REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERI de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'informatique

MEMOIRE

DE FIND'ETUDE

En vue de l'obtention d'un diplôme de Master Académique en informatique.

Option : Systèmes Informatiques (SI).

THEME : Apprentissage automatique du dialecte algérien.

Encadré par :
Mme AMIROUCHE Fatiha

Réalisé par :
Mr ABBAS Jugurtha
Mr HAMDAD Anis

Jury :
Mme AOUDJIT Rachida : Présidente
Mr SADI Samy

Promotion 2019 – 2020

Remerciements

« بسم الله الرحمن الرحيم»

Nous tenons tout d'abord à remercier ALLAH le tout puissant, qui nous a donné la force, la capacité et surtout la patience pour accomplir ce travail.

L'aboutissement de ce travail de mémoire n'est que le fruit d'échanges, de conseils et de soutiens d'un grand nombre de personnes auxquelles on tient à adresser nos profondes reconnaissances.

On tient à exprimer nos gratitudes et nos vifs remerciements à notre encadrante de mémoire Mme. AMIROUCHE Fatiha Professeur à l'Université de Mouloud Mammeri de Tizi-Ouzou pour sa disponibilité, son soutien, ses conseils avisés, son encouragement ainsi que pour sa patience. Ce qu'on a appris en travaillant avec elle ne se limite pas à l'aspect scientifique mais s'étend aux aspects humain et relationnel. On la remercie infiniment.

Nous remercions nos très chers parents, qui ont été toujours là pour nous, « vous avez tout sacrifié pour vos enfants n'épargnant ni sante ni efforts, on ne pourra jamais assez-vous remercier. Vous avez su nous protéger et nous donner une éducation dont tout enfant rêverait de recevoir. Vous nous avez donné un magnifique modèle de labeur et de persévérance. Nous sommes redevables d'une éducation dont nous sommes fiers. Merci infiniment, on vous aime sans limites ».

Notre reconnaissance va aussi à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'Informatique, Universitaire Mouloud MAMMERI de Tizi Ouzou et à nos collègues de la promotion 2019-2020. On remercie également tous ceux qui ont participé de près ou de loin à élaborer ce travail.

Nos remerciements sincères vont également à toutes les personnes qu'on a pu côtoyer quotidiennement au sein du L'UMMTO, leur bonne humeur quotidienne sans faille et leur capacité de travail en équipe exemplaire. On terminera cette partie en remerciant tous nos amis et amies, ceux et celles qu'on a eu la chance de côtoyer et qui nous ont toujours encouragés et supportés moralement.

Enfin, à tous ceux qu'on n'a pas pu citer, auxquels on réitère nos sincères remerciements. À vous tous, Merci !

Table des matières

Liste des tableaux

Table des figures

Introduction générale :

L'apprentissage automatique du langage est une discipline qui vise à comprendre et à apprendre le langage naturel des humains, cette dernière regroupe les trois domaines suivants : l'informatique, l'intelligence artificielle et la linguistique. L'évolution et l'apparition de nouvelles techniques d'intelligence artificielle ont permis de réaliser des exploits impressionnants en apprentissage du langage. De nos jours plusieurs applications utilisent l'apprentissage automatique du langage naturel afin de réaliser des tâches plus au moins complexes comme la reconnaissance des écritures manuscrites, le traitement de la parole, la traduction et l'extraction de l'informations.

La plupart des travaux réalisés dans ce domaine s'effectuent sur les langues officielles (langue académique), laissant de côté tous ce qui est dialecte et les phénomènes liés à ce dernier, mais l'augmentation d'utilisation des réseaux sociaux et l'apparition de plusieurs bloggeurs et rédacteurs ainsi que les différentes compagnies de marketing réalisées avec le dialecte local à la région ou au pays ont fait émerger le besoin de traiter ce dernier.

L'apprentissage automatique du dialecte arabe algérien s'avère donc très important, vu son utilisation très vaste dans notre pays, commençant par les émissions télévisées, les réseaux sociaux, et on finit par les discours des autorités algériennes.

Notre travail commence par un premier chapitre qui porte sur le domaine de l'intelligence artificielle. Dans ce chapitre on va se baser sur : l'apprentissage automatique ainsi que ses différentes méthodes, la notion du *deep learning* et découvrir vers la fin le *word embeddding*.

Dans le deuxième chapitre, nous parlerons sur la présentation du dialecte arabe algérien où on parle de l'histoire de ce dernier. Ensuite, vu que le dialecte arabe algérien est composé principalement de la langue arabe, on présente la langue arabe et ses structures linguistiques, puis on fait une comparaison entre lui et la langue arabe. Nous exposons à la fin de ce chapitre les différents problèmes liés au traitement du dialecte arabe algérien et son apprentissage par la machine.

Le troisième chapitre est consacré pour l'état de l'art sur le traitement automatique du dialecte algérien, où on va faire le tour sur les différentes études menées sur ce sujet.

Dans le quatrième chapitre, nous présenterons notre approche, et notre méthode proposée pour l'apprentissage du dialecte arabe algérien, et enfin nous ferons l'implémentation et l'évaluation de cette dernière dans le cinquième et le sixième chapitre respectivement.

Chapitre I

L’apprentissage automatique

Introduction

De nos jours, les données sont devenues l'un des atouts majeurs qui constituent la richesse des entreprises. Les informations présentes, mais noyées dans la grande masse de données, sont devenues pour ces entreprises un facteur de compétitivité et d'innovation. Les GAFA (Google, Apple, Facebook, Amazon) ainsi que les acteurs des télécoms comme Orange sont des exemples d'entreprise ayant exploité les données de leurs utilisateurs/clients afin de connaitre leurs préférences à partir de leurs données de comportement. En général, ces données permettent aux analystes de découvrir et d'expliquer certains phénomènes existants ou bien d'extrapoler des nouvelles connaissances à partir des informations présentes. Pour exploiter ces grandes masses de données, de nombreuses techniques d'apprentissage automatique ont été développées. Dans ce chapitre, nous allons découvrir le domaine de l'apprentissage automatique (ou *machine learning*) en général et de l'apprentissage profond en particulier. Puis, nous exposons leur utilisation pour le langage naturel.

L'apprentissage automatique est issu du domaine plus vaste de l'intelligence artificielle (IA, ou AI en anglais pour *Artificial Intelligence*) (voir figure 1). L'IA est née dans les années 1950 grâce au mathématicien Alan Turing qui souleva alors la question d'apporter aux machines une forme d'intelligence proche de l'intelligence humaine (voir « test de Turing »[1]).

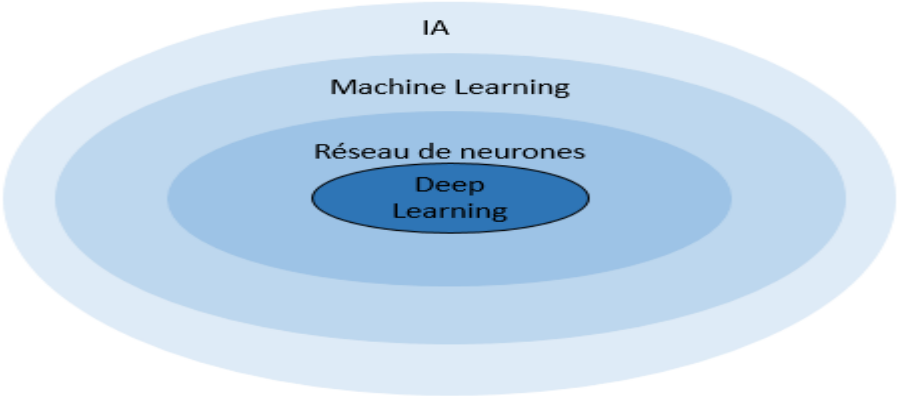


Figure 1 : L'IA et ses sous domaines

[1] Test de Turing : Si la personne qui engage les conversations n'est pas capable de dire lequel de ses interlocuteurs est un ordinateur, on peut considérer que l'ordinateur a passé avec succès le logiciel de test. Cela sous-entend que l'ordinateur et l'humain essaieront d'avoir une apparence sémantique humaine.

