Les utilisations possibles de l'Intelligence Artificielle dans la linguistique historique

3 étudiants de CPBx

0.1 Résumé

0.2 Abstract

0.3 Remerciements

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Mise en contexte pour arriver à la problématique, quel est le potentiel de l'intelligence artificielle dans la linguistique historique ? ... ?

Chapitre 2

La linguistique historique et l'Intelligence Artificielle

2.1 La linguistique historique

2.1.1 Introduction à la linguistique historique

Définir ce qu'est la linguistique historique, ce qu'elle étudie, et les mots de vocabulaires que nous allons rencontrer tout au long du mémoire.

La linguistique est la discipline s'intéressant à l'étude du langage. Elle se distingue de la grammaire, dans la mesure où elle n'est pas prescriptive mais descriptive. La prescription correspond à la norme, c'est-à-dire ce qui est jugé correct linguistiquement par les grammairiens. A l'inverse, la linguistique descriptive des linguistes se contente de décrire la langue telle qu'elle est et non telle qu'elle devrait être. La linguistique a des rapports très étroits avec d'autres sciences qui tantôt lui empruntent des données, tantôt lui en fournissent. Les limites qui l'en séparent n'apparaissent pas toujours nettement. Par exemple, la linguistique doit être soigneusement distinguée de l'ethnographie et de la préhistoire, où la langue n'intervient qu'à titre de document ; distinguée aussi de l'anthropologie, qui n'étudie l'homme qu'au point de vue de l'espèce. La linguistique peut se définir comme une science qui a pour objet l'étude du langage, des langues envisagées comme systèmes sous leurs aspects phonologiques, syntaxiques, lexicaux et sémantiques.

La langue est une partie du langage. C'est un produit social de la faculté du langage et un ensemble de conventions nécessaires, adoptées par le corps social pour permettre l'exercice de cette faculté chez les individus. La langue est une convention extérieure à l'individu. Elle n'existe qu'en vertu d'une sorte de contrat passé entre les membres de la communauté. C'est une institution sociale. La langue n'est pas une fonction du sujet parlant, elle est le produit que l'individu enregistre passivement. La parole est un fait de langage distinct de la langue, c'est un acte individuel qui se distingue des institutions sociales. Elle est l'initiative d'un sujet parlant et acte de volonté et d'intelligence. Le langage quant à lui est un système de signes qui permet l'expression de la

communication.

Les signes linguistiques forment un ensemble indissociable de sons et de sens, de signifiants (réalité matérielle, acoustique ou visuelle) et de signifiés (sens d'un mot). L'aspect "matériel" du signe, le signifiant, est en fait une réalité psychique : il ne s'agit pas du son comme tel, mais du son perçu. C'est pourquoi Saussure parle d' "image acoustique". L'aspect "conceptuel" du signe, le signifié, est également une réalité psychique : il ne faut pas confondre le signifié avec le référent (ce à quoi renvoie le signe dans la réalité extérieure). Un signe a un sens (son signifié) que l'objet auquel il fait référence par ce sens existe ou non dans la réalité.

La linguistique historique permet de déchiffrer des textes anciens, mais aussi d'étudier la dynamique migratoire des humains au cours de l'histoire, grâce à des mots et des variations de langage communes. C'est une discipline de la linguistique qui étudie l'histoire et l'évolution des langues, et des familles des langues. Elle peut aussi être désignée sous le nom de linguistique comparée ou de grammaire comparée. La principale méthode de travail consiste à une comparaison entre les différents états d'une même langue ou entre des langues différentes mais issues d'un même ancêtre, mais aussi à rechercher des concordances syntaxiques ou sémantiques régulières, obéissant presque à des lois de probabilité, mettant en évidence la relation parenté entre les langues. De plus, cette comparaison et ce travail de reconstruction permet de retrouver la langue mère à partir de la langue fille. La linguistique historique permet alors de caractériser la nature des évolutions, innovations et rétentions l'état initial et les états finaux (phonétique, phonologie, lexique, syntaxe etc.)

