinary insights from cognitive science, in which meaning is considered a construct shaped by language and sensory experience.

CHAPITRE 6

ANCRAGE DU LANGAGE ET DE LA VISION : LES VECTEURS VISUELS LATENTS COMME REPRÉSENTATION CONCEPTUELLE POUR UN ENCODAGE BIMODAL DU SENS DES MOTS

.

6.1 Détails de l'article

VLG-BERT : Towards Better Interpretability in LLMs through Visual and Linguistic Grounding

Toufik Mechouma, Ismail Biskri and Serge Robert

The 5th International Conference on Natural Language Processing for Digital Humanities–NLP4DH,ACL anthology,Albuquerque,USA,Mai,2025. (published)

6.2 Résumé

Nous présentons VLG-BERT, un nouveau modèle LLM conçu pour améliorer l'encodage du sens du langage. VLG-BERT fournit des informations plus approfondies sur l'encodage du sens dans les grands modèles de langage (LLM) en se concentrant sur la sémantique linguistique et la sémantique du monde réel. Il utilise les dépendances syntaxiques comme une forme de vérité de terrain pour superviser l'apprentissage de la représentation des mots. VLG-BERT intègre des représentations visuelles latentes à partir de modèles de vision pré-entraînés et de leurs étiquettes correspondantes. Un vocabulaire de 10 000 tokens correspondant à ce que l'on appelle des mots concrets est construit en étendant l'ensemble des étiquettes ImageNet. Cette extension est basée sur les synonymes, les hyponymes et les hypernymes de WordNet. Une table de recherche pour ce vocabulaire est donc utilisée pour initialiser la matrice d'intégration pendant l'apprentissage, plutôt qu'une initialisation aléatoire. Cette base multimodale permet d'établir une base sémantique plus solide pour l'encodage du sens des mots. Son architecture s'aligne parfaitement sur les théories fondamentales des sciences cognitives. L'intégration des bases visuelles et linguistiques rend VLG-BERT compatible avec de nombreuses théories cognitives. Notre approche participe à l'effort continu de création de modèles qui comblent l'écart entre le langage et la vision, et qui les rapprochent de la façon dont les êtres humains comprennent et interprètent le monde. Des expériences de classification de textes ont montré des résultats excellents par rapport à la base BERT.

6.3 Abstract

We present VLG-BERT, a novel LLM model conceived to improve the language meaning encoding. VLG-BERT provides a deeper insights about meaning encoding in Large Language Models (LLMs) by focusing on linguistic and real-world semantics. It uses syntactic dependencies as a form of a ground truth to supervise the learning process of the words representation. VLG-BERT incorporates visual latent representations from

pre-trained vision models and their corresponding labels. A vocabulary of 10k tokens corresponding to so-called concrete words is built by extending the set of ImageNet labels. The extension is based on synonyms, hyponyms and hypernyms from WordNet. Thus, a lookup table for this vocabulary is used to initialize the embedding matrix during training, rather than random initialization. This multimodal grounding provides a stronger semantic foundation for encoding the meaning of words. Its architecture aligns seamlessly with foundational theories from across the cognitive sciences. The integration of visual and linguistic grounding makes VLG-BERT consistent with many cognitive theories. Our approach contributes to the ongoing effort to create models that bridge the gap between language and vision, making them more aligned with how humans understand and interpret the world. Experiments on text classification have shown an excellent results compared to BERT Base.

6.4 Introduction

The growing need for interpretability and grounding in Large Language Models (LLMs) is driven by their increasing use in critical and diverse applications, as well as ethical, practical, and technical challenges. LLMs assist in diagnosing diseases and generating treatment plans. They are also used for contract analysis and legal reasoning. They personalize the learning experience for students. Despite their outstanding performance in many downstream tasks, LLMs often produce plausible but factually incorrect outputs, referred to as hallucination. This behavior results from their reliance on patterns in training data rather than true semantic understanding. LLMs must provide an explainable insights about their black-boxes. Their decisions must meet legal and ethical standards. Therefore, interpretability allows users to trace the reasoning or data sources behind a model's outputs, providing accountability. The integration of visual real-world data and domain knowledge into LLMs, could be good lead to anchor their responses to verifiable facts. The Text-based LLMs have made significant advancements in natural language processing. LLMs two fundamental learning policies are next-word generation and bidirectional representation. The first approach is used for text generation, by predicting the next word based on prior context. The second approach focuses on understanding text by predicting masked words using both left and right context. However, these models have notable limitations when it comes to representing meaning, particularly in relation to real-world semantics. While LLMs excel at capturing contextual relationships between words, they do not inherently ground meaning in the real world, unlike humans who learn language through sensory and perceptual experiences. In this paper, we introduce VLG-BERT, a multimodal model which combines syntactic knowledge and visual grounding to improve word representation learning. It extends our recent modal capabilities to incorporate real-world

semantics. Unlike traditional models that learn embeddings solely from textual space, VLG-BERT uses latent representations of real-world concepts to learn embeddings. Latent representations are extracted from the Vision Transformer (ViT) trained on the ImageNet dataset. VLG-BERT aims to go beyond the purely textual space as the only source of words representation learning, by involving the real-word semantics in the learning process. This grounding bridges the gap between vision and language, allowing the model to process and encode richer semantic information. It is also particularly useful for multimodal downstream tasks. VLG-BERT is also designed to inject syntactic knowledge into the attention mechanism using augmented Lagrange multipliers. The model employs syntactic dependencies as a form of ground truth to supervise the learning process of word representation, thereby ensuring that syntactic structure exerts an influence on the model's word representations. The application of augmented Lagrangian optimization impose constraints on the attention mechanism. It makes the learning of syntactic relationships easier. This approach involves the customization of prediction layer of the standard BERT architecture. The objective is to predict an adjacency matrix that encodes words' syntactic relationships rather than masked tokens. VLG-BERT introduces a merge between bottom-up or data driven approach and rules driven or a top-down approach. Furthermore, VLG-BERT brings clear insights about the interpretability of transformer-based models.

6.5 Related work

Transformer models like BERT and its variants have paved the way for great advancements in NLP. These models are primarily geared towards modeling the semantics of language. They've resulted in tremendous performance in many different fields Devlin *et al.* (2019); Liu *et al.* (2019); Lan *et al.* (2020); Sanh *et al.* (2020); He *et al.* (2021). The scientific community developed new versions of BERT as a consequence of the inaccurate results in some downstream tasks and appraisal of the linguistic properties of the natural language Htut *et al.* (2019); Wiegreffe et Pinter (2019); Clark *et al.* (2020). Some of the proposed models aim to inject linguistic knowledge into transformer models while others try to ground the language via visual data. Syntactic connections between words are not just what lends language its richness, but are also what make meaning beyond mere word correlations Mechouma *et al.* (2022b); Bai *et al.* (2021). One way of adding syntactic knowledge to transformer models is Syntax-BERT. It is an extension of the original BERT that introduces explicit syntactic information through syntax trees and instructs the self-attentional system in relation to linguistic dependencies such as parent, child, and sibling. This strategy preserves BERT's pre-trained expertise and combines it with structure and efficiency to help it better excel in NLP scenarios when syntactic clarity is required or data is finite. Syntax-BERT is a system that allows syntax trees to be

included during fine-tuning without the need to train from scratch Bai *et al.* (2021); Sundararaman *et al.* (2019). The Syntactic Knowledge via Graph Attention with BERT is another proposed model which adopts syntactic knowledge injection into transformer models. SGB is a machine translation dedicated model. It explicitly uses the syntactic dependency knowledge via Graph Attention Networks (GAT) and BERT-based encoders. The GAT treats syntactic structures as graphs, enhancing token representations with dependency relations. It also combines them with BERT outputs through two methods. The first one is called SGBC. it concatenates BERT and GAT outputs for encoder-decoder attention. The second one is SGBD (decoder-guided syntax). This approach leaverage a translation fluency Dai *et al.* (2023). In addition to the syntax-aware model in transformer models, a vision-oriented models have emerged. One of these models has been developed with the objective of grounding natural language in visual data is VisualBERT. It is based on the architecture of BERT. VisualBERT uses image-text alignment to ground language in visual contexts. It employs cross-attention layers to establish a connection between the visual and textual modalities. Visual information is conveyed through a convolutional neural network (CNN) to extract visual embeddings, which are subsequently integrated with the textual embeddings. The cross-modal attention layers grant bidirectional influence between text and image representations during the encoding process. VisualBERT employs a fusion strategy that unites textual tokens and visual features within a unified transformer Li *et al.* (2019). LXMERT, which stands for Learning Cross-Modality Encoder Representations from Transformers is a multimodal model. It processes both visual and textual data. It uses a cross-attention mechanism to merge the image and text features. LXMERT architecture is based on two-stream transformer. The first stream processes the visual features. It consists of image regions such as objects and objects parts encoded by a pretrained Faster R-CNN model. The encoded visual features are then fed into LXMERT to learn contextual relationships between image regions. The second stream processes textual features. It comprises BERT's word embeddings. Both streams interact with each other through Cross-Attention Encoder. This interaction enables the model to learn relationships between the image and its corresponding textual description Li *et al.* (2019). The list of multimodal models is longer enough to overpass the limited pages number of the present paper. Without dissecting technical details, we mention among others, UNITER, ImageBERT, and Multimodal-BERT, which are Transformer-based models. They are conceived to connect visual and textual data in order to improve the performance in multimodal tasks Rahman *et al.* (2020) Chen *et al.* (2020) Qi *et al.* (2020). UNITER, UNiversal Image-Text Representation learns joint embeddings by pre-training on diverse image-text datasets, enabling tasks like image-text retrieval and visual question answering Chen *et al.* (2020). Similarly, ImageBERT depends on a shared embedding space and cross-modal interaction to align text and images Qi *et al.* (2020). In turn, Multimodal-BERT customize BERT's architecture to handle multi-