1 – Apprentissage automatique (*Machine Learning*)

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle (IA). C'est un ensemble de techniques et d'algorithmes permettant à la machine de comprendre ou d'effectuer des tâches jusqu'à maintenant considérées propres à l'humain, comme la reconnaissance d'images, les systèmes de décision, les prédictions …etc.

Les algorithmes d'apprentissage automatique permettent aux ordinateurs de s'entraîner sur des données existantes (données d'apprentissage ou ensemble d'entrainement, ou *training data*), pour produire de nouvelles informations (ou modèles de données). Ces informations détermineront plus tard le comportement du système (décisionnel) vis-à-vis des données saisies pour un problème complexe donné, et permettront ainsi de le résoudre. Le fonctionnement d'un algorithme d'apprentissage est représenté en figure 2 suivante :

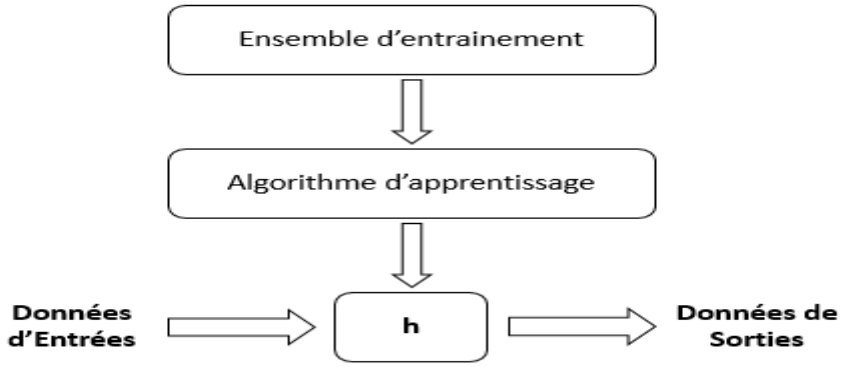


Figure 2 : Fonctionnement du Machine Learning

1.1. Les méthodes d'apprentissage :

Les méthodes d'apprentissage automatique les plus largement adoptées sont **l'apprentissage supervisé** et **l'apprentissage non supervisé.** Dans **l'apprentissage supervisé**, les algorithmes sont basés sur des données d'entrée et de sortie étiquetées par l'homme, tandis que l'**apprentissage non supervisé** ne se base pas sur des données étiquetées. Explorons donc ces méthodes plus en détail (Bousquet, 2002).

A - L’apprentissage supervisé :

Dans l'apprentissage supervisé, l'ordinateur est fourni avec des exemples d'entrées qui sont étiquetés avec les sorties souhaitées. Le but de cette méthode est que l'algorithme puisse « apprendre » en comparant sa sortie réelle avec les sorties « enseignées » pour trouver des erreurs et modifier le modèle en conséquence. L'apprentissage supervisé utilise donc des modèles pour prédire les valeurs d'étiquettes sur des données non étiquetées supplémentaires.
Par exemple, avec un apprentissage supervisé, un algorithme peut être alimenté avec des images de requins étiquetés *Poisson*, et des images d'océans étiquetés comme *Ocean*. En étant formé sur ces données, l'algorithme d'apprentissage supervisé devrait être capable d'identifier plus tard des images de requin non marquées comme *Poisson* et des images océaniques non étiquetées comme *Océan*. L'apprentissage supervisé consiste à utiliser des données historiques pour prédire des événements futurs statistiquement probables (Kim, 2017).

B - L’apprentissage non supervisé

Dans l'apprentissage non supervisé, les données sont non étiquetées. L'algorithme d'apprentissage trouve tout seul les points communs parmi ses données d'entrées. Ce type d’apprentissage est très utile pour des applications comme la *recherche d’information* (où les données non étiquetées sont plus abondantes que les données étiquetées), ou le *clustering* (à travers l'apprentissage des caractéristiques ce qui permet à la machine de classer les données brutes) (Palash Goyal, 2018).

L'apprentissage non supervisé est souvent utilisé pour la détection d'anomalies, y compris pour les achats frauduleux de cartes de crédit et les systèmes de recommandation.

Outre ces deux méthodes d’apprentissage, il en existe d’autres que l’on cite ci-après :

C. L’apprentissage semi-supervisé :

C’est une combinaison des deux précédentes méthodes. Il s’agit d’utiliser une petite quantité de données étiquetées conjointement avec une masse importante de données non-étiquetées.

D. L’apprentissage actif :

L’apprentissage actif consiste à combiner la construction de modèles à partir de données issues d’expérience, avec un système visant à produire de nouvelles données de manière à accélérer

l'apprentissage. L'idée est de chercher à chaque pas de l'apprentissage, les exemples les plus informatifs de façon à minimiser l'effort ou le temps d'apprentissage.

E. L'apprentissage par renforcement :

C'est une forme d'apprentissage en interaction dont le but est d'apprendre, à partir des expériences, ce qu'il convient de faire en différentes situations. Pour présenter les problèmes d'apprentissage par renforcement, on considère plongé au sein d'un environnement, un agent autonome qui doit prendre des décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense[2], qui peut être positive ou négative. L'agent cherche, au travers des expériences itérées, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associant à l'état courant l'action à exécuter) optimal. Comme pour l'apprentissage actif, il s'agit de trouver une bonne politique, un compromis entre l'exploitation du modèle courant et l'exploration d'un nouveau meilleur modèle (en réitérant par exemple de nouvelles expériences). C'est le cas par exemple, dans le jeu vidéo FIFA (jeu de football) où, dans une situation donnée, un joueur qui se retrouve avec un ballon possède plusieurs solutions (faire une passe, un tir, …). Cet apprentissage permet d'atteindre la meilleure solution que va faire le joueur à cette position du terrain pour marquer un but de façon plus sûr (Palash Goyal, 2018).

F. L'apprentissage par transfert :

De plus en plus utilisé notamment par la communauté « *deep learning* », cet apprentissage peut être vu comme la capacité d'un système à reconnaître et à appliquer des connaissances et des compétences, apprises à partir des tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes. Par exemple comment le fait d'avoir appris à reconnaitre des chats dans une image peut-il aider à transférer cette connaissance (ou le modèle appris) vers la reconnaissance de tigres ?

Après avoir introduit les différentes méthodes d'apprentissage automatiques, dans ce qui suit nous allons présenter quelques modèles d'apprentissage courants.

[2]la récompense indique si on est proche de la bonne prédiction, d'en dévoiler ce qu'aurait été la décision optimale.

1.2. Les modèles d’apprentissage :

1.2.1. Machine à vecteur de support (SVM) :

Les SVMs (*Support Vector Machine* ou Machine à vecteurs de support), développés dans les années 1990, sont une famille d’algorithmes d‘apprentissage automatique qui permettent de résoudre des problèmes de classification. Les SVMs ont pour but de séparer les données en classes à l’aide d’une frontière, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de la frontière.

1.2.2. Arbre de décision :

Pour un usage général, les arbres de décision sont utilisés pour représenter visuellement les décisions et montrer ou éclairer la prise de décision. Lorsqu’on travaille avec l'apprentissage automatique et l'exploration de données, les arbres de décision sont utilisés comme modèle prédictif. Ces modèles cartographient les observations de données et tirent des conclusions sur la valeur cible des données.

L'objectif de l'apprentissage par arbre de décision est de créer un modèle qui prédira la valeur d'une cible en fonction de variables d'entrée. Dans le modèle prédictif, les attributs des données qui sont déterminés par l'observation sont représentés par les branches, tandis que les conclusions sur la valeur cible des données, sont représentées dans les feuilles.