La linguistique historique aurait été introduite par Sir William Jones (1746-1794) lorsqu'il émet l'hypothèse que le grec, le latin et le sanskrit auraient des origines communes, menant à une langue Indo-européenne. Une partie de ses hypothèses se seront révélées incorrectes plus tard, mais la langue Proto Indo-Européenne est toujours étudiée aujourd'hui, se voulant être la protoforme du latin, du gothique, du celte, du grec et du perse. Si toutes les langues indo-européennes descendent d'une langue primitive commune, quel était donc le peuple qui parlait cette langue, où se situait-il et à quelle époque ? Pour essayer de répondre à cette question, on se base généralement sur des éléments de linguistique et d'archéologie. Selon l'hypothèse kourgane, l'indo-européen viendrait d'un peuple semi-nomade ayant vécu il y a environ 6000 à 7000 ans dans la steppe située au nord de la Mer Noire, aux environs de l'actuelle frontière entre la Russie et l'Ukraine. Dans ce scénario, ce peuple de guerriers et de cavaliers conquérants aurait entrepris de nombreuses migrations, permettant ainsi la diffusion de leur langue en Europe et en Asie. Selon l'hypothèse anatolienne, l'indo-européen trouverait son origine en Anatolie (l'actuelle Turquie) il y a environ 8 à 10 000 ans, à l'époque de l'apparition de l'agriculture dans cette région. La langue se serait alors répandue dans toute l'Eurasie en même temps que la diffusion de l'agriculture. C'est actuellement l'hypothèse Kourgane qui est considérée comme la plus vraisemblable par les spécialistes, sans cependant être une certitude.

2.1.2 Les différents principes

Évidemment cette science repose sur des concepts, allant des propriétés synchoniques des mots aux à leurs aspects diachroniques.

Il existe deux manières d'aborder la linguistique historique : l'étude d'une langue à un moment donné, la méthode synchronique, ou de l'étudier au cours du temps, la méthode diachronique. En plus de cela, il existe un principe fondamental : la proximité entre des langues peut traduire une histoire commune. Cependant, il faut distinguer les emprunts à une ou plusieurs autres langues des caractéristiques ancestrales de la langue étudiée. Après avoir fait la distinction, on peut dire que deux langues ayant des caractéristiques voisines descendent d'une langue plus ancienne commune. Par exemple, les langues romanes comme l'espagnol et le français descendent du latin).

Ainsi, à partir de ces langues voisines, on peut tenter de reconstruire leur langue ancestrale, la protolangue ou langue-mère. Selon plusieurs théories de linguistiques, cette protolangue est hypothétique et elle descendrait elle-même d'une langue plus ancienne encore. Pour reconstruire cette protolangue, on doit l'étudier sur plusieurs niveaux : la syntaxe, le lexique, la phonologie, la phonétique De là, on peut aborder un processus récursif : à partir d'un ensemble de protolangues apparentées, on peut donc reconstruire cette protolangue plus ancienne. Pour cela, la linguistique historique doit prendre en compte le sens des mots, mais aussi la forme des mots, c'est-à-dire les sons qui la composent. Cependant, les mots de deux langues peuvent être proches, par hasard, à cause d'un emprunt, ou d'une évolution commune des langues. Par exemple, meli veut dire miel en hawaïen et en grec ancien, sans cependant être liés. Algorithme, girafe, orange, sont des mots français empruntés à l'arabe. Et le mot main et mano en espagnol veut dire la même chose, étant un exemple d'évolution commune.

Un phonème est un élément sonore du langage articulé considéré d'un point de vue physiologique (disposition des organes vocaux) et d'un point de vue acoustique (perception auditive). Comme expliqué précédemment, l'évolution des langues se fait de façon plus ou moins régulière, quantifiable, et prévisible. Ces changements sont appelés correspondances régulières les transformations phonologiques qui changent sans exception un phonème A de la langue 1 en phonème B de la langue 2. Elles indiquent une relation génétique de deux langues, étant liées à l'hypothèse d'évolution des sons. Un morphème est quant à lui une unité minimale de nature grammaticale. Un phonème se définit au niveau phonologique, le morphème au niveau morphologique. Chaque unité peut se substituer avec des unités de même niveau et chaque unité s'intègre dans une unité de niveau supérieur, dont elle est un constituant. Le morphème a une signification, c'est une unité minimale significative, la plus petite unité qui possède encore du sens. C'est l'étude de la façon dont sont formés les mots. chant-eur, jongl-eur, ou jou-eur. Le suffixe "eur" signifie celui qui fait l'action. Ainsi, "eur" est un morphème.