modal inputs. It is particularly dedicated to applications like medical image and text classification Rahman *et al.* (2020). The research community is moving toward the integration of visual and textual data to encode the meaning of language. These models offer an excellent way of grounding the language by aligning visual information, such as images, with textual context. In the next sections, we present VLG-BERT, a multimodal model which combines syntactic knowledge and visual grounding to improve word representation learning.

6.6 Two Categories of Words

The present work assume two categories of words. The first is called concrete words, while the second is called abstract words. The former refers to all the words that they generally denote classes of entities perceived by the senses. The ladder refers to all words that do not have a physical referent in the real world. From cognitive sciences point of view, the term real world here, differs from Lackoff's definition Lakoff (1987). It is more in line with the definitions of Materialism and Empirical Realism.

6.7 Visual Grounding

Most LLMs use a random initialization to learn word embeddings. We propose a human-like model by initializing the embeddings matrix of words with their corresponding latent representation from the real-world. In other words, the visual grounding in VLG-BERT consists of using the latent representations extracted from the Vision Transformer ViT. The latent representations are learned by ViT based on the ImageNet dataset, which contains 1000 labels or classes corresponding to real objects Dosovitskiy *et al.* (2021b); Deng *et al.* (2009). We extend the vocabulary by building a lookup table that corresponds to our embeddings matrix, using wordnet. The vocabulary extension uses Synonymy, Hyponymy and Hypernymy relations Miller (1995). Semantically similar words are extended using WordNet semantic relations. Hyponyms are more specific terms, while Hypernyms are general terms or categories. The semantic similarity of hyponyms should be more similar to each other than to their hypernyms. This can be done by incorporating hierarchical WordNet semantic relations. In other words, several path-based similarity measures can be used to compute the shortest path between two words in the hypernym-hyponym tree. The shorter the path between the two words, the more semantically related they are. Finally the lookup table is implemented using JSON, where keys are the tokens IDs and values are the latent representations before and after regularization. The second category of words which have no referent in the real world, are randomly initialized as in traditional LLMs.

The metric that measures the relationship between a word w and its hyponym w_{hypo}, and its hypernym w_{hyper} is given by :

$$R(w, w_{\text{hyp}}, w_{\text{hyper}}) = \lambda \cdot \max\Big(0, \text{PathDist}(w, w_{\text{hyper}}) - \text{PathDist}(w, w_{\text{hypo}}) + \delta\Big). \tag{6.1}$$

where :

— λ is the regularization strength parameter, it controls the influence of the term.

— σ is a small margin to avoid zero and trivial solutions.

The intuition behind this regularization is to penalize the model when the path distance between a word w and its hypernym w_{hyper} is smaller than the path distance between the word and its hyponym w_{hypo}. Using the above metric, we compute hyponyms and hypernyms latent representations. Thus, we built a vocabulary of 10 000 concrete words. It take the form of a lookup table. It is used to initialize the embeddings, if the word is concrete and does not exist in the lookup table, we initialize it randomly.

6.8 Linguistic Grounding

VLG-BERT is a syntax-Aware model. It is designed to inject syntactic knowledge into the attention mechanism. It uses augmented Lagrange multipliers as a constraint based as convex optimization method. VLG-BERT deploys syntactic dependencies as a ground truth to supervise the learning process. The syntactic relations between the sentence words are encoded in an adjacency matrix. VLG-BERT is forced to predict a a matrix that approximate the adjacency matrix that encode the syntactic relations between words. The use of the augmented Lagrangian optimization method is an innovative way of integrating constraints in attention mechanisms. The prediction layer of the standard BERT architecture is customized to predict the syntactic matrix.

6.9 Conceptual Model

The model is based on transformer architectures and incorporates syntactic dependencies through the use of an adjacency matrix, M. M is used to encode the syntactic dependencies. During the training phase, it is employed as the ground truth to converge to. The positional encoding is kept as in BERT base, while the next sentence prediction is not integrated.

6.9.1 Input Layer

The input comprises word embeddings, represented as a matrix $E \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. The model takes both tokens and position embeddings as input to the Transformer layers.

6.9.2 Syntactic Dependencies Encoding

A binary adjacency matrix, $M \in \mathbb{R}^{n \times n}$, is incorporated into the model, to encode syntactic dependencies, where n is the number of words in a sentence. If word i has a direct dependency on word j, the corresponding entry in the matrix $\mathbf{M}$ is set to 1, indicating a dependency. Otherwise, the entry is set to 0. This matrix serves as a ground truth and a target for the model to learn during training.

6.9.3 Encoders Stack

The encoder stack is structured in accordance with the architectural principles of BERT Base. The encoder stack comprises a series of 12 Transformer layers, 12 attention heads, 768 hidden size, 512 maximum sentence length which perform attention-based learning over the input embeddings.

6.9.4 Prediction Layer

The input to the prediction layer is the output from the last encoder layer, denoted as matrix $H \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. To generate the syntactic dependency matrix A of shape $n \times n$, where n is the number of words in the input sentence. The model uses a fully connected (dense) layer that takes the encoded word representations H and maps them to an adjacency matrix representing the syntactic dependencies as follows.

$$A = \text{softmax}(H \cdot W) \tag{6.2}$$

Where : $H \in \mathbb{R}^{n \times d}$ is the output of the encoder stack.

$W \in \mathbb{R}^{d \times n}$ is a learnable weight matrix of the prediction layer.

$A \in \mathbb{R}^{n \times n}$ is the predicted syntactic adjacency matrix, representing the dependencies between the tokens in the input sequence. The output values $A_{ij} \in [0, 1]$ represent the strength of the syntactic dependency between the words i and j. A value close to 1 indicates a strong dependency, while a value close to 0 indicates weak or no dependency.

6.9.5 Why a Softmax and not a Sigmoid ?

In our context the question ties directly into the concepts of dependent and independent variables in the field of probability. From linguistic perspective, words are connected by syntactic dependencies, and these dependencies usually carry semantic meaning. By applying softmax, we introduce a distributional hypothesis where words with strong syntactic relationships have higher probabilities compared to unrelated words. In the case of sigmoid activation, we treat the syntactic relationships between words as independent events. In other words word-pairs are processed in isolation. From computational perspective, by introducing probability distribution, softmax squashes negative values towards zero and brings probabilities to one for relevant relationships, which is beneficial when used with the Lagrangian multiplier to converge quickly to a binary adjacency matrix. One potential downside of softmax is enforcing mutual exclusivity in its outputs, which could be problematic because a word can have multiple syntactic relationships simultaneously. In our case, softmax makes more sense than sigmoid.

6.9.6 Augmented Lagrangian Formulation

The Augmented Lagrangian method represents an extension of the classical Lagrangian approach to optimization, particularly suited for handling constraints in problems where traditional Lagrangian multipliers may be insufficient. In the present context, the Augmented Lagrangian framework is applied to enforce syntactic dependencies during the learning of word representations in a Transformer-based model. The mathematical foundation involves modifying the objective function by incorporating a penalty term to enforce the constraint.

The choice of the Augmented Lagrangian method is driven by the non-convex nature of the underlying optimization problem, particularly in the context of training deep learning models such as Transformers. While traditional gradient descent methods are effective for unconstrained optimization, they often encounter difficulties in satisfying hard constraints, particularly in complex, non-convex landscapes Fioretto *et al.* (2020); Basir et Senocak (2023); Wu *et al.* (2024).

$$A - M = 0 \tag{6.3}$$

where :

A is the predicted adjacency matrix and

M is the target syntactic matrix.