1.2.3. Forêt d’arbre décisionnelle :

Les forêts d’arbres décisionnels ou forêts aléatoires (en anglais *Random Forest classifier*). L’algorithme des forêts d’arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entrainés sur des sous-ensembles de données légèrement différents.

1.2.4. KNN (ou K plus proche voisins) :

KNN, une abréviation de K-Nearest-Neighbors, est une méthode d’apprentissage supervisé. Elle est utilisée pour la classification. L'entrée consistera en les k exemples d'entraînement les plus proches dans un espace, la sortie est l'appartenance à une classe. L’algorithme assignera un nouvel objet à la classe la plus commune parmi ses k plus proches voisins.

2 – Apprentissage profond (*Deep Learning*)

L'apprentissage profond (dit aussi apprentissage en profondeur) ou *deep learning*, est un type d'intelligence artificielle dérivé de l'apprentissage automatique, où la machine est capable d'apprendre par elle-même.

Dans l'apprentissage profond, les algorithmes peuvent être supervisés et servir à classer les données, ou non supervisés et servir à effectuer une analyse de modèle. Parmi les algorithmes d'apprentissage automatique actuellement utilisés et développés, l'apprentissage en profondeur absorbe le plus de données, et a été capable de battre les humains dans certaines tâches cognitives. En raison de ces attributs, il est devenu, avec un potentiel significatif dans le monde, l'approche de l'intelligence artificielle par excellence. La reconnaissance faciale par ordinateur et la reconnaissance vocale ont toutes deux permis de réaliser des progrès significatifs grâce à des approches d'apprentissage profond. IBM Watson[3] est un exemple bien connu d'un système qui exploite cet apprentissage (Kim, 2017).

2.1. Fonctionnement du deep learning

Au sein du cerveau humain, chaque neurone reçoit environ 100 000 signaux électriques des autres neurones. Chaque neurone en activité peut produire un effet excitant ou inhibiteur sur ceux auxquels il est connecté. Au sein d'un réseau artificiel, le principe est similaire. Les signaux voyagent entre les neurones. Toutefois, au lieu d'un signal électrique, le réseau de neurones assigne un certain poids à différents neurones. Un neurone qui reçoit plus de charge exercera plus d'effet sur les neurones adjacents (Palash Goyal, 2018).

Le *deep Learning* s'appuie sur un réseau de neurones artificiels (RNA) s'inspirant du cerveau humain. RNA est constitué de plusieurs neurones artificiels connectés entre eux. Plus le nombre de couches est élevé, plus le réseau est « profond ». Ces neurones sont organisés en couches. Le réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. La sortie d'une couche sert d'entrée de la couche suivante. Chaque couche se spécialise dans le traitement d'une caractéristique à apprendre. La couche finale fournit les résultats recherchés.

[3] Watson : est un programme informatique d'intelligence artificielle conçu par IBM dans le but de répondre à des questions formulées en langage naturel.

Exemple :

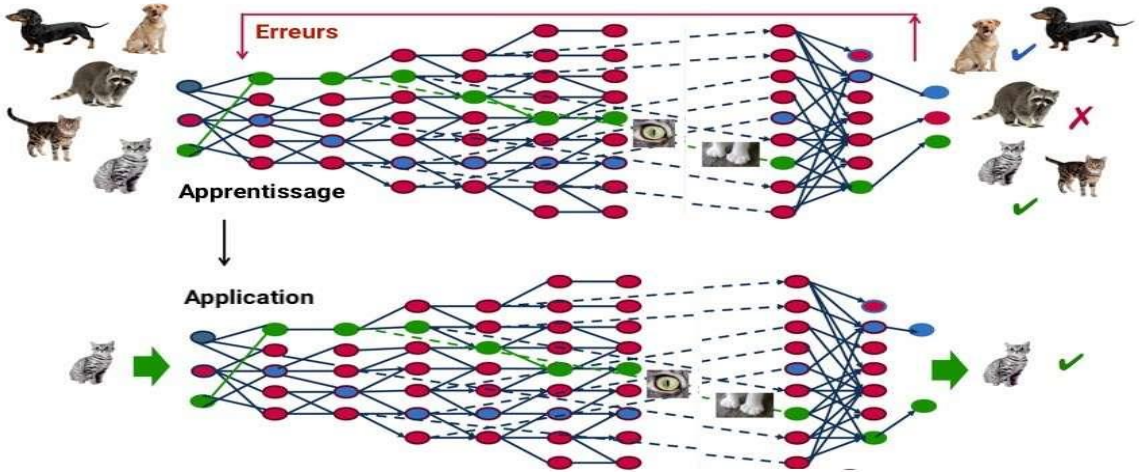


Figure 3 : Détermination d'un chat par deep learning

La figure 3 représente le fonctionnement d'un ARN dans un processus de reconnaissance d'image par auto-apprentissage profond. À chaque couche du réseau neuronal correspond un aspect particulier de l'image. Le *deep Learning* est capable d'identifier un chat sur une photo, par apprentissage supervisé ou non :

- Cas d'un apprentissage supervisé : le réseau de neurones doit être entraîné. Pour ce faire, il est nécessaire de compiler un ensemble d'images d'entraînement pour pratiquer le *deep Learning*. Cet ensemble va regrouper des milliers de photos de chats différents, mélangés avec des images d'objets qui ne sont pas des chats. Ces images sont ensuite converties en données et transférées sur le réseau. Les neurones artificiels assignent ensuite un poids aux différents éléments. La couche finale de neurones va alors rassembler les différentes informations pour déduire s'il s'agit ou non d'un chat. Le réseau de neurones va ensuite comparer cette réponse aux bonnes réponses indiquées par les humains. Si les réponses correspondent, le réseau garde cette réussite et s'en servira plus tard pour reconnaître les chats. Dans le cas contraire, le réseau prend note de son erreur et ajuste le poids placé sur les différents neurones pour corriger son erreur. Le processus est répété des milliers de fois jusqu'à ce que le réseau soit capable de reconnaître un chat sur une photo dans toutes les circonstances.
- Apprentissage non supervisé : dans ce cas, les données ne sont pas étiquetées. Alors, les réseaux de neurones doivent reconnaître des patterns au sein des ensembles de données pour apprendre par eux-mêmes quels éléments d'une photo peuvent être pertinents.

3 – Apprentissage automatique et langue naturelle : le Plongement de mot (Word Embedding)

Le *word embedding* désigne un ensemble de techniques de *machine learning* qui visent à représenter les mots ou les phrases d'un texte par des vecteurs contextuels de nombres réels, décrits dans un modèle d'espace vectoriel (ou *Vector Space Model*).

Tout traitement automatique de la langue naturelle nécessite la représentation adéquate et pertinente des mots et/ou des phrases du corpus cible. Initialement, la communauté scientifique utilisait comme représentation : l'approche 'bag of words' soit bien sac de mots en français. Cette approche consiste à représenter les mots dans un vecteur dont la taille dépend de nombres de mots que contient le corpus. Un mot du corpus sera alors représenté par un vecteur dont toutes les coordonnées sont à 0 sauf la coordonnée qui correspond au mot cible (on met un 1). Cette représentation présente l'inconvénient d'être creuse (*sparse* en anglais). De plus, cette représentation ne tient pas compte du contexte du mot, ni de ses relations (sémantiques ou de co-occurrence) avec les autres mots du corpus. Pour pallier à ces inconvénients, de nombreuses approches ont vu le jour, comme par exemple l'approche par les n-grammes qui consiste à regrouper les mots d'une même phrase par couple (bi-grammes) ou plus(n-grammes), l'approche Tf-idf (*term frequency -inverse document Frequency)* qui tient compte de la relation d'un mot d'un document donné, aux autres documents du corpus… Toutes ces méthodes ont amélioré considérablement la représentation des mots, mais sans vraiment palier aux inconvénients principaux qui persistent toujours, et que le w*ord embedding* a pu résoudre.