Un phonème est un élément sonore du langage articulé considéré d'un point de vue physiologique (disposition des organes vocaux) et d'un point de vue acoustique (perception auditive). Comme expliqué précédemment, l'évolution des langues se fait de façon plus ou moins régulière,

quantifiable, et prévisible. Ces changements sont appelés correspondances régulières les transformations phonologiques qui changent sans exception un phonème A de la langue 1 en phonème B de la langue 2. Elles indiquent une relation génétique de deux langues, étant liées à l'hypothèse d'évolution des sons. Un morphème est quant à lui une unité minimale de nature grammaticale. Un phonème se définit au niveau phonologique, le morphème au niveau morphologique. Chaque unité peut se substituer avec des unités de même niveau et chaque unité s'intègre dans une unité de niveau supérieur, dont elle est un constituant. Le morphème a une signification, c'est une unité minimale significative, la plus petite unité qui possède encore du sens. C'est l'étude de la façon dont sont formés les mots. chant-eur, jongl-eur, ou jou-eur. Le suffixe "eur" signifie celui qui fait l'action. Ainsi, "eur" est un morphème.

Ces correspondances régulières sont détectées grâce à la méthode de comparaison. Pour cela, elle considère un ensemble de mots. On doit alors apparier les mots selon leur sens, en prenant en compte les éventuels glissements sémantiques. Après cela, on doit discriminer les emprunts après les avoirs détectés. Et après cela, on doit chercher les régularités entres les mots et tenter de d'expliquer les différentes évolutions qui ont pu conduire au cas étudié. (rajouter tableaux + texte explication tableaux + sources tableau)

Cependant, si des emprunts ne sont pas détectés, ils vont fausser toute mesure, en conduisant à sous-estimer la profondeur d'un ancêtre commun à plusieurs langues. Pour cela, il existe des listes de mots du lexique simple, pouvant être exploités par les linguistes, comme par exemple les listes de Swadesh. (https ://en.wiktionary.org/wiki/Appendix :Latin Swadesh list). Ces listes sont utilisées en glottochronologie et en lexicostatistique, qui sont deux méthodes développées par Morris Swadesh.

La glottochronologie qui est la méthode de datation des langues proposée par Swadesh fut comparée à la détermination de l'âge des fossiles à partir de la désintégration radioactive du carbone 14, qui est constante (elle n'est cependant plus utilisée car sujette à beaucoup d'inexactitudes).

La lexicostatistique est une méthode utilisée en linguistique comparative et historique pour mesurer la proximité entre différentes langues. Elle se base sur l'analyse statistique des mots partagés entre ces langues, notamment les cognats, c'est-à-dire les mots ayant une origine commune. Cette approche permet d'évaluer le degré de parenté entre deux langues et de reconstituer leur histoire évolutive. En comparant la fréquence des cognats, on peut ainsi déterminer si les langues étudiées sont issues d'une même langue ancestrale. Toutefois, la lexicostatistique présente des limites, notamment en raison de possibles emprunts ou convergences culturelles qui peuvent fausser les résultats.

Cependant, bien que ces listes soient toujours utilisées, ces deux méthodes ont fait place à une application récursive de reconstruction de langue, plus adaptée et tenant compte du contexte sémantique, phonétique et syntaxique des langues.

Après une reconstruction de ces protolangues, on arrive à regrouper 12 macro-familles de langues : Ces macro-familles sont cependant aussi sujettes à débat. Bien qu'on admette comme

très plausible que les proto-langues des familles de langues du monde ont fait partie à leur tour de familles encore plus anciennes, il n'y a pas d'unité de vues concernant la possibilité de reconstruire les proto-langues des superfamilles, parce qu'après une certaine période, les langues changent dans une telle mesure qu'on ne peut plus leur détecter une origine commune. (cf figure)

2.1.3 Les atouts de l'Intelligence Artificielle dans ce domaine

La linguistique historique fait face à de nombreux problèmes récurrents (traiter une grande quantité de textes pour l'homme, remarquer des motifs dans ces documents historiques). Alors que ce travail pourrait être effectué par une machine, grâce à sa capacité à traiter un grand nombres de données, et à chercher des similarités dans ces données. Avant, de voir les tâches où l'Intelligence Artificielle peut intervenir, il est d'abord nécessaire de voir en détail la conception des ces IA.

Résoudre des problèmes de Linguistique Historique avec un ordinateur nécessite de lui faire traiter du contenu textuel devant être abstrait sur des terrains parmi ceux de la **phonétique**, de la **sémantique**, de la **morphologie** ou encore de la **syntaxe**.

Développper un exemple pour illustrer ces 4 niveaux d'abstractions

La réalisation de ces abstractions s'inscrit dans le Traitement Automatisé du Langage Naturel (TAL), un domaine à cheval entre la Linguistique et l'Informatique. L'Intelligence Artificielle y occupe une place centrale pour sa capacité à effectuer des approximations améliorables avec de l'entraînement.