The objective function is defined as $L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2$. This represents the squared Frobenius norm, which quantifies the discrepancy between the predicted and actual syntactic matrices. The Augmented Lagrangian introduces Lagrange multipliers λ and a penalty parameter μ to modify this loss function, yielding :

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top (\text{vec}(A) - \text{vec}(M)) + \frac{\mu}{2} \|\text{vec}(A) - \text{vec}(M)\|_F^2 \tag{6.4}$$

Where :

$L_{\text{task}}(A, M)$ is the previous defined objective function.

λ are the Lagrange multipliers vector $n^2 \times 1$ that adjust dynamically to enforce the constraint.

$operatornamevec()$ denotes the matrix vectorization obtained by stacking its columns into a single column vector (flattened vecto)

μ is a positive scalar controlling the strength of the penalty term. It can be viewed as a form of regularization.

6.9.7 Loss Function

The prediction layer's output A is compared with the true adjacency matrix M which contains the actual syntactic dependencies using a task-specific loss function. The loss can be formulated as :

$$L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2 \tag{6.5}$$

Where : $\| \cdot \|_F^2$ is the Frobenius norm, which measures the difference between the predicted and true syntactic adjacency matrices.

6.9.8 Lagrange Multipliers

The term $\lambda^\top(A - M)$ plays crucial role in the enforcement of constraints during the optimization process. In this context, the vector λ represents the Lagrange multipliers associated with the constraints defined in the optimization problem. The constraints are that the learned matrix A should closely approximate the target adjacency matrix M, which encodes the syntactic dependencies between words. The denotation $\lambda^\top(A - M)$ represents the dot product between the vector λ and the matrix $A - M$. The λ vector is of length n dimension. Each entry of λ corresponds to a specific word in the sentence. This allows for individual weighting of the constraint violations associated with each word's syntactic dependencies. This configuration allows the model to ascertain the extent to which each word's representation should be modified in accordance with its relationship to other words within the sentence, thereby reflecting its significance within the context of the syntactic structure.

When λ is treated as importance weights of words, the model emphasizes the syntactic influence of each word on the overall structure. This aligns well with the goal of capturing linguistic dependencies, as the adjustments made by λ can reflect the importance of each word in maintaining syntactic relationships. The gradient updates influenced by λ can help shape the learning process, as the model adjusts the embeddings based on the weighted contributions of each word. This can lead to more effective embeddings that respect syntactic constraints more closely.

6.9.9 Constrained Learning with Penalization

The term $\frac{\mu}{2}\|A - M\|_F^2$ serves as a penalty that increases in severity when the predicted adjacency matrix A diverges from the target adjacency matrix M. This penalty discourages the model from making predictions that contravene the syntactic constraints, in a manner analogous to how regularisation techniques prevent overfitting by penalising complex models. The value of μ directly influences how strongly the constraints are enforced during training. The value of μ exerts a direct influence on the degree to which constraints are enforced during the training process. A larger μ places greater emphasis on satisfying the constraints, effectively guiding the optimisation process towards solutions that adhere closely to the required syntactic structure. This is analogous to a regularisation parameter in traditional regularisation methods such as $L2$ regularisation, where a larger value results in more stringent constraints on the model parameters.

6.9.10 Balancing Objective Function and Constraint Satisfaction

By adjusting μ, you can find a balance between minimizing the objective function $L_{\text{task}}(A, M)$ and ensuring that the predicted matrix A aligns with the constraints defined by M. In this way, μ serves a dual purpose : enhancing model performance on the primary task while also ensuring that the learned representations are constrained by the linguistic structure, similar to how regularization techniques aim to improve generalization.

6.9.11 Optimization

1. Loss Computing : at the start of each training iteration, compute the task loss

$$\frac{1}{2}\|A - M\|_F^2 \tag{6.6}$$

2. Constraint Violation Computing : determine the constraint violations function as

$$g(A) = A - M \tag{6.7}$$

3. Lagrange Multipliers Update : the Lagrange multipliers λ are updated to measure the current constraint violations

$$\lambda^{(k+1)} = \lambda^{(k)} + \mu \left(\text{vec}\left(A^{(k)}\right) - \text{vec}(M) \right) \tag{6.8}$$

By applying the softmax function to the sum of the constraint violations, it effectively normalizes these constraint violations across the word embedding space.

4. Total Loss Computing : the total loss function is then expressed as

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top (\text{vec}(A) - \text{vec}(M)) + \frac{\mu}{2} \|\text{vec}(A) - \text{vec}(M)\|_F^2 \tag{6.9}$$

5. Total Gradient Computing : compute the gradient of the total loss with respect to A

$$\nabla_A L_A(A, \lambda, \mu) = \nabla_A L_{\text{task}}(A, M) + \nabla_A(\lambda^\top (\text{vec}(A) - \text{vec}(M))) + \nabla_A \left(\mu \|A - M\|_F^2\right) \tag{6.10}$$

6. Gradient Descent Optimization : update A using the computed gradients

$$A \leftarrow A - \eta \nabla_A L(A, \lambda, \mu) \tag{6.11}$$

where η is the learning rate, controlling how much A is updated in each iteration.

7. Backpropagation Computing : the gradients $\nabla_A L_A(A, \lambda, \mu)$ are computed based on the loss with respect to the output A. These gradients will indicate how changes in A affect the overall loss, providing information about how to adjust the weights in all encoder layers. Using the chain rule, the gradients of the loss with respect to the encoder weights can be calculated by tracing back through the layers of the model.

$$\nabla L_A = \nabla_A L_A + \nabla_H L_A \cdot W^T + \nabla_{W_q} L_A + \nabla_{W_k} L_A + \nabla_{W_v} L_A \tag{6.12}$$

Where : $\nabla_A L_A$ the gradient of the loss function with respect to the output matrix A.

$\nabla_H L_A$ is the gradient of the loss function with respect to the hidden states H.

W^T is the transposed weight matrix connecting H to the output matrix A.

∇W_q is the gradient of the loss L_A with respect to the weights W_q of the query projection in the self attention mechanism of the encoder.

∇W_k is the gradient of the loss L_A with respect to the weights W_k of the key projection in the self attention mechanism of the encoder.

∇W_v is the gradient of the loss L_A with respect to the weights W_v of the values projection in the self attention mechanism of the encoder.

6.10 VLG-BERT under the Spotlight of Cognitive Sciences

LLMs learn the probability distribution of sequences of words in natural language. They are designed based on the idea of maximizing the probability of certain words under certain conditions. This can be the next word in a sequence, or a masked word. In an auto-regressive model, given a sequence of words $w_1, w_2, \ldots, w_{n-1}$, the model learns to predict the probability distribution for the next word w_n. Unlike the auto-regressive model, bidirectional models learn to predict a word by conditioning on both the preceding and succeeding words in the sequence. Given a sequence of words $w_1, w_2, \ldots, w_n$, the model predicts a representation for each word by conditioning on both the left and right context. The LLMs community considers next word prediction models to be text generation models, while they consider bidirectional encoding models to be text understanding models. The integration of different sensory modalities is necessar to humans to perceive and understand the world. The architecture of VLG-BERT can be seen as a computational model that mimics humans by combining textual and visual data for a better and deeper encoding of the language meaning. VLG-BERT aligns with many theories like Symbol Grounding. Symbol Grounding refers to the association of the abstract symbols like words with real-world objects Harnad (1990). In cognitive science, grounding is fundamental to how humans link linguistic symbols to sensory experiences like seeing an apple. In Embodied Cognition theory, the mind is considered to be rooted in the body's interactions with the world. This implies that understanding comes from both perceiving and acting in the world. VLG-BERT aligns with the idea of Embodied Cognition by grounding language in visual data Barsalou (1999). The representations in VLG-BERT approximate Rosch Prototypes theory by clustering features from both latent visual features and linguistic domains, improving generalization for concept categories Rosch et Heider (1973). VLG-BERT aligns with Dual Coding theory that combines verbal and imaginal codes that reinforce the comprehension and the retrieval of concrete concepts Evans et Frankish (2009). By combining visual signs and linguistic signs, VLG-BERT aligns with Peirce's triadic model of signification, offering a robust semiotic framework for word meaning Peirce (1878, 1958). The visual and linguistic signs can be considered as iconic and symbolic representamens while the learned embeddings of words like Interpretants.