Le *word embedding* repose sur la théorie linguistique fondée par ZELLIG Harris et connue sous le nom de *Distributional Semantics*. Cette théorie considère qu'un mot est caractérisé par son contexte, c'est à dire par les mots qui l'entourent dans une phrase. Ainsi, des mots qui partagent des contextes similaires partagent également des significations similaires. Ces mots peuvent alors être intégrés au sein d'un même mot qui les représente.

Les algorithmes de *word embedding* qui s'appuient sur des réseaux de neurones entrainés sur des gros corpus de données, sont le plus souvent employés pour décrire des mots à travers de vecteurs numériques denses, mais ils peuvent également être utilisés pour construire des représentations vectorielles de phrases entières.

Il existe plusieurs approches de *Word Embedding*. Les premières remontent aux années 1960 et reposent sur des méthodes de *réduction de dimensionnalité*[4]. Plus récemment de nouvelles techniques basées sur des modèles probabilistes et des réseaux de neurones, comme *Word2Vec*, ont permis d'obtenir de meilleures performances. Dans ce qui suit, nous nous intéressons au modèle Word2Vec.

3.1. Le modèle Word2Vec

Word2Vec est l'une des techniques les plus populaires du *word embedding* pour apprendre le contexte des mots à l'aide d'un réseau neuronal peu profond. Ce modèle a été développé par Tomas Mikolov en 2013 chez Google.

Word2Vec est un modèle prédictif particulièrement efficace sur le plan informatique pour l'apprentissage des plongées de mots à partir de texte brut. Il peut être obtenu en utilisant deux modèles (tous deux impliquant des réseaux de neurones) : *Skip-Gram* et *Common Bag Of Words* (CBOW). CBOW reçoit en entrée le contexte d'un mot, et essaye de prédire le mot en question. *Skip-Gram* fait exactement l'inverse : il prend en entrée un mot et essaye de prédire son contexte. Dans les deux cas, l'entraînement du réseau se fait en parcourant le texte fourni et en modifiant les poids neuronaux afin de réduire l'erreur de prédiction de l'algorithme (Voir figure 4) (Tomas Mikolov, 2013).

[4]*Réduction de dimensionnalité* : On désigne ainsi toute méthode permettant de projeter des données issues d'un espace de grande dimension dans un espace de plus petite dimension. Cette opération est cruciale en apprentissage automatique pour lutter contre ce qu'on appelle le fléau des grandes dimensions (le fait que les grandes dimensions altèrent l'efficacité des méthodes).

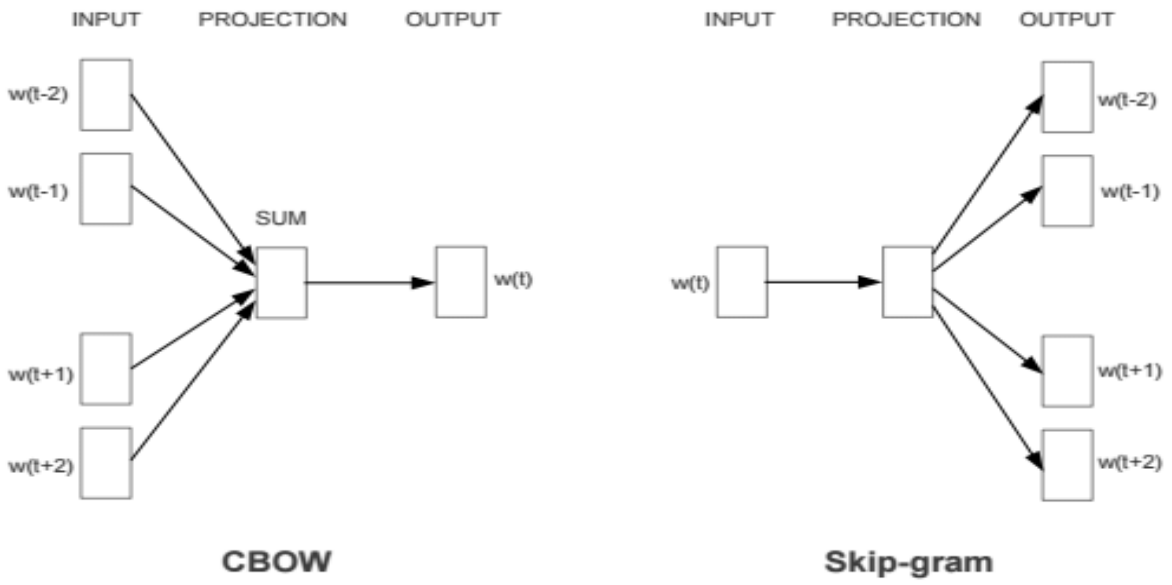


Figure 4 : CBOW vs Skip-Gram

Les méthodes Word2vec utilisent des perceptrons linéaires simples afin de calculer les vecteurs qui représentent les mots. Ces perceptrons sont organisés en 3 couches : une couche d'entrée, une couche de sortie et une couche intermédiaire (ou couche cachée).

- La couche d'entrée : elle prend un ensemble de vecteurs codés à chaud 'One Hot', qui représentent les mots du corpus.
- La couche de sortie : elle nous renvoie en sortie, un vecteur comportant la probabilité qu'un mot appartient au contexte du mot cible entré en couche d'entrée.
- La couche cachée : elle comporte un nombre fixe de neurones définies initialement. Chaque neurone représente une dimension du vecteur de sortie finale du mot cible.

L'idée de l'approche Word2vec est de compresser le corpus cible en un dictionnaire de vecteurs denses de dimension choisie très réduite.

Ces représentations vectorielles denses ainsi construites, possèdent des capacités surprenantes. Par exemple, on peut retrouver beaucoup de régularités linguistiques simplement en effectuant des translations linéaires dans cet espace de représentation. Par exemple le résultat de vec("man") - vec("King") + vec("woman") donne une position dont le vecteur le plus proche est vec("queen") (Voir figure 5).

Word2vec possède plusieurs implémentations en accès libre. Ces implémentations sont paramétrables. Les principaux paramètres sont décrits en annexe 1.

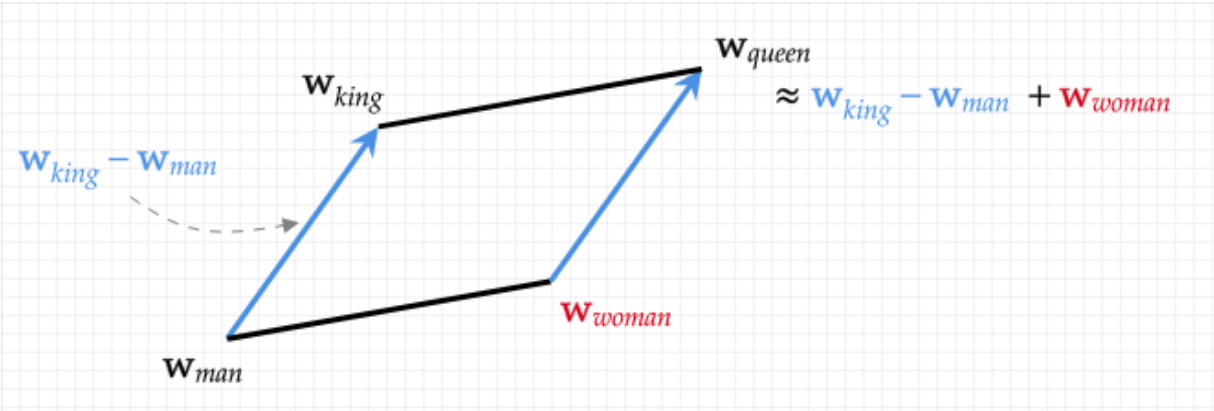


Figure 5 : Exemple 4 : Prédiction du mot approprié

Conclusion

Dans ce chapitre nous avons présenté le domaine de l'apprentissage automatique en général et du *deep learning* en particulier. Nous avons alors introduit les modèles d'apprentissage automatique en particulier ceux utilisés en traitement automatique de la langue naturelle, dont la technique du *word embedding* et son modèle word2Vec qui nous semble opportun d'utiliser dans le cadre de notre présent travail, qui a pour objet l'apprentissage automatique de l'arabe dialectal algérien.