2.2 L'IA dans le Traitement Automatisé du Langage Naturel

2.2.1 Introduction à l'apprentissage automatique

Qu'est ce qu'une intelligence artificielle ?
Qu'est ce qu'un réseau de neurones ?
Quel est le principe derrière l'apprentissage automatique ?
Définition des apprentissages supervisés/non supervisés Définition de propagation avant. Définition rétro-propagation du gradient. Exemple de FFNN pour tâche de classification

Un important nombre de problèmes informatiques peut être résolu à travers la détermination d'une fonction mathématique f d'un espace vectoriel $\mathbb{K}^n$ vers un espace vectoriel $\mathbb{K}^{n'}$ (avec $\mathbb{K}$ correspondant à $\mathbb{R}$ ou $\mathbb{C}$).

Lorsqu'un algorithme conventionnel est développé pour réaliser un tâche, f a déjà implicitement été trouvé. Par exemple, derrière un traitement opéré sur une chaîne de caractères, elle existe bien, avec pour entrée une séquence de n caractères encodés sous forme de bits qui forme un vecteur de l'espace $\mathbb{R}^n$ et pour sortie un élément d'un espace $\mathbb{R}^{n'}$ représentant la chaîne de sortie.

En revanche, de nombreux cas demeurent où il est difficile – voire impossible – de poser une expression mathématique ou un algorithme pour répondre à certains problèmes. On considère alors f comme hypothétique et on cherche à l'approcher à partir d'un **modèle**, qu'on construit à partir des informations qu'on dispose sur f, comme un ensemble de ses points $\{(x_k, y_k = f(x_k)), k \in S\}$, à travers une tâche dite de **régression**.

Les **réseaux de neurones** sont des outils performants pour établir des modèles. Mathématiquement, ce sont des compositions d'applications non-linéaires et linéaires recevant un vecteur d'entrée représentant une donnée et sortant un vecteur de sortie représentant un résultat dans un format cohérent avec le problème.

Définitions, du neurone au réseau

Le neurone artificiel le plus élémentaire effectue la **somme pondérée** des coefficients du vecteur d'entrée, à laquelle il ajoute une valeur de **biais** pour enfin calculer l'image de la somme à travers une fonction non-linéaire dite **d'activation**. La sortie du neurone est donc un réel ou un complexe. Si on la note y_i, qu'on note x le vecteur d'entrée dans $\mathbb{K}^n$, w_i le vecteur de **poids** associé au neurone, b_i son biais et σ sa fonction d'activation, on a :

$$y_i = \sigma(b_i + \sum_{j=0}^{n} w_{ij}x_j) = \sigma(b_i + < w_i, x >) \tag{2.1}$$

[1, section 1]

Une **couche de neurones** est la mise en commun d'un nombre abritraire N de neurones devant prédire des sorties y_i différentes. Leurs vecteurs de poids w_i diffèreront donc. En revanche, leur fonction d'activation est identique. La sortie d'une couche est donc un vecteur y pouvant s'écrire comme dans l'équation 2.2 [1].

$$\begin{aligned} y &= \begin{pmatrix} \sigma(b_1 + < w_1, x >) \\ \sigma(b_2 + < w_2, x >) \\ \vdots \\ \sigma(b_i + < w_i, x >) \\ \vdots \\ \sigma(b_N + < w_N, x >) \end{pmatrix} = \sigma \begin{pmatrix} b_1 + \sum_{j=0}^{n} w_{1j}x_j \\ b_2 + \sum_{j=0}^{n} w_{2j}x_j \\ \vdots \\ b_i + \sum_{j=0}^{n} w_{ij}x_j \\ \vdots \\ b_N + \sum_{j=0}^{n} w_{Nj}x_j \end{pmatrix} \\ &= \sigma \left(\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_N \end{pmatrix} + \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \dots & w_{Nj} & \dots & w_{Nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{pmatrix} \right) = \sigma(b + Wx) \end{aligned} \tag{2.2}$$

Un réseau neuronal est ainsi formé à partir de la mobilisation d'une ou plusieurs couches. L'intuition derrière l'utilisation de couches intermédiaires, qu'on nomme des **couches cachées**, est que la machine puisse être capable d'apprendre à construire des **représentations adéquates** des données pour effecuter la prédiction finalement voulue avec pertinence. On parle alors d'**apprentissage**

1. On y confond la fonction d'activation avec la fonction vectorielle σ s'appliquant indépendamment à chaque coefficient.

profond et cette technique offre des réponses face aux difficultés d'abstraction soulevées par les problèmes de TAL.