6.11 Architecture

The proposed architecture consists of two interconnected components : The BERT Base and a customized prediction Layer. The former is BERT Base follows the standard Transformer architecture, which operates without any constraints and leverages gradient descent optimization and the latter is the modified prediction layer that introduces a novel constraint-based optimization mechanism using Augmented Lagrangian

Optimization. At the input layer, lookup table is used to map visual latent representation to corresponding tokens of the sentence to i

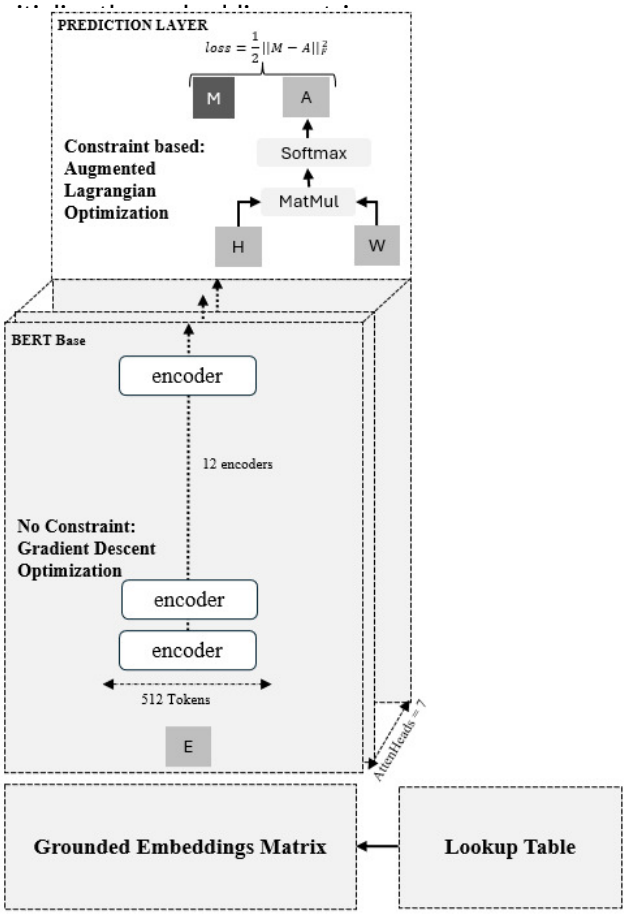

Figure 6.1 – Proposed Architecture

6.12 Experiments

In order to evaluate and test VLG-BERT, the same datasets already used by BERT were employed : the English Wikipedia dump and BookCorpus. The Wikipedia dump yielded 16 GB of plain text. In turn, BookCorpus provides access to a substantial corpus of over 11,000 free, unpublished books sourced from the internet. To ensure a meaningful comparison with BERT and its derived models, we used a high performance hardware configuration. The training was conducted on a commercial cloud platform utilizing 8 GPUs, 128 Gig of RAM and 32 of vCPUs Cores. For model evaluation, we concentrated on text classification task. To evaluate the generated embedding from VLG-BERT, the AG News dataset is used to focus on categorizing news articles into predefined categories. Hyper-parameters are defined as following λ for equation 1 is 0.01,μ for equation 4 is 0.001, **Learning Rate :** 2×10^{-5}, **Train Batch Size :** 16, **Evaluation Batch Size :** 8, **Seed :** 42, **Optimizer :** Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$, **Number of Epochs :** 30. While BERT-base took around

96 hours to train on 16 TPUs, we notice that VLG-BERT, on the other hand, took a longer training time of 122 hours. This is expected because the hardware configuration in that case was less powerful than that of BERT-base. That would reflect the efficiency of the learned embeddings with VLG-BERT. This confirms that the model converged quickly. This proves the efficiency of the visual grounding and also the use of constraint-based optimization with an augmented Lagrangian to reduce the training time.

Metric	**BERT Base**	**VLG-BERT**
Precision (Class 0)	0.9539	0.9815
Recall (Class 0)	0.9584	0.9833
F1-Score (Class 0)	0.9562	0.9784
Precision (Class 1)	0.9884	0.9903
Recall (Class 1)	0.9879	0.9901
F1-Score (Class 1)	0.9882	0.9912
Precision (Class 2)	0.9251	0.9602
Recall (Class 2)	0.9095	0.9513
F1-Score (Class 2)	0.9172	0.9526
Precision (Class 3)	0.9127	0.9482
Recall (Class 3)	0.9242	0.9458
F1-Score (Class 3)	0.9184	0.9437
Accuracy	0.9450	0.9756

Table 6.1 – Performance of the three model on AGNews Dataset

The comparison of the two models on the AGNews dataset shows that VLG-BERT outperforms BERT Base in all metrics. VLG-BERT scored the highest accuracy (97.56%) and F1-Scores for all classes. It demonstrates notable improvements in precision, recall, and F1-Scores. Compared to SCABERT, which benefits from only syntactic grounding.

6.13 Conclusion

VLG-BERT has valuable contributions from both computer science and cognitive science standpoints. Computer science, with regard to the advance of multimodal learning, it efficiently combines visual and linguistic

data that could lead to richer, more robust representations of words. The integration of visual grounding with textual information enables this model to handle complex, real-world tasks more efficiently. Such a setup from a cognitive science viewpoint is in consonance with VLG-BERT, as it grounds the words in the physical world, incorporating syntactic structures to mirror computationally human-like understanding of concepts. The model supports the perceptual gap between language and vision, representing and leveraging visual and linguistic inputs cohesively to interpret the world, much like humans. This will be further demonstrated by future comparisons with models like VisualBERT, LXMERT, and CLIP, especially on multimodal tasks such as image captioning and visual question answering. These will serve to underline its ability to integrate visual, syntactic, and semantic knowledge to provide a deeper understanding of multimodal interactions.

CONCLUSION

Dans cette thèse, nous avons exploré de nouvelles approches pour intégrer des connaissances linguistiques et visuelles aux mécanismes d'attention, afin d'améliorer l'encodage du sens des mots et de rendre les grands modèles de langage plus lisibles. Nos travaux visent à dépasser les limites des modèles existants en intégrant des structures syntaxiques et des représentations visuelles latentes, afin d'aligner l'apprentissage automatique sur les principes fondamentaux des sciences cognitives. Nous nous sommes notamment appuyés sur la théorie sémiotique de Charles Sanders Peirce, qui fournit un cadre solide pour appréhender le sens à travers les symboles, les indices et les icônes, applicables aussi bien au langage qu'à la perception visuelle.

Nous avons tout d'abord introduit un masque d'analyse des dépendances (DPM) qui améliore le mécanisme d'attention de BERT en exploitant les relations syntaxiques entre les mots. Cette approche s'appuie sur la catégorie du symbole chez Peirce : le sens naît ici de conventions grammaticales formelles, dans lesquelles la signification d'un mot dépend de sa position et de sa relation aux autres dans la structure syntaxique.

Dans un second temps, nous avons conçu lingBERT, une variante de BERT qui intègre une stratégie de masquage hybride combinant aléatoirement des tokens avec des mots ayant des relations syntaxiques. Cette approche permet d'améliorer la capture du contexte tout en réduisant la complexité computationnelle. En s'appuyant sur les arbres syntaxiques comme vérité de terrain, lingBERT inscrit son encodage sémantique dans un cadre symbolique tout en amorçant une ouverture vers une supervision plus formelle du sens, en traitant les structures syntaxiques comme des objets de connaissance à injecter dans le modèle.

Pour poursuivre cette exploration, nous avons développé SCABERT, une méthode d'optimisation par multiplicateurs de Lagrange augmentés qui contraint l'apprentissage des représentations lexicales en fonction des dépendances syntaxiques. Ici, la structure syntaxique devient une vérité de supervision. Cette approche renforce la catégorie du symbole dans le modèle et commence également à refléter une forme d'indice : les dépendances syntaxiques ne sont plus seulement des conventions, mais deviennent des contraintes indiquant la manière dont les mots interagissent dans la dynamique de la phrase.

Enfin, nous avons proposé VLG-BERT, un modèle multimodal combinant des connaissances linguistiques et visuelles afin d'enrichir la sémantique des représentations lexicales. Il s'agit de l'évolution directe de SCA-

BERT. En s'appuyant sur un vocabulaire structuré à partir des étiquettes ImageNet et des relations lexicales de WordNet, VLG-BERT introduit une base iconique dans l'encodage du sens. À travers les représentations visuelles latentes, le modèle exploite la ressemblance entre les objets perçus et leur signifiant, conformément à la notion d'icône chez Peirce. De plus, le lien entre les mots et leurs représentations visuelles établit une relation de contiguïté perceptive, propre à la notion d'indice. VLG-BERT est ainsi le premier modèle de notre lignée à incarner pleinement la triade sémiotique peircéenne. :

— les symboles, avec la structure linguistique et syntaxique ;
— les indices, via les associations perceptives et catégorielles ;
— les icônes, à travers la ressemblance visuelle.