Dans le chapitre suivant, nous ferons l'exposé des caractéristiques de l'arabe dialectalisé en général et de l'arabe algérien en particulier.

Chapitre II

Le dialecte arabe algérien

Introduction

Ce chapitre est consacré à la définition et à la présentation du dialecte algérien, on va commencer par une première section ou on parlera de l'histoire du dialecte algérien, ainsi que sa diversité est les influences des langues étrangères sur lui, et puis on passera à la deuxième section dans laquelle on expliquera la grammaire et le lexique du dialecte algérien, en passant par celle de l'arabe qui en est indissociable. Finalement on exposera les différents problèmes d'analyse et du traitement automatique du dialecte algérien.

1. Histoire du dialecte algérien

Le paysage linguistique de l'Algérie, produit de son histoire et de sa géographie, est caractérisé par la coexistence de plusieurs variétés langagières du substrat berbère aux différentes langues étrangères qui l'ont plus ou moins marquée, en passant par la langue arabe avec les conquêtes islamiques en Afrique du Nord.

L'Algérie est un pays plurilingue, riche de sa diversité, de ses références culturelles plurielles, majoritairement arabophone. La Constitution dispose l'arabe comme langue officielle. Cela ne désigne pas l'arabe dialectal algérien, appelé localement *darja*, mais l'arabe standard moderne (MSA). Mais depuis 2016, une révision de la Constitution algérienne ajoute l'article 4, reconnaissant le « tamazight » comme langue officielle. L'arabe dialectal est la principale langue véhiculaire utilisée par la population, Le tamazight et le français sont également répandus.

Le plurilinguisme, en Algérie, s'organise autour de deux principales sphères langagières (ZENNATI, 2020) (Kerras Nassima, 2019).

Sphère berbérophone :

Elle est constituée par les dialectes berbères actuels, prolongement des plus anciennes variétés connues dans l'aire berbérophone qui s'étend de l'Atlantique à l'Égypte, de la Méditerranée au fleuve Niger. Leurs locuteurs les désignent par le tamazight.

Le berbère était la langue maternelle de la population du Maghreb en général et de l'Algérie en particulier avant la conquête islamique, aujourd'hui il est la langue maternelle d'une partie de la population algérienne. Il intègre quelques mots arabes en raison des échanges commerciaux entre les populations locales d'Afrique du Nord et les arabes qui sont venus de l'Orient.

Ces parlers amazighs, comme on les dénomme maintenant, constituent le plus vieux substrat linguistique du pays et sont, de ce fait, la langue maternelle d'une partie de la population. Les principaux parlers amazighs algériens sont le kabyle ou taqbaylit (Kabylie), le chaoui ou tachaouit (Aurès), le mzabi (Mzab) et le targui ou tamachek des Touaregs du grand Sud (Hoggar et Tassili).

La sphère arabophone :

L'arabe est la langue du Coran et de l'islam. C'est la langue officielle utilisée dans tous les domaines de l'activité nationale tels que : administrations, entreprises publiques, information et enseignement où elle est enseignée à tous les niveaux du cursus scolaire. Elle est la plus étendue par le nombre de ses locuteurs. En premier lieu vient l'arabe fus'ha (ou arabe classique), ensuite l'arabe standard ou moderne, véritable langue d'intercommunication entre tous les pays arabophones, enfin les dialectes ou parlers qui se distribuent dans tous les pays en variantes locales et régionales.

L'arabe algérien (ou *darja*) noté AA, aussi appelé جزايري, *jazayriy* ou دزيري *dziyriy* signifiant simplement 'algérien', est la langue utilisée par la majorité de la population. C'est la principale langue véhiculaire d'Algérie. Elle est utilisée par 70 à 90 % de la population.

L'AA est considéré comme un langage de basse variété (faible variété). Ceci signifie que l'AA est faiblement normalisé et standardisé. Il est utilisé dans la presse, la télévision, la communication sociale, les échanges Internet, SMS, etc.
L'AA est un idiome qui a pour origine lexicale et grammaticale l'arabe principalement, mais il comprend aussi d'importants apports du berbère, du punique (carthaginois) et de manière plus relative du turc, de l'espagnol et du français. L'influence de ces langues sur l'AA diffère d'une région du pays à une autre, produisant ainsi différentes variétés linguistiques du dialecte. Ces variétés sont matérialisées par la présence de mots étrangers dans le dialecte et de systèmes de prononciation différents variant sensiblement d'une région à une autre : on retrouve ainsi, l'arabe algérois ou l'arabe bougiote influencés par le turc et le kabyle, l'arabe oranais présentant des mots d'origine ibérique influencé par le Zénète, l'arabe tlemcénien, ou encore l'arabe nedromi influencés par l'arabe andalou, l'arabe rural parlé à Constantine, Annaba ou Sétif, ou encore le saharien , dialecte parlé par la population du sud…

2. La grammaire et le lexique de la langue arabe

La grammaire et le lexique du dialecte algérien sont fortement liés à la grammaire et au lexique de la langue arabe, en effet le dialecte algérien se constitue à 95% de l'arabe mélangé avec des mots d'autres langues. Cela est dû aux différents colonisateurs qui ont colonisé l'Algérie par le passé.

Dans ce qui suit nous donnons une présentation globale du lexique et grammaire de la langue arabe, et puis on va mettre en avant les différences qui existent entre celle-ci et celle du dialecte algérien.

2.1. Eléments de base :

La grammaire de la langue arabe se compose de trois sous-ensembles : nom, verbes et particules. Ce classement a rapidement montré ses limites en ce qui concerne le traitement automatique de la langue à cause du peu d'ensemble qu'il contient. Pour pallier à cet inconvénient, d'autres sous-ensembles ont été rajoutés. Cette réorganisation a donné naissance à une autre classification se composant de quatre sous-ensembles : nom, verbe, pronom et les mots outils (MESFAR, 2008) et (SAADANE, 2015).

2.1.1. Les noms (الأسماء)

Un nom est un élément grammatical utilisé pour désigner un objet ou un être, indépendamment du temps. Les noms sont classés selon le système morphologique, en trois catégories :

i. **Les noms primitifs** (الأسماء الجامدة) : ce sont tous les noms qu'on ne peut pas produire à partir d'une racine verbale, comme les noms propres ou les noms communs. Par exemple : طاولة 'table' ou bien le prénom ' يوغرطة ', 'Jugurtha'.

ii. **Les noms dérivés ou déverbaux** (الأسماء المشتقة) **:** ce sont tous les noms qu'on peut dériver ou former à partir d'une racine verbale ; ils peuvent être sous plusieurs catégories grammaticales (participe actif, participe passif, les noms de lieu, les noms d'instruments, les noms d'une fois ...).

iii. **Les nombres** : cette catégorie contient les numéros simples qui représentent les unités comme (تسعة 'neuf').