L'agencement des couches et la manière de calculer la sortie finale à partir de chacune d'elles, qu'on peut rassembler sous le terme de "mode de **propagation**", est un **paramètre architecturale** à part entière qu'il faut judicieusement définir en fonction de la tâche à réaliser. Pour traiter de l'utilisation de l'IA dans le TAL, au moins deux principaux types de réseaux de neurones seront introduits au cours de ce chapitre, différant par la nature cyclique ou non de l'enchaînement de leurs couches internes.[1, introduction + section 3] [*ajouter illustration*]

Processus d'apprentissage

L'entraînement d'un réseau de neurones s'effectue à travers des **ajustements des poids et des biais dans chaque couche**, dans le cadre d'une **minimisation de fonctions de perte**, déterminée par les sorties temporairement prédites par le réseau. Ceci s'effectue par un algorithme de **descente du gradient**. Le calcul du gradient de la fonction de perte selon tous les poids du réseau s'effectue avec un algorithme de **rétropropagation de l'erreur**, une adaptation pour les réseaux de neurones de celui de la discrimination rétropropagative sur des graphes d'exécution.[1, section 6].

expliquer la diff entre le supervisé et le non-supervisé
expliquer les choix à faire sur l'entraînement (optimiseurs, taille de batch et parallélisation, nombre d'époques, teacher forcing)
[2, section 2.1.2]

Concevoir correctement

dresser une liste récapitulant les configurations à effectuer sur un réseau (topologie + paramètres d'entraînement) et introduire le paramètre d'initialisation

expliquer que le choix des paramètres se fait expérimentalement, à partir d'intuitions basées sur des travaux antérieurs sur de la conception de réseaux

expliquer comment évaluer un réseau

Exemple de la tâche de classification

introduire le réseau FFNN
fonction d'activation softmax à la fin
entropie croisée pour la fonction de perte

2.2.2 Traitement des donnnées

Avant de donner à un réseau de neurones des données au format quelconque récupérées pour son entraînement, il est nécessaire d'effectuer un pré-traitement dessus. Le but est de les rendre correctes et compréhensibles pour l'intelligence artificielle. Cette préparation s'effectue selon le type de tâche souhaité. Néanmoins, dans sa généralité, les étapes de préparation des données en

TAL restent les mêmes.

Tout d'abord, il faut normaliser (nettoyer) la base de données qui peut être, par exemple, un corpus de textes, une liste de mots, provenant de la toile, ou d'un système de transcription audio-visuelle [2] Dans ces données se trouvent évidemment des mots, avec des « caractéristiques »de la langue [3] tel que les ponctuations, les lettres en capitales, les chiffres, ou bien les caractères spéciaux (comme le dièse que l'on retrouve souvent dans les *tweets*). Seulement, il arrive parfois que certains de ces éléments rajoutent inutilement de la complexité pour une machine, sans que cela apporte plus de sens, ou bien sont non-désirées (voir incorrectes), ainsi on décide de les supprimer (ou les remplacer) pour ne garder que ce qui nous intéresse. [Exemple]

Après avoir nettoyé notre base de données, il faut segmenter notre texte en mots ou sous-mots, autrement dit, il faut tokeniser notre texte. La tokenisation correspond à la segmentation de chaînes de caractères (comme notre texte) en tokens (mots, sous-mots, ponctuations). Par la suite, on peut extraire l'ensemble des tokens uniques d'un texte que l'on nommera : vocabulaire. Prenons un exemple « J'aime les bananes », une approche naïve est de tokeniser notre texte suivant les espaces [4] ce qui nous donne les tokens [« J'aime », « les », « bananes. »] avec un vocabulaire de taille 3 (dans cette phrase tout les tokens sont uniques). Cependant, cette approche pose des problèmes, en commençant par le token « bananes. »qui a la même signification avec ou sans point, mais qui sera considéré comme différent pour une IA. Nous pourrions ajouter la séparation suivant la ponctuation, mais alors nous obtiendrons pour « J'aime »les tokens [« J », « ' », « aime »] qui est une forme tout aussi problématique. Et il existe encore de nombreux cas (les abréviations, les points de suspension, etc.) où ce type de tokenisation pose problème. Une approche alternative est de tokeniser suivant des règles définis, par exemple, de prendre en compte les contractions comme « J'aime »et de le transformer en deux tokens [« Je », « aime »], qui est une méthode beaucoup plus efficace que la première approche, mais montrera des limites face à des situations (ou mots) rares, ou alors il faudrait spécifier de nouvelles règles pour gérer ces cas. Ainsi, la solution proposée est une approche statistique, consistant à décomposer de plus en plus un mot en sous-mots [5] au fil que sa fréquence diminue. [Exemple]. Les quatres algorithmes de tokenisation en sous-mots les plus utilisées sont le *Byte-Pair Encoding* (Sennrich et al., 2016), l'*unigram language modeling* (Kudo, 2018), le *WordPiece* (Schuster et Nakajima, 2012), et le *SentencePiece* [6] (Kudo et Richardson, 2018).