Les résultats obtenus grâce à ces différentes contributions confirment que l'intégration de connaissances linguistiques et visuelles dans les mécanismes d'attention constitue une avancée significative pour l'encodage du sens des mots. Nos approches ont non seulement permis d'améliorer les performances des modèles sur des tâches en aval, mais aussi de renforcer leur explicabilité et leur alignement avec les structures linguistiques et cognitives sous-jacentes.

Dans la continuité de ces travaux, plusieurs perspectives peuvent être envisagées. D'une part, l'extension de ces approches à des modèles plus larges et plus complexes permettrait d'explorer les limites de l'injection de connaissances structurées à grande échelle. D'autre part, une validation approfondie sur des tâches multimodales, telles que la génération d'images à partir de descriptions textuelles ou la compréhension de dialogues visuels, permettrait d'évaluer plus précisément l'apport des représentations intégrant la syntaxe et la vision. Enfin, il serait intéressant de combiner ces approches à des méthodes d'apprentissage contrastif afin de renforcer la cohérence entre les différentes modalités et d'affiner l'encodage du sens des mots dans les LLM.

Cette thèse ouvre ainsi de nouvelles perspectives pour l'amélioration des modèles de langage, en s'appuyant sur une fusion plus riche entre les connaissances symboliques, les représentations visuelles et les mécanismes d'attention. En nous appuyant sur la sémiotique de Peirce, nous avons montré qu'il est possible de construire des modèles plus interprétables, plus efficaces et plus proches du fonctionnement cognitif humain en reliant le langage non seulement à sa forme grammaticale, mais aussi à son ancrage perceptif et conceptuel dans le monde.

BIBLIOGRAPHIE

Ainslie, J., Ontanon, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q. et Yang, L. (2020). Etc : Encoding long and structured inputs in transformers. 268–284. http://dx.doi.org/10.18653/v1/2020.emnlp-main.19

Aitchison, J. (1987). *Words in the Mind : An Introduction to the Mental Lexicon*. Blackwell.

Alex, W., Yu, C., Ioana, G., Wei, P., Hagen, B., Yining, N., Anna, A., Shikha, B., Haokun, L., Alicia, P., Sheng-Fu, W., Jason, P., Anhad, M., Phu Mon, H., Paloma, J. et Samuel, R. B. (2019). Investigating bert's knowledge of language : Five analysis methods with npis. Dans *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2877–2887., Hong Kong, China. Association for Computational Linguistics.

Antoun, W., Baly, F. et Hajj, H. (2020). Arabert : Transformer-based model for arabic language understanding.

Armstrong, D. M. (1989). *Universals : An Opinionated Introduction*. Boulder, Colorado : Westview Press.

Arras, L., Horn, F., Montavon, G., Müller, K.-R. et Samek, W. (2017). "what is relevant in a text document ?" : An interpretable machine learning approach. *PLOS ONE*, *12*(8), e0181142. http://dx.doi.org/10.1371/journal.pone.0181142. Récupéré de http://dx.doi.org/10.1371/journal.pone.0181142

Arrieta, A. B., Díaz-Rodríguez, N., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R. et Herrera, F. (2019). Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. Récupéré de https://arxiv.org/abs/1910.10045

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. et Ives, Z. (2007). Dbpedia : A nucleus for a web of open data. Dans C. Bizer, R. Heese, et H. Stuckenschmidt (dir.). *Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*, 722–735. Springer. http://dx.doi.org/10.1007/978-3-540-76298-0_52. Récupéré de http://www.springerlink.com/index/x725j37762416v72.pdf

Bahdanau, D., Cho, K. et Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate. Récupéré de https://arxiv.org/abs/1409.0473

Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J. et Tong, Y. (2021). Syntax-bert : Improving pre-trained transformers with syntax trees. *CoRR*, *abs/2103.04350*. Récupéré de https://arxiv.org/abs/2103.04350

Bar-Hillel, Y. (1964). Linguistics and machine translation. *Language and Information*, *3*, 19–27.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*(4), 577–660. http://dx.doi.org/10.1017/S0140525X99002149

Barsalou, L. W., Santos, A., Simmons, W. K. et Wilson, C. D. (2008). Language and simulation in conceptual processing. In M. De Vega, A. M. Glenberg, et A. C. Graesser (dir.), *Symbols*, *Embodiment*, *and Meaning* 245–284. Oxford, UK : Oxford University Press.

Barthes, R. (1972). *Mythologies*. Hill and Wang. Original work published 1957.

Basir, S. et Senocak, I. (2023). An adaptive augmented lagrangian method for training physics and equality constrained artificial neural networks. Récupéré de https://arxiv.org/abs/2306.04904

Beltagy, I. et Cohn, T. (2020). Scibert : A pretrained language model for scientific text. Dans *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3615–3620. Récupéré de https://aclanthology.org/D19-1371

Beltagy, I., Peters, M. et Cohan, A. (2020). Longformer : The long-document transformer. http://dx.doi.org/10.48550/arXiv.2004.05150

Bender, E. et Koller, A. (2020). Climbing towards nlu : On meaning, form, and understanding in the age of data. 5185–5198. http://dx.doi.org/10.18653/v1/2020.acl-main.463

Bengio, Y., Ducharme, R., Vincent, P. et Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Berners-Lee, T., Hendler, J. et Lassila, O. (2001). The semantic web. *Scientific American Magazine*.

Blei, D. M., Ng, A. Y. et Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bojanowski, P., Grave, E., Mikolov, T. et Joulin, A. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Brown, T. *et al*. (2020a). Language models are few-shot learners. *arXiv preprint arXiv :2005.14165*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C. et Amodei, D. (2020b). Language models are few-shot learners. http://dx.doi.org/10.48550/arXiv.2005.14165

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. et Amodei, D. (2020c). Language models are few-shot learners. Dans *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.

Carnap, R. (1947). *Meaning and Necessity : A Study in Semantics and Modal Logic*. Chicago : University of Chicago Press.

Carnap, R., Hahn, H. et Neurath, O. (1929). The scientific conception of the world : The vienna circle.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. et Mukhopadhyay, D. (2018). Adversarial attacks and defences : A survey. Récupéré de https://arxiv.org/abs/1810.00069

Chen, C.-F., Fan, Q. et Panda, R. (2021). Crossvit : Cross-attention multi-scale vision transformer for image classification. Récupéré de https://arxiv.org/abs/2103.14899

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y. et Liu, J. (2020). Uniter : Universal image-text representation learning. Récupéré de https://arxiv.org/abs/1909.11740

Child, R., Gray, S., Radford, A. et Sutskever, I. (2019). Generating long sequences with sparse transformers. http://dx.doi.org/10.48550/arXiv.1904.10509

Chilton, P. (2013). Frames of reference and the linguistic conceptualization of time. In *Frames of Reference*. Oxford University Press

Chomsky, N. (2000). *New Horizons in the Study of Language and Mind*. Cambridge University Press.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L. et Weller, A. (2020). Rethinking attention with performers. http://dx.doi.org/10.48550/arXiv.2009.14794

Clark, K., Khandelwal, U., Levy, O. et Manning, C. (2019). What does bert look at ? an analysis of bert's attention. Dans *Proceedings of the 2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, 276–286. Association for Computational Linguistics. Récupéré de https://aclanthology.org/W19-4902

Clark, K., Luong, M. T., Le, Q. V. et Manning, C. D. (2020). Electra : Pre-training text encoders as discriminators rather than generators. Dans *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia. International Conference on Learning Representations (ICLR).

Cross, R. (2005). Duns scotus on universals. *Archiv für Geschichte der Philosophie*, *87*(3), 214–240. http://dx.doi.org/10.1515/AGPH.2005.87.3.214

Dai, Y., Sharoff, S. et de Kamps, M. (2023). Syntactic knowledge via graph attention with bert in machine translation. Récupéré de https://arxiv.org/abs/2305.13413

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V. et Salakhutdinov, R. (2019). Transformer-xl : Attentive language models beyond a fixed-length context. Récupéré de https://arxiv.org/abs/1901.02860

Dalvi, F., Nortonsmith, A., Bau, D. A., Belinkov, Y., Sajjad, H., Durrani, N. et Glass, J. (2018). Neurox : A toolkit for analyzing individual neurons in neural networks. Récupéré de https://arxiv.org/abs/1812.09359

Damasio, A. R. et Damasio, H. (1994). Cortical systems for retrieval of concrete knowledge : the convergence zone framework. In C. Koch et J. L. Davis (dir.), *Large-Scale Neuronal Theories of the Brain. Computational neuroscience* 61–74. Cambridge, MA : MIT Press.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. et Fei-Fei, L. (2009). Imagenet : A large-scale hierarchical image database. Dans *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848

Devlin, J., Chang, M.-W., Lee, K. et Toutanova, K. (2019). Bert pre-training of deep bidirectional transformers for language understanding. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, 4171–4186.