2.1.2. Les verbes (الأفعال)

Un verbe est une entité qui exprime action ou bien un évènement dépendant du temps. Les verbes arabes sont généralement formés d'un radical à 3 consonnes comme (ضرب entrer) ou à 4 consonnes par exemple (دحرج). Ces radicaux peuvent donner naissance à d'autres verbes grâce à des transformations morphologiques comme le dédoublement d'une consonne ou l'allongement d'une voyelle. Ces dernières donnent lieu à ce qu'on appelle des racines à schéma augmentés. Les verbes sont classés en différentes catégories comme suit :

i. **Les verbes à racine simple (الفعل المجرد)** : Ce sont les verbes qui ont une base de 3 consonnes seulement. Ils suivent le schéma de verbe'فعل'. Si le verbe ne contient pas une variable longue dans l'une de ces consonnes, il est dit **verbe sain** (الفعل الصحيح), sinon c'est un **verbe altéré** (الفعل المعتل). Nous distinguons plusieurs cas selon le nombre de voyelles longues et leurs positions :

- **Verbe mahmouz (**مهموز**)** : si l'une des consonnes radicales est le glide " أ" (hamza), quelle que soit sa position dans le verbe.
- **Verbe assimilé (**مثال**)** : si la 1ère consonne radicale est le glide " و" (w-wâw) ou " ي" (y–yâ').
- **Verbe creux (**أجوف**)** : si la 2ème consonne radicale est " و" (w) ou " ي" (y).
- **Verbe défectueux (**ناقص**)** : si la 3ème consonne radicale est l'un des glides " و " (w) ou " ي" (y).
- La dernière catégorie est pour les verbes qui contiennent deux voyelles longues, soit ces deux voyelles longues sont ensemble (l'une après l'autre donc on l'appelle (اللفيف المقرون)) ou bien elles sont séparées par une consonne est on l'appelle (اللفيف المفروق).
- Par ailleurs, une autre classe de verbe existe et s'appelle verbe redoublé. On peut la distingue par la présence d'une consone redoublée au sien du verbe comme (رَدَّ).

ii. **Les verbes à racine augmentée (الفعل المزيد)** : Ce sont tous les verbes qu'on peut obtenir par l'application d'une des opérations citées auparavant (le redoublement, l'adjonction, l'allongement). Le tableau ci-dessus résume ces opérations :

Schéma verbal augmenté	Opération morphologique Associées	Exemple
فَعَّلَ	Redoublement de la deuxième consonne radicale	كرّر
فَاعل	Allongement de la première consonne radicale par l'ajout d'un 'ا'	واجه
أَفْعَلَ	Adjonction d'une 'أ' au début de la racine	أَيْقَضَ
تَفَعَّلَ	Adjonction d'une 'ت' au début de la racine + redoublement de la deuxième consonne radicale	تَقَدَّم
تفاعل	Adjonction d'une 'ت' au début de la racine + allongement de la première consonne radicale par l'ajout d'un 'ا'	تعاون
اِفتَعَلَ	Adjonction d'un 'ا' au début de la racine +insertion d'un morphème mono-consonantique 'ت' à la suite de la première consonne	اِفتَرس
اِنْفَعَلَ	Adjonction d'un morphème bi-consonantique 'ان" au début de la racine	اِنْقَلَبَ
اِسْتَفْعَلَ	Adjonction d'un morphème tri-consonantique 'است' au début de la racine	اِسْتَنْشَقَ

Tableau 1 : Les différentes opérations liées aux verbes

2.1.3. Les pronoms

Les pronoms sont des noms invariables. Leur flexion ne change pas quel que soit leur position dans la phrase, d'une autre manière cela veut dire qu'ils ne sont pas dérivables, mais dénombrables. Les pronoms se subdivisent en 3 catégories :

i. **Les pronoms personnels** : sont des pronoms utilisés pour désigner une personne qu'elle soit absente, auditrice ou locutrice. Les pronoms personnels à leur tour se divisent en 2 catégories :

- ***Les pronoms personnels isolés*** **:** ce sont des pronoms qui s'écrivent seuls dans la phrase. Le tableau 2 suivant en donne un récapitulatif.

- ***Les pronoms personnels collés* :** ce sont des pronoms qui sont liés au verbe ou au nom. Parmi ces pronoms on a :

- ‘هاء الغائب’ : en français peut désigner : sa, ses, son, …etc.
- ‘كاف الخطاب’ : peut désigner : ta, tes, ton, te, …etc.
- ‘ياء المتكلم’ : c’est pour désigner : ma, mes, mon, me, ...etc.
- ‘نون المتكلمين’ : pour désigner : nos, notre, nous, …etc.

Type de la personne	Genre		Pronom
La première personne	Singulier		أنا => je
	Pluriel		نحن => nous
La deuxième personne	Singulier	Masculin	أنت => tu
		Féminin	أنت => tu
	Duel		أنتما => vous
	Pluriel	Masculin	أنتم => vous
		Féminin	أنتن => vous
La troisième personne	Singulier	Masculin	هو =>il
		Féminin	هي=> elle
	Duel		هما => ils /elles
	Pluriel	Masculin	هم => ils
		Féminin	هن => elles

Tableau 2 : Les Pronoms personnels

ii. **Les pronoms relatifs :** Ce sont des pronoms isolés qui se placent souvent avant le verbe, pour désigner celui qui fait l’action (ou celui qui la subit). L’ensemble de ces pronoms est représenté dans le tableau 3 suivant :

	Nominatif	Subjonctif et génitif	Cas du nom …	
Pronom	الذي	الذي	Singulier	Masculin
	اللذان	اللذين	Duel	
	اللذين	اللذين	Pluriel	
	التي	التي	Singulier	Féminin
	اللتان	اللتين	Duel	
	اللاتي/اللائي	اللاتي/اللائي	Pluriel	

Tableau 3 : Les pronoms relatifs

iii. **Les pronoms démonstratifs** : il s'agit des pronoms isolés, qui se placent avant le nom pour le désigner et pour bien le définir. Ces pronoms sont divisés en 2 catégories

a) **Les pronoms démonstratifs de proximité** : utilisés pour désigner quelque chose de proche.

b) **Les pronoms démonstratifs d'éloignement** : utilisés pour désigner quelque chose de loin.

L'ensemble des pronoms démonstratifs est résumé dans le tableau 4 qui suit :

Caractéristique		Démonstratif de proximité		Démonstratifs d'éloignement	
Propre au lieu		هنا		هناك/هنالك	
Cas du nom		Subjonctif et	Nominatif	Subjonctif et	Nominatif
Masculin	Singulier	هذا	هذا	ذاك/ذلك	ذاك/ذلك
	Duel	هذين	هذان	ذينك	ذانك
	Pluriel	هؤلاء	هؤلاء	أولئك/أولائك	أولئك/أولائك
Féminin	Singulier	هذه	هذه	تلك	تلك
	Duel	هاتين	هاتان	تينك	تانك
	Pluriel	هؤلاء	هؤلاء	أولئك/أولائك	أولئك/أولائك

Tableau 4 : Les pronoms Démonstratifs

2.1.4. Les mots outils

Il s'agit d'un ensemble de mots utilisés pour situer des faits ou des objets par rapport au temps ou aux lieux. Ils servent également à garantir la cohérence et l'enchainement dans un texte. Les mots outils peuvent exprimer la conséquence, la cause le but et d'autres rapports entre phrases selon leur utilisation. Parmi ces derniers nous citons :

- Les prépositions : 'تحت', 'في'.
- Les particules : 'كيف', 'لم', 'لن'.
- Les conjonctions de coordination : 'ثم', 'ف', 'و'.
- Les conjonctions de subordination : 'حيثما', 'بينما'.
- Les quantificateurs : 'بعض', 'كل'.
- Les adverbes : 'أبدا', 'أخيرا'.

2.2. Règles grammaticales et conjugaison de la langue arabe :

2.2.1 Morphologie flexionnelle :

La flexion en linguistique et le fait d'appliquer des modifications sur une racine (un radical) afin de dénoter des traits grammaticaux souhaité. Toute langue utilisant ces opérations est dite langue flexionnelle. L'arabe appartient à cette catégorie et sa flexion se base sur l'ajout des suffixes et de préfixes pour exprimer plusieurs indices de mode, de temps, de personne et de genre.