Pour la tâche de classification, vous avez vu que les mots d'entrées, [exemple], étaient convertis en une liste de nombre, autrement dit un vecteur, sur lequel il a été effectué des calculs afin d'obte-

2. Dans le cadre de ce mémoire, le but n'est pas de savoir comment on parvient à extraire du texte à partir de ces sources, mais plutôt de savoir comment on peut rendre ce texte compréhensible à une l'intelligence artificielle choisie.

3. Pour simplifier, nous avons pris l'exemple d'une base de données monolingue, mais il est possible qu'elle comporte plusieurs langues. Dans tout les cas, le procédé sera le même, si ce n'est qu'il faut s'adapter à chaque langue.

4. Vous remarquerez déjà que ce processus ne s'applique pas au langue comme le Japonais, ou le Chinois.

5. Remarquez que le terme *mot* a un sens différent que celui qui le précède.

6. Cet algorithme est une implémentation des deux premiers.

nir un résultat (un nouveau vecteur), [exemple]. Une machine, un réseau de neurones, ne comprend que des nombres et ne sait procéder qu'à des calculs. Il existe différentes façons de convertir un token en un vecteur numérique, mais on retiendra deux méthodes l'encodage 1 parmi n et le plongement lexical (respectivement et plus communément appelés en anglais le *one-hot encoding* et le *word embedding*).

L'encodage 1 parmi n consiste à créer un vecteur binaire de la taille du vocabulaire $|V|$, c'est-à-dire, que chaque dimension correspond à un mot dans le vocabulaire. Dans ce vecteur binaire, la valeur est 1 pour la dimension correspondant au mot dans le vocabulaire, et 0 pour toutes les autres dimensions. Ainsi, en supposant que la dimension du mot « bananes »se trouve à la troisième dimension (autrement dit c'est le troisième mot de notre vocabulaire V), sa représentation correspondra à un 1 à la troisième dimension et à des 0 sur les autres, soit le vecteur :

$$\begin{bmatrix} 0 & 0 & 1 & 0 & \dots & 0 \\ 1 & 2 & 3 & 4 & \dots & |V| \end{bmatrix}$$

Comme vous pouvez le voir l'avantage de cet encodage est sa simplicité de mise en oeuvre. Mais le problème survient quand le vocabulaire devient très grand, les vecteurs générés, par définition, deviennent à leurs tours très grands et très dispersés (beaucoup de 0 et peu de 1), ce qui entraine une augmentation de la complexité du modèle (le nombre de dimension pour décrire les données) et des temps de calcul. De plus, cet encodage ne prend pas en compte les relations sémantiques entre les mots, on ne peut mesurer la similarité entre les mots, car ils sont encodés de façon indépendante les uns des autres, ce qui peut être particulièrement problématique pour les nombreuses tâches de TAL tel que la compréhension ou traduction d'une langue.

Le plongement lexical, en revanche, permet de représenter les mots en encapsulant leur *sens sémantique* par des vecteurs denses de plus faibles dimensions, c'est à dire des vecteurs de nombres réels de petites tailles. Pour obtenir la représentation des tokens en vecteur sémantique, on utilise une matrice d'enchassement. [7]. Cette matrice contient, à chaque ligne, les vecteurs de plongement lexical pour chaque mot du vocabulaire, et à chaque colonne correspond une dimension du vecteur de plongement. Par ailleurs, les dimensions de ces vecteurs n'ont pas une représentation claire (Jurasky, Chap 6). Cette matrice représente un espace vectoriel pour les mots du vocabulaire, c'est à dire que les propriétés des vecteurs vont pouvoir s'appliquer pour nos mots vectorisés. Par exemple, étudier la similarité entre deux mots revient à calculer la distance entre les deux vecteurs correspondants, ou encore comme la figure X le montre, les vecteurs peuvent s'additionner/se soustraire permettant d'obtenir un nouveau vecteur qui aura garder une cohérence sémantique face à ces opérations. [Figure : Exemple de vecteurs sémantiques montrant bien qu'ils portent un sens et permettent des relations sémantiques entre eux ; vecteur "king" $-$ "man" $+$ "women" $->$ "queen"] Pour obtenir cette matrice d'enchassement, deux méthodes sont possibles, soit elle est crée par entrainement par notre propre modèle, soit elle est crée par un modèle externe (tel que Word2Vec, GloVe, BERT, etc.).