Dewey, J. (1916). *Democracy and Education*. Macmillan.

Dewey, J. (1938). *Logic : The Theory of Inquiry*. Henry Holt and Company.

Doshi-Velez, F. et Kim, B. (2017). Towards a rigorous science of interpretable machine learning. Récupéré de https://arxiv.org/abs/1702.08608

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. et Houlsby, N. (2021a). An image is worth 16x16 words : Transformers for image recognition at scale. Récupéré de https://arxiv.org/abs/2010.11929

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. et Houlsby, N. (2021b). An image is worth 16x16 words : Transformers for image recognition at scale. Récupéré de https://arxiv.org/abs/2010.11929

Evans, J. S. B. T. et Frankish, K. (2009). Dual-process theories of reasoning : Contemporary issues and developmental applications. In J. S. B. T. Evans et K. Frankish (dir.), *In Two Minds : Dual Processes and Beyond*. Oxford University Press.

Fillmore, C. J., Baker, C. F., Seneff, S. et Jurafsky, D. (2006). The role of framenet in the representation of semantic knowledge. Dans *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 1–11., Genoa, Italy. Récupéré de http://www.lrec-conf.org/proceedings/lrec2006/pdf/160_pdf.pdf

Fioretto, F., Hentenryck, P. V., Mak, T. W., Tran, C., Baldo, F. et Lombardi, M. (2020). Lagrangian duality for constrained deep learning. Récupéré de https://arxiv.org/abs/2001.09394

Firth, J. R. (1957). *Papers in Linguistic Methodology*. Oxford University Press.

Fodor, J. A. (1975). *The Language of Thought*. Crowell.

Fodor, J. A. (1998). *Concepts : Where Cognitive Science Went Wrong*. New York : Oxford University Press.

Frege, G. (1892). *Über Sinn und Bedeutung*. Leipzig : Zeitschrift für Philosophie und philosophische Kritik.

Gheini, M., Ren, X. et May, J. (2021). Cross-attention is all you need : Adapting pretrained transformers for machine translation. Récupéré de https://arxiv.org/abs/2104.08771

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. et Kagal, L. (2019). Explaining explanations : An overview of interpretability of machine learning. Récupéré de https://arxiv.org/abs/1806.00069

Goodman, N. (1954). Fact, fiction, and forecast.

Graves, A. (2012). Long short-term memory. In *Studies in Neural Networks*. Springer

Green, C. (1969). Application of theorem proving to problem solving. Dans *Proceedings of the 1st International Joint Conference on Artificial Intelligence (IJCAI)*, 219–240. Morgan Kaufmann.

Gruber, T. (1993). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, *43*(5–6), 907–928.

Gruber, T. (2009). Ontology. In Springer-Verlag (dir.), *Encyclopedia of Database Systems*. Springer-Verlag.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D. et Giannotti, F. (2018). A survey of methods for explaining black box models. Récupéré de https://arxiv.org/abs/1802.01933

Gärdenfors, P. (2000). *Conceptual Spaces : The Geometry of Thought*. Cambridge, Massachusetts : MIT Press.

Gärdenfors, P. (2019). From sensations to concepts : a proposal for two learning processes. *Review of Philosophy and Psychology*, *10*, 441–464. http://dx.doi.org/10.1007/s13164-017-0379-7

H, S. et S, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735

Hamdan, M., Rahiche, A. et Cheriet, M. (2024). Hand : Hierarchical attention network for multi-scale handwritten document recognition and layout analysis. http://dx.doi.org/10.48550/arXiv.2412.18981

Hao, Y., Yu, H. et You, J. (2025). Beyond facts : Evaluating intent hallucination in large language models. Récupéré de https://arxiv.org/abs/2506.06539

Harnad, S. (1990). The symbol grounding problem. *Physica D : Nonlinear Phenomena*, *42*(1-3), 335–346. http://dx.doi.org/10.1016/0167-2789(90)90232-T

Harris, Z. S. (1954). *Distributional structure*, volume 10.

He, P., Liu, X., Gao, J. et Chen, W. (2021). Deberta : Decoding-enhanced bert with disentangled attention. Dans *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Récupéré de https://aclanthology.org/2021.emnlp-main.26/

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B. et Darrell, T. (2016). Generating visual explanations. Récupéré de https://arxiv.org/abs/1603.08507

Hintikka, J. (1962). *Knowledge and Belief : An Introduction to the Logic of the Two Notions*. Ithaca : Cornell University Press.

Hinton, G., Vinyals, O. et Dean, J. (2015). Distilling the knowledge in a neural network. Récupéré de https://arxiv.org/abs/1503.02531

Holtzman, A., Buys, J., Du, J. et Oguz, B. (2020). The elephant in the room : Evaluating the long-term dependencies in generative transformers. Dans *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Honnibal, M. et Montani, I. (2017). spacy 2 : Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. Dans *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, 1116–1121. Association for Computational Linguistics. Récupéré de https://aclanthology.org/D17-1162

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L. et Weischedel, R. (2006). Ontonotes : The 90% solution.

Htut, P. M., Phang, J., Bordia, S. et Bowman, S. R. (2019). Do attention heads in bert track syntactic dependencies ? Récupéré de https://arxiv.org/abs/1911.12246

Jackendoff, R. (1983). *Semantics and Cognition*. MIT Press.

Jain, S. et Wallace, B. C. (2019). Attention is not explanation. Dans *Proceedings of the 2019 ACL Workshop on Interpretable NLP*, 54–60.

James, P. (1992). Knowledge graphs. In R. P. V. de Riet (dir.), *Linguistic Instruments in Knowledge Engineering* p. 98. Elsevier Science Publishers.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A. et Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12), 1–38. http://dx.doi.org/10.1145/3571730. Récupéré de http://dx.doi.org/10.1145/3571730

Jiao, X., Yang, Y., Liu, Y., Du, J., Han, J. et Lin, J. (2020). Tinybert : Distilling bert for natural language understanding. Dans *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Récupéré de https://arxiv.org/abs/2003.07874

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1), 11–21.

Jordan, M., Valentin, M., Versace, R. et Vallet, G. T. (2021). Les « sens » de la mémoire. *Intellectica*, *74*, 185–209. http://dx.doi.org/10.3406/intel.2021.1990

Joshi, M., Chen, E., Levy, O., Zettlemoyer, L. et Gardent, C. (2020). Spanbert : Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, *8*, 64–77. Récupéré de https://www.aclweb.org/anthology/2020.tacl-1.6/

Kanakarajan, K. R., Kundumani, B. et Sankarasubbu, M. (2021). Bioelectra :pretrained biomedical text encoder using discriminators. 143–154. http://dx.doi.org/10.18653/v1/2021.bionlp-1.16

Katharopoulos, A., Vyas, A., Pappas, N. et Fleuret, F. (2020). Transformers are rnns : Fast autoregressive transformers with linear attention. Récupéré de https://arxiv.org/abs/2006.16236

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C. et Socher, R. (2019). Ctrl : A conditional transformer language model for controllable generation. Récupéré de https://arxiv.org/abs/1909.05858

Klyne, G. et Carroll, J. (2004). W3c recommendation. World Wide Web Consortium (W3C). Editors.

Kovaleva, O., Romanov, A., Rogers, A. et Rumshisky, A. (2019). Revealing the dark secrets of BERT. Dans K. Inui, J. Jiang, V. Ng, et X. Wan (dir.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4365–4374., Hong Kong, China. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/D19-1445. Récupéré de https://aclanthology.org/D19-1445/

Lakoff, G. (1987). *Women, Fire, and Dangerous Things : What Categories Reveal About the Mind*. University of Chicago Press.

Lamb, S. M. (1966). Outline of stratificational grammar. *Washington, D.C. : Georgetown University Monograph Series on Language and Linguistics*, *18*, 83–104.

Lan, Z., Chen, J., Goodman, S., Gimpel, K. et Sharma, P. (2020). Albert : A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv :1909.11942*. Récupéré de https://arxiv.org/abs/1909.11942

Landauer, T. K. et Dumais, S. T. (1997). A solution to plato's problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*(2), 211–240. http://dx.doi.org/10.1037/0033-295x.104.2.211

Lauly, S., Zheng, Y., Allauzen, A. et Larochelle, H. (2016). Document neural autoregressive distribution estimation. *Journal of Machine Learning Research*, *18*. http://dx.doi.org/10.48550/arXiv.1603.05962

Le, Q. et Mikolov, T. (2014). Distributed representations of sentences and documents. Dans *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 1189–1197. MIT Press.