Il existe deux classes de flexions, une pour le système nominal qui est la déclinaison et une pour les verbes qui est la conjugaison.

2.2.1.1. Flexion des verbes (conjugaison) :

La conjugaison en langue arabe est déterminée par plusieurs facteurs, chaque facteur influant sur la syntaxe du verbe, en y ajoutant des suffixes et des préfixes spécifiques :

- De temps (accompli, inaccompli)
- De nombre du sujet (singulier, duel, pluriel)
- De genre du sujet (féminin, masculin)
- De personne (première, deuxième et troisième)
- De mode (actif, passif)

En langue arabe, il existe 3 modes de conjugaison : l'accompli, l'inaccompli, et l'impératif. Chaque mode possède ses propre suffixes et préfixes.

i. **L'accompli (الماضي)** : ce mode fait référence à une action achevée, qui s'est déroulée dans le passé. Dans ce qui suit un exemple de conjugaison de verbe 'كتب' :

أنا	نحن	أنت	أنت	أنتما	أنتم	أنتن	هو	هي	هما	هما	هم	هن
كَتَبْتُ	كَتَبْنَا	كَتَبْتَ	كَتَبْتِ	كَتَبْتُمَا	كَتَبْتُمْ	كَتَبْتُنَّ	كَتَبَ	كَتَبَتْ	كَتَبَا	كَتَبَتَا	كَتَبُوا	كَتَبْنَ

Tableau 5 : Conjugaison du verbe 'كتب' à l'accompli

ii. **L'inaccompli (المضارع)** : exprime une action dans le présent ou bien dans le futur, l'action ne s'est pas encore écoulée. Ce mode possède plusieurs variantes :

- **Inaccompli indicatif** : il présente l'action énoncée comme certaines.
- **Inaccompli subjonctif(المنصوب) ou apocopé (المجزوم)** : ces deux paradigmes sont de mode potentiel (sauf pour les deux négations لَمْ et لَنْ). La voyelle finale 'الفتحة' caractérise le subjonctif et l'absence de voyelle finale ou 'السكون' caractérise l'apocopé. Le tableau suivant présente la conjugaison du verbe 'كتب' en subjonctif et en apocopé :

	أنا	نحن	أنْتَ	أنْتِ	أنتما	أنتم	أنتن	هو	هي	هما	هما	هم	هن
Apocopé	أَكْتُبْ	نَكْتُبْ	تَكْتُبْ	تَكْتُبِي	تَكْتُبَا	تَكْتُبُوا	تَكْتُبْنَ	يَكْتُبْ	تَكْتُبْ	يَكْتُبَا	تَكْتُبَا	يَكْتُبُوا	يَكْتُبْنَ
Subjonctif	أَكْتُبَ	نَكْتُبَ	تَكْتُبَ	تَكْتُبِي	تَكْتُبَا	تَكْتُبُوا	تَكْتُبْنَ	يَكْتُبَ	تَكْتُبَ	يَكْتُبَا	تَكْتُبَا	يَكْتُبُوا	يَكْتُبْنَ

Tableau 6 : Conjugaison du verbe 'كتب' à l'inaccompli

Notons qu'il en existe un autre mode qui exprime l'inaccompli futur, pour cela il suffit d'ajouter soit la conjonction 'س' ou bien 'سوف' à l'inaccompli indicatif.

Exemple : 'سوف أدخل' ou 'سأدخل'.

iii. **L'impératif(الأمر) :** Il est utilisé pour exprimer un ordre, une recommandation ou un conseil. Ce mode ne se conjugue qu'à la deuxième personne. Dans ce qui suit un exemple de la conjugaison du verbe 'كتب' à l'impératif.

أنت	أنت	أنتما	أنتم	أنتن
اُكْتُبْ	اُكْتُبِي	اُكْتُبَا	اُكْتُبُوا	اُكْتُبْنَ

Tableau 7 : Conjugaison du verbe 'كتب' à l'impératif

Remarque : Dans la langue arabe, les voyelles courtes sont d'une importance capitale pour reconnaitre le mode de la conjugaison du verbe.

2.2.1.2. Flexion des noms (déclinaison) :

En arabe, la déclinaison des noms se présente selon trois formes : le nominatif (المرفوع), l'accusatif (المنصوب), le génitif (المجرور). Ces déclinaisons sont faites selon le rôle du mot dans la phrase. Par exemple, le mot sera au nominatif s'il prend la notion du sujet, à l'accusatif s'il prend la fonction complément (المفعول به).

La déclinaison dépend de 3 critères : la forme du nom (simple ou diptote), le nombre (singulier, duel, pluriel) et le genre (féminin ou masculin) :

i. **Les déclinaisons au singulier :** On distingue :

- **Les nom définis par l'article 'ال' ou par une annexion (**معرف بالإضافة**)** : les suffixes sont comme suivis : 'الضمة' pour le cas nominatif, 'الفتحة' pour l'accusatif, 'الكسرة' pour le génitif.
- **Les noms indéfinis** : la déclinaison se fait par la nounatation (التنوين) par les trois signes, par exemple le nom 'درس' (une leçon) devient دَرْسًا. Pour l'accusatif la nounatation est associé avec un 'alif' sauf pour le cas où le nom possède un 'ة' à la fin.
- **Le diptote (**الإسم الجامد**)** : c'est un nom qui ne respecte pas les règles de déclinaison, et reste sous une même forme quelque soit sa position dans la phrase. Le tableau suivant recense les différentes catégories de diptotes :

Règle	Exemple
Noms propre féminins (العلم المؤنث)	'صارة', 'خديجة', 'هاجر'
Un nom propre masculin, mais se terminent par le	'أسامة' , 'حذيفة' , 'حمزة'
Adjectifs et couleurs de schémas 'أفعل'	'أخضر' , 'أحمر'
Adjectifs de schémas 'فعلان'	'كسلان' , 'عطشان'
Les noms propres étranges	'باريس', 'لندن'

Tableau 8 : les différentes catégories de diptotes

- **Les cinq noms (**الأسماء الخمسة**)** : c'est un ensemble de cinq exceptions qui se caractérisent par l'allongement de leur seconde syllabe lorsqu'ils sont définis par annexion (الإضافة). Ces mots sont : 'أب' (père), 'أخ' (frère), 'حم'(beau-père), 'فم' (bouche), 'ذو' (possesseur).

ii. Les déclinaisons au duel(المثنى) :

En arabe, il existe une catégorie, le duel **(المثنى)**, pour désigner un ensemble de deux choses ou de deux personnes.

La déclinaison au duel se fait par l'ajout du suffixe 'ان' pour le nominatif et par l'ajout du suffixe 'ين' pour l'accusatif et le génitif. Par exemple, le mot 'طفل' (enfant), devient 'طفلان' ou bien 'طفلين' selon la règle annoncée précédemment.

Il peut y avoir plusieurs cas où c'est nécessaire d'appliquer quelques modifications avant d'ajouter le suffixe 'ان' ou 'ين'. Par exemple, pour les mots qui se terminent par la lettre 'ة', il faut d'abord transformer la lettre 'ة' fermée à 'ت' ouverte (التاء المربوطة الى التاء المفتوحة) puis ajouter l'un des deux suffixes. Par exemple, le mot 'كرة' (ballon) devient 'كرتان' ou 'كرتين'.

iii.Les déclinaisons au pluriel (الجمع) :

Nous présentons dans ce qui suit les deux types de pluriel existants :

- **Le pluriel externe ou régulier :** Dans cette classe, on obtient le pluriel par l'ajout d'un suffixe (un pour le féminin et un pour le masculin) sans toucher la racine :

- Le pluriel externe masculin (الجمع المذكر السالم) : On obtient ce pluriel par l'ajout du suffixe 'ون' pour le nominatif et le 'ين' pour l'accusatif et le génitif. Par exemple, 'مهندس' (ingénieur) devient 'مهندسون' au nominatif, et 'مهندسين' à l'accusatif ou au génitif.