7. En réalité, les tokens sont d'abord transformer en vecteur binaire ou en un nombre correspondant à l'index dans le vocabulaire, puis on applique la matrice enchassement.

2.2.3 Architectures neuronales utiles au TAL

Quels sont les différents outils ? Suivant, comment les parties précédentes ont été traités, ou comment les parties futures seront discutées, cette partie pourrait ne pas être nécessaire. Sinon, elle regroupera l'idée de comment on passe de notre langue naturelle à celle de la machine, de passer aux mots à des vecteurs ? Quels traitements théoriques (théoriques pour ce distinguer de la pratique dans la partie future) doient être effectués sur les mots ? En fait cette partie fait référence aux chapitres 2 et 6 de Jurasky. Voir même le chapitre 9, en supprimant la sous partie précendente pour pouvoir parler directement des réseaux de neurones appliqués à la linguistique, en d'autres termes, des réseaux récurrents, des modèles séquentielles (encodeurs-décodeurs) avec l'attention, et des Transformers.

Réseaux de neurones récurrents

...

Transformeurs

Précédemment, avec les RNN et les LTSM, nous avons introduit le mécanisme d'attention, permettant au réseau de se focaliser sur la manière dont les mots (éloignés) sont reliées les uns aux autres. Seulement, comme nous l'avons vu aussi, ces réseaux se basent sur des connexions récurrentes, rendant le calcul coûteux et la parallélisation difficile. Ainsi, pour pallier ce problème et gagner en performance, un nouveau modèle de réseau de neurones apparait sous le nom de *Transformers* (Transformeurs en français) dans le papier « Attention Is All You Need » de Vaswani et al. (2017). Ce modèle de type encodeur-décodeur se base sur l'attention multi-têtes, l'innovation majeure des *Transformers* [8].

L'attention multi-têtes permet d'étudier tous les vecteurs d'entrées, comme des mots, en même temps (de façon parallèle), et dont chaque tête h qui la compose se focalise sur un aspect des *interactions* entre les différents éléments de la séquence, [exemple].

Chaque tête contient un module d'auto attention (*self-attention* en anglais) qui est utilisé pour permettre à chaque tokens x_i de pouvoir *intéragir* avec tous les autres tokens de la séquence X. Pour cela on extrait pour chaque x_i un trio de vecteurs : un vecteur requête q_i, un vecteur clé k_i, un vecteur valeur v_i. Le vecteur requête (*query*) correspondant au vecteur sur lequel on porte notre attention et qui sera comparé à tous les autres vecteurs, nommés les vecteurs de clés (*keys*). Puis, nous avons le vecteur valeur(*value*) qui sera utilisé pour représenter la « valeur sémantique (du mot) » et sera multiplié par le poids d'attention calculé avec les autres vecteurs (requêtes

8. Par simplification, nos efforts se concentrerons sur l'attention multi-têtes, sans évoquer qu'un *Transformers* se décompose en blocs de *transformer*, dont chaque bloc contient une unité d'attention multi-têtes (masqué ou non) et un FFNN, accompagné de connexions résiduelles et des couches de normalisation. Pour plus de détails, nous vous invitons à regarder le papier de Vaswani et al. (2017).

et clés). [Figure : Exemple de la formation des différents vecteurs q_i, k_i, v_i suivant une phrase X $[x_1, x_2, \ldots, x_i, \ldots, x_n]$]. Pour créer ces vecteurs, les vecteurs x_i sont multipliés avec les matrices d'enchassement (W^Q, W^K, W^V) qui sont obtenus à l'entrainement du modèle :

$$q_i = W^Q x_i \,;\; k_i = W^K x_i \,;\; v_i = W^V x_i$$

On pose les matrices Q, K, V respectivement l'ensemble des vecteurs q_i, k_i, v_i.

Ensuite, le calcule d'auto-attention s'effectue par la multiplication du score d'attention $((QK^T)/\sqrt{d_k})$ normalisé par la fonction *softmax* avec la matrice des valeurs V :

$$Z_h \;=\; SelfAttention(Q,K,V) \;=\; softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Enfin, toutes les matrices Z_h, de chaque tête h, sont concaténées en une seule grande matrice qui est multipliée par une matrice W^O (également obtenus à l'entrainement) pour former la matrice résultante de l'attention multi-têtes Z. [Figure : résumant toute les étapes.]

$$MultiHeadAttention(X) \;=\; (Z_1 + Z_2 + \cdots + Z_h)W^O \quad \text{h correspondant au nombre de têtes}$$

[Prenons un exemple : ...]