Li, B., Wisniewski, G. et Crabbé, B. (2023). Assessing the capacity of transformer to abstract syntactic representations : A contrastive analysis based on long-distance agreement. *Transactions of the Association for Computational Linguistics*, *11*, 18–33. Récupéré de https://aclanthology.org/2023

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J. et Chang, K.-W. (2019). Visualbert : A simple and performant baseline for vision and language. Récupéré de https://arxiv.org/abs/1908.03557

Lipton, Z. C. (2017). The mythos of model interpretability. Récupéré de https://arxiv.org/abs/1606.03490

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. et Stoyanov, V. (2019). Roberta : A robustly optimized bert pretraining approach. Récupéré de https://arxiv.org/abs/1907.11692

Lu, J., Batra, D., Parikh, D. et Lee, S. (2019). Vilbert : Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Récupéré de https://arxiv.org/abs/1908.02265

Lundberg, S. et Lee, S.-I. (2017). A unified approach to interpreting model predictions. Récupéré de https://arxiv.org/abs/1705.07874

Luong, M.-T. (2015). Effective approaches to attention-based neural machine translation. Dans *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, 1412–1421. Association for Computational Linguistics. Récupéré de https://aclanthology.org/D15-1166

Marcheggiani, D. et Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. Dans *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 193–203. Association for Computational Linguistics. Récupéré de https://aclanthology.org/E17-1022

Marenbon, J. (2007). *Medieval Philosophy : An Historical and Philosophical Introduction*. London, UK : Routledge.

Martin, J., Vial, T., Scieur, D., D., B. P., M., A. et L., G. (2019). Camembert : A tasty french language model. *arXiv preprint arXiv :1911.03894*. Récupéré de https://arxiv.org/abs/1911.03894

Mechouma, T., Biskri, I. et Meunier, J. G. (2022a). Reinforcement of bert with dependency-parsing based attention mask. Dans *Proceedings of the 14th International Conference on Computational Collective Intelligence (ICCCI)*, 112–122. Springer, Cham. http://dx.doi.org/10.1007/978-3-031-15779-4_11

Mechouma, T., Biskri, I. et Meunier, J. G. (2022b). Reinforcement of bert with dependency-parsing based attention mask. Dans C. Bădică, J. Treur, D. Benslimane, B. Hnatkowska, et M. Krótkiewicz (dir.). *Advances in Computational Collective Intelligence*, 112–122., Cham. Springer International Publishing.

Mechouma, T., Biskri, I. et Robert, S. (2024). Lingbert, linguistic knowledge injection into attention mechanism based on a hybrid masking strategy. Dans *Proceedings of the 2024 International Conference on Machine Learning and Applications (ICMLA)*.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. et Galstyan, A. (2022). A survey on bias and fairness in machine learning. Récupéré de https://arxiv.org/abs/1908.09635

Melamud, O., Goldberger, J. et Dagan, I. (2016). Context2vec : Learning generic context embedding with bidirectional lstm. Dans *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 51–61. http://dx.doi.org/10.18653/v1/K16-1006

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. et Khudanpur, S. (2010). Recurrent neural network based language model. Dans *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2, 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. et Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, *26*, 3111–3119.

Miller, G. A. (1995). Wordnet : A lexical database for english.

Minsky, M. (1974). A framework for representing knowledge. In P. H. Winston (dir.), *The Psychology of Computer Vision* 211–277. McGraw-Hill.

Minsky, M. (1986). *The Society of Mind*. Simon & Schuster.

Mohr, R. D. (1981). *Plato's Theory of Forms : The Metaphysical Foundation of Meaning and Reality*. Columbia, Missouri : University of Missouri Press.

Narasimhan, H., Cotter, A., Zhou, Y., Wang, S. et Guo, W. (2020). Approximate heavily-constrained learning with lagrange multiplier models. Dans *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 8693–8703. Curran Associates, Inc. Récupéré de https://proceedings.neurips.cc/paper_files/paper/2020/file/62db9e3397c76207a687c360e0243317-Paper.pdf

Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T. et Swartout, W. R. (1991). Enabling technology for knowledge sharing.

Newell, A. et Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.

Newell, A. et Simon, H. A. (1976). Computer science as empirical inquiry : Symbols and search. *Communications of the Acm*, *19*, 113–126.

Ogden, C. K. et Richards, I. A. (1923). *The Meaning of Meaning*. Harcourt, Brace & World.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M. et Carter, S. (2020). Zoom in : An introduction to circuits. *Distill*, *5*. http://dx.doi.org/10.23915/distill.00024.001

Pan, X., Mehta, M. et Srikumar, V. (2020). Learning constraints for structured prediction using rectifier networks. *arXiv preprint arXiv :2006.01209*. http://dx.doi.org/10.48550/arXiv.2006.01209

Panaccio, C. (2004). Mental language and the medieval tradition : Some philosophical presuppositions. *History and Philosophy of Logic*, *25*(4), 255–270. http://dx.doi.org/10.1080/0144534042000278196

Peirce, C. S. (1878). How to make our ideas clear. *Popular Science Monthly*.

Peirce, C. S. (1931–1958). *Collected Papers of Charles Sanders Peirce*. Harvard University Press.

Peng, N., Yan, S., Geng, Y. et Lu, Z. (2019). Structbert : Incorporating language structures into pretrained language models. Dans *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2447–2457., Florence, Italy. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/P19-1240. Récupéré de https://doi.org/10.18653/v1/P19-1240

Pennington, J., Socher, R. et Manning, C. D. (2014). Glove : Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. et Luke, Z. (2018). Deep contextualized word representations. Dans *Proceedings of NAACL-HLT 2018*, 2227–2237. Association for Computational Linguistics. Récupéré de https://aclanthology.org/N18-1202

Petito, L. A., Zatorre, R. J., Gauna, K., Nikelski, E. J., Dostie, D. et Evans, A. C. (2000). Speech-like cerebral activity in profoundly deaf people processing signed languages : implications for the neural basis of human language. *Proceedings of the National Academy of Sciences of the United States of America (Proc. Natl. Acad. Sci. U.S.A.)*, *97*(25), 13961–13966.

Prinz, J. J. (2002). *Furnishing the Mind : Concepts and Their Perceptual Basis*. MIT Press.

Prior, A. N. (1957). *Time and Modality*. Oxford : Clarendon Press.

Pulvermüller, F. (2013). *Neuroscience of Language : On Brain Circuits of Words and Serial Order*. Cambridge University Press.

Pylyshyn, Z. W. (1984). *Computation and Cognition : Toward a Foundation for Cognitive Science*. MIT Press.

Qi, D., Su, L., Song, J., Cui, E., Bharti, T. et Sacheti, A. (2020). Imagebert : Cross-modal pre-training with large-scale weak-supervised image-text data. Récupéré de https://arxiv.org/abs/2001.07966

Quillian, M. R. (1968). Semantic networks. In M. L. Minsky (dir.), *Semantic Information Processing*. MIT Press.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. et Sutskever, I. (2021). Learning transferable visual models from natural language supervision. Récupéré de https://arxiv.org/abs/2103.00020

Radford, A., Narasimhan, K., Salimans, T. et Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*. Récupéré de https://openai.com/research/language-unsupervised

Radford, A., Wu, J., Amodei, D., Clark, J. et OpenAI (2019). Language models are unsupervised multitask learners. *OpenAI Blog*. Récupéré de https://openai.com/research/language-unsupervised

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. et Liu, P. J. (2023). Exploring the limits of transfer learning with a unified text-to-text transformer. Récupéré de https://arxiv.org/abs/1910.10683

Raghavan, M., Barocas, S., Kleinberg, J. et Levy, K. (2020). Mitigating bias in algorithmic hiring : Evaluating claims and practices. Dans *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 469–481. ACM. http://dx.doi.org/10.1145/3351095.3372828. Récupéré de https://ssrn.com/abstract=3408010

Rahman, W., Hasan, M. K., Lee, S., Bagher Zadeh, A., Mao, C., Morency, L.-P. et Hoque, E. (2020). Integrating multimodal information in large pretrained transformers. Dans *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2359–2369., Online. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/2020.acl-main.214. Récupéré de https://www.aclweb.org/anthology/2020.acl-main.214

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. et Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. Récupéré de https://arxiv.org/abs/2204.06125

Rastier, F. (1996). Problématiques du signe et du texte. *Intellectica*, *23*(2), 47–70. Récupéré de https://intellectica.org/SiteArchives/archives/n23/23_04_Rastier.pdf

Ribeiro, M. T., Singh, S. et Guestrin, C. (2016a). "why should i trust you ?" : Explaining the predictions of any classifier. Récupéré de https://arxiv.org/abs/1602.04938

Ribeiro, M. T., Singh, S. et Guestrin, C. (2016b). "why should i trust you ?" explaining the predictions of any classifier. Dans *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1135–1144. http://dx.doi.org/10.1145/2939672.2939778

Rich, E. et Knight, K. (2009). *Artificial Intelligence*. McGraw-Hill.

Richens, R. H. (1956). *The first semantic network for computers was Nude : The Cambridge Language Research Unit in 1956 as an interlingua for machine translation of natural languages*.

Rosch, E. et Heider, E. (1973). Natural categories. *Cognitive Psychology*, *4*, 328–350.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L. et Zhong, C. (2021). Interpretable machine learning : Fundamental principles and 10 grand challenges. Récupéré de https://arxiv.org/abs/2103.11251

Russell, B. et Whitehead, A. N. (1910–1913). *Principia Mathematica*. Cambridge : Cambridge University Press.

Russell, S. J. et Norvig, P. (2010). *Artificial Intelligence : A Modern Approach*. Pearson Education.

Sak, H., Senior, A. et Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Dans *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1–5. http://dx.doi.org/10.1109/ICASSP.2014.6853696

Salton, G. (1971). *Automatic Information Organization and Retrieval*. New York : McGraw-Hill.

Salton, G., Wong, A. et Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Sanh, V., Debut, L., Chaumond, J. et Wolf, T. (2019). Distilbert, a distilled version of bert : Smaller, faster, cheaper and lighter. *arXiv preprint arXiv :1910.01108*. Récupéré de https://arxiv.org/abs/1910.01108

Sanh, V., Debut, L., Chaumond, J. et Wolf, T. (2020). Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. Récupéré de https://arxiv.org/abs/1910.01108

Sapir, E. (1985). *Selected Writings in Language, Culture, and Personality*. University of California Press.

Saussure, F. d. (1916). *Course in General Linguistics*. Philosophical Library. Original work published 1916.

Schank, R. C. (1969). A conceptual dependency parser for natural language. Dans *Proceedings of the 1969 Conference on Computational Linguistics*, 1–3., Sång-Säby, Sweden.

Schank, R. C. et Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*. Erlbaum.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, *3*(3), 417–457.

Searle, J. R. (2006). What is language. Dans G. Abel (dir.). *Proceedings of the German Philosophy Conference in Berlin, Kreativität. XX Deutscher Kongress für Philosophie*, Hamburg. Felix Meiner Verlag.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. et Batra, D. (2019). Grad-cam : Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, *128*(2), 336–359. http://dx.doi.org/10.1007/s11263-019-01228-7. Récupéré de http://dx.doi.org/10.1007/s11263-019-01228-7

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423. Récupéré le 2003-04-22 de http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

Sheynin, S., Benaim, S., Polyak, A. et Wolf, L. (2021). Locally shifted attention with early global integration. http://dx.doi.org/10.48550/arXiv.2112.05080

Si, Y. et Roberts, K. (2021). Three-level hierarchical transformer networks for long-sequence and multiple clinical documents classification. Récupéré de https://arxiv.org/abs/2104.08444

Simmons, R. F. (1963). Synthetic language behavior. *Data Processing Management*, *5*(12), 11–18.

Sokolowski, R. (1964). *Realism and the Background of Phenomenology*. Notre Dame, Indiana : University of Notre Dame Press.

Sowa, J. F. (1976). Conceptual graphs for a database interface. *IBM Journal of Research and Development*, *20*(4), 336–357.

Sowa, J. F. (1995). Top-level ontological categories. *International Journal of Human-Computer Studies*, *43*(5–6), 669–685.

Sun, R. (2000). Artificial intelligence : Connectionist and symbolic approaches. In *International Encyclopedia of the Social & Behavioral Sciences*

Sundararaman, D., Subramanian, V., Wang, G., Si, S., Shen, D., Wang, D. et Carin, L. (2019). Syntax-infused transformer and bert models for machine translation and natural language understanding. Récupéré de https://arxiv.org/abs/1911.06156

Sung, M., Jeon, H., Lee, J. et Kang, J. (2020). Biomedical entity representations with synonym marginalization. http://dx.doi.org/10.48550/arXiv.2005.00239

Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M. et Liu, Y. (2020). Neural machine translation : A review of methods, resources, and tools. *AI Open*, *1*, 5–21. http://dx.doi.org/10.1016/j.aiopen.2020.11.001

Tenney, I., Das, D. et Pavlick, E. (2019a). BERT rediscovers the classical NLP pipeline. Dans A. Korhonen, D. Traum, et L. Màrquez (dir.). *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593–4601., Florence, Italy. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/P19-1452. Récupéré de https://aclanthology.org/P19-1452/

Tenney, I., Das, D. et Pavlick, E. (2019b). Bert rediscovers the classical nlp pipeline. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, 4596–4611. Récupéré de https://aclanthology.org/N19-1452

Thérien, G. (1989). Sémiotique et intelligence artificielle. *Études littéraires*, *21*(3), 67–80. http://dx.doi.org/10.7202/500871ar

Tous, R., Guerrero-Zapata, M. et Delgado, J. (2011). Semantic web for reliable citation analysis in scholarly publishing. *Information Technology and Libraries*, *30*(1), 24–33. http://dx.doi.org/10.6017/ital.v30i1.3042

Turney, P. D. et Pantel, P. (2010). From frequency to meaning : Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188.

van Inwagen, P. (2004). A theory of properties. *Oxford Studies in Metaphysics*, *1*, 107–138.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. et Polosukhin, I. (2017). Attention is all you need. Dans *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*, 6000–6010. Curran Associates, Inc. Récupéré de https://arxiv.org/abs/1706.03762

Wang, S., Li, B., Khabsa, M., Fang, H. et Ma, H. (2020). Linformer : Self-attention with linear complexity. http://dx.doi.org/10.48550/arXiv.2006.04768

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viegas, F. et Wilson, J. (2019). The what-if tool : Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, p. 1–1. http://dx.doi.org/10.1109/tvcg.2019.2934619. Récupéré de http://dx.doi.org/10.1109/TVCG.2019.2934619

Whorf, B. L. (1940). Science and linguistics. *Technology Review*, *42*(6), 229–231, 247–248.

Wiegreffe, S. et Pinter, Y. (2019). Attention is not not explanation. Récupéré de https://arxiv.org/abs/1908.04626

Wittgenstein, L. (2001). *Philosophical Investigations* (1st éd.). Oxford : Blackwell Publishing. First edition published in 1953.

Wu, J., Jiang, B., Li, X., Liu, Y.-F. et Yuan, J. (2024). A new adaptive balanced augmented lagrangian method with application to isac beamforming design. Récupéré de https://arxiv.org/abs/2410.15358

Wu, S., Zhang, D., Zhang, Z., Yang, N., Li, M. et Zhou, M. (2018). Dependency-to-dependency neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*, 1–1. http://dx.doi.org/10.1109/TASLP.2018.2855968. Récupéré de https://doi.org/10.1109/TASLP.2018.2855968

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R. et Cohen, W. W. (2019). Xlnet : Generalized autoregressive pretraining for language understanding. Dans *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada. Neural Information Processing Systems Foundation.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. et Hovy, E. (2016a). Hierarchical attention networks for document classification. Dans K. Knight, A. Nenkova, et O. Rambow (dir.). *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, 1480–1489., San Diego, California. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/N16-1174. Récupéré de https://aclanthology.org/N16-1174/

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. et Hovy, E. (2016b). Hierarchical attention networks for document classification. 1480–1489. http://dx.doi.org/10.18653/v1/N16-1174

Zhang, D., Yuan, Z., Liu, Y., Liu, H., Zhuang, F., Xiong, H. et Chen, H. (2021). Domain-oriented language modeling with adaptive hybrid masking and optimal transport alignment. Dans *Proceedings of the 2021 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2145–2153. http://dx.doi.org/10.1145/3447548.3467215

Zhao, B., Liu, W. et Lu, X. (2018). Hsa-rnn : Hierarchical structure-adaptive rnn for video summarization. 7405–7414. http://dx.doi.org/10.1109/CVPR.2018.00773

Zhao, C., Zhou, X., Xie, X. et Zhang, Y. (2024). Hierarchical attention graph for scientific document summarization in global and local level. 714–726. http://dx.doi.org/10.18653/v1/2024.findings-naacl.45