Cependant, au tout début, nous avons évoqué que le modèle étudié tout les tokens x_i en même temps, entrainant que le modèle ne tient pas compte de l'ordre des tokens, une information pourtant capitale. Par conséquent, pour injecter l'information de la position de nos tokens dans les vecteurs d'entrées, le papier Vaswani et al. (2017) propose une solution en additionnant, à nos tokens vectorisés sémantiquement, des vecteurs positions créent à base de fonction sinus et cosinus [9]. Ainsi, cette méthode permet à notre modèle de travailler avec tous les tokens x_i de notre phrase X en même temps tout en ayant l'information de la position de chaque mot dans leur vecteur.

Dans certaines situations, notamment dans la tâche de modélisation de langue en TAL, où le but est de prédire le prochain mot d'une phrase, l'entrainement avec ce type de Transformeurs est assez innaproprié, car vous connaissez déjà le prochain mot de votre phrase vu que vous étudiez tous les mots en même temps. Pour résoudre ce problème, on applique un masque tel que les valeurs de la partie triangulaire supérieure de la matrice QK^T sont remplacés par des $-\infty$ (qui seront transformés en 0 par la fonction *softmax*). Cette variante se nomme l'auto-attention masquée. De ce fait, le champ de visibilité de notre modèle sera réduit au mot qu'il est entrain de voir et à ceux qu'il a déjà vu, et pourra prédire de façon plus correcte (sans triche) les prochains mots de nos phrases. [Figure : montrant la matrice réduite]

9. Nous invitons le lecteur à regarder le papier de Vaswani et al. (2017) pour se convaincre de la pertinence de ce choix d'encodage de position. Simplement, retenez que cela permet au modèle de s'entrainer avec la position relative des tokens, plutôt que la position absolue.

Chapitre 3

Les contributions de l'IA dans la linguistique historique

Petite introduction avant de passer au papiers.

3.1 Restoration de documents anciens

...

3.2 Déchiffrement de langues anciennes

Une autre tâche possible par l'intelligence artificielle est le déchiffrement de langue anciennes...

Chapitre 4

Étude du cas de l'application de l'IA pour la reconstruction des proto-formes d'une langue

Ici, on se place dans un cas concret, pour montrer que ce n'est pas que de la théorie. En proposant une expérience.

4.1 État de l'art

4.1.1 Conceptualisation du problème

Définir clairement le problème du titre, énoncer et justifier le choix de notre modèle réseaux de neurones et des différents outils appliqués. Voir s'il est possible de faire apparaitre plusieurs démarches, c'est à dire, une approche statistique et une approche neuronale (toujours pour renforcer et montrer le potentiel de l'IA).

4.1.2 Dernières solutions neuronales

Solution supervisée + non supervisée

4.1.3 Limites d'applicabilité

Expliquer en quoi le non-supervisé donne plus d'espoir que le supervisé mais en quoi même cette approche présente des limites.
Transition avec la problématique de l'article scientifique

4.2 Observation expériementale d'une limite d'applicabilité d'une approche

4.2.1 Méthode

4.2.2 Récupération de la base de données

Il est fort possible que cette partie se regroupe avec la partie suivante, car il n'y aura pas grand chose à dire.

4.2.3 Normaliser les données

4.2.4 Analyse

4.2.5 Critiques

Il reste ici quelques sous parties à détailler.

Chapitre 5

Conclusion

5.1 Synthèse

Résume tout ce qui a été dit.

5.2 Les différentes limites posées aujourd'hui

une partie des limites aura déjà été traitée dans le chapitre précédent. Cette sous partie se veut résumer ces limites, et aller dans les limites générales (voir acutelles) de l'IA dans la linguisitique historique.

5.3 Les perspectives de l'IA dans la linguistique historique

Ouverture, dépassement de certaines limites, évolution des modèles.

Chapitre 6

Références

6.1 Bibliographie

Bibliographie

[1] James H. Martin DAN JURAFSKY. « Neural Networks and Neural Language Models ». In : *Speech and Language Processing*. 2023. Chap. 7. URL : https://web.stanford.edu/~jurafsky/slp3/7.

[2] Clémentine FOURRIER. « Neural Approaches to Historical Word Reconstruction ». Theses. Université PSL (Paris Sciences & Lettres), sept. 2022. URL : https://hal.inria.fr/tel-03793299.