- Le pluriel externe féminin (الجمع المؤنث السالم) : On obtient le féminin pluriel régulier par l'ajout du suffixe 'ات', nous ajoutons la désinence 'ُ' pour le nominatif et 'ِ' pour le génitif, 'َ' pour l'accusatif. Par exemple, pour le mot 'سيارة'(voiture), on obtient :

Nominatif	سياراتُ
Accusatif	سياراتَ
Génitif	سياراتِ

Tableau 9 : Le pluriel du mot < سيارة >

- **Le pluriel interne (**جمع تكسير**) :** Contrairement au pluriel régulier, le pluriel interne subit des modifications majeures, qui, dans la plupart des cas restent imprévisibles. Il se transforme selon plusieurs schémas citons : (أفعلة), (أفعال), (فعلة), (أفعل), (فعول). Par exemple : 'قناع' (masque) => 'أقنعة', 'سهم' (flèche) => 'أسهم', 'سهل' (plaine) => 'سهول'.

2.2.2. Morphologie dérivationnelle :

De chaque racine verbale, on peut dériver plusieurs noms, c'est ce qu'on appelle la morphologie dérivationnelle. Le nombre de noms à dériver reste fortement lié au verbe, le nombre et la nature des formes dérivées varient selon le statut du verbe. L'ensemble de ces formes est résumé dans le tableau suivant :

Le nom	Définition	Exemple
Le nom verbal (اسم الفعل)	Ou aussi 'le nom d'action', c'est un nom dérivé du verbe qui exprime une action, un verbe peut avoir plusieurs non d'action.	Le verbe 'وَدَّ' (aimer) a plusieurs noms d'action : 'مودة', 'وِدٌّ'.
Le participe actif (اسم الفاعل)	C'est nom qui désigne celui qui fait l'action.	Le verbe 'ضرب' (frapper) a le participe actif 'ضارب'.
Le participe passif	C'est un nom qui désigne celui qui subit l'action.	Le verbe 'ضرب' (frapper) a le participe passif 'مضروب'.
Le nom de lieu (ou de temps) (اسم المكان و الزمان)	C'un nom dérivé à partir du verbe, il désigne le lieu, le temps (respectivement).	Le verbe 'غَرَبَ' (se coucher) a respectivement le nom du lieu et de temps suivant 'مغرب', 'غَرْبْ'.
Le nom d'instrument (اسم الآلة)	Il exprime l'outil avec lequel on fait l'action.	Le verbe 'ضرب' (frapper) a le nom d'instrument 'مضرب'.
Le nom d'une fois	Il désigne une occurrence unique de l'action exprimé.	Le verbe 'ضرب' (frapper) a le nom d'une fois suivant 'ضربة'.
Le nom de manière (اسم الحالة)	Un nom qui désigne la manière dont l'action exprimé.	Le verbe 'جلس' (s'asseoir) a le nom d'une fois suivant 'جلسة'.

Tableau 10 : Ensemble des formes de Dérivation

3. La grammaire et le lexique du dialecte algérien :

Nous avons vu dans la section précédente la grammaire et le lexique de la langue arabe. Le dialecte algérien dérive de l'arabe, son lexique et sa grammaire dépendent fortement de ceux de la langue arabe. Dans cette section, nous présentons la grammaire et le lexique du dialecte algérien sous forme d'une comparaison avec l'arabe **fus'ha**, où nous mettons le point sur leurs différences. Cette synthèse est basée sur les travaux de (Kerras Nassima, 2019), (Houda, 2015).

1- Variations morphologiques :

1.1. Changements qui touchent toutes les structures :

1- La première différence à constater dans le dialecte algérien c'est la disparition des différentes terminaisons dans les noms et les verbes (pas de désinences à la fin du mot comme 'ُ' (الضمة), 'َ' (الفتحة), 'ِ' (الكسرة)). Ainsi, dans le dialecte algérien on n'aura pas de nominatif, d'accusatif, ou de génitif en ce qui concerne les noms. En ce qui concerne les verbes, seul l'indicatif est utilisé, les autres modes ne sont pas utilisés.

2- Le dialecte algérien ajoute plusieurs nouveaux clitiques[5] qui n'existent pas dans l'arabe, comme par exemple la négation circonfixe : 'ما' + 'verbe' + 'تش' => ما قريتش.

3- Pour la forme 'استفعل', le dialecte algérien introduit une autre variante, qui est 'سفعل' ou bien 'ستفعل' par exemple : le verbe 'استكلف' (prendre en charge) devient 'سكلف' ou bien 'ستكلف'.

1.2. Changements qui touchent seulement certaines structures :

1.2.1. Verbes :

A- On peut constater à première vue, que le duel (masculin et féminin) ainsi que le pluriel féminin ne sont pas présents, ils sont tous représentés par le pluriel masculin.

En outre, la première et la deuxième personne du singulier sont conjuguées de la même façon dans le dialecte.

Exemple : comparaison entre le passé, le présent, et l'impératif entre l'arabe **fus'ha** et l'algérien :

[5] Un clitique, en linguistique, est un élément à mi-chemin entre un mot indépendant et un morphème lié.

-Le passé

Pronom	La langue arabe	Le dialecte algérien
أنا	كَتَبْتُ	كْتبتْ
نحن	كَتَبْنَا	كْتبنا
أنتَ	كَتَبْتَ	كْتبتْ
أنتِ	كَتَبْتِ	كْتبتِ
أنتما	كَتَبْتُمَا	/
أنتم	كَتَبْتُمْ	كْتبتوا
أنتن	كَتَبْتُنَّ	/
هو	كَتَبَ	كْتبْ
هي	كَتَبَتْ	كْتبتْ
هما	كَتَبَا	/
هما	كَتَبَتَا	/
هم	كَتَبُوا	كْتبوا
هن	كَتَبْتُنَّ	/

-Le présent

Pronom	La langue arabe	Le dialecte algérien
أنا	أَكْتُبُ	نكتبْ
نحن	نَكْتُبُ	نكتبو
أنتَ	تَكْتُبُ	تكتبْ
أنتِ	تَكْتُبِينَ	تكتبي
أنتما	تَكْتُبَانِ	/
أنتم	تَكْتُبُونَ	تكتبو
أنتن	تَكْتُبْنَ	/
هو	يَكْتُبُ	يكتبْ
هي	تَكْتُبُ	تكتبْ
هما	يَكْتُبَانِ	/
هما	تَكْتُبَانِ	/
هم	يَكْتُبُونَ	يكتبو
هن	يَكْتُبْنَ	/

-l’impératif

Pronom	La langue arabe	Le dialecte algérien
أنتَ	اُكْتُبْ	كْتَبْ
أنتِ	اُكْتُبِي	كْتبي
أنتما	اُكْتُبَا	/
أنتم	اُكْتُبُوا	كْتبو
أنتن	اُكْتُبْنَ	/

Tableau 11 : Comparaison de la conjugaison du verbe « كَتَبَ» entre l’Arabe fusha et AA

B- La forme passive en dialecte algérien diffère de celle de la langue arabe.

En effet, la voix passive en langue arabe se fait par le changement des voyelles courtes du verbe. Par exemple, le verbe 'كَتَبَ' s'écrit 'كَتَبَ' en forme active, et devient 'كُتِبَ' en forme passive.

Tandis qu'en dialecte algérien, la voix passive se fait par l'ajout d'un des morphèmes (préfixes) suivants selon le verbe :

- la lettre 'تَ' exemple 'بنى' devient 'تَبنى'.
- la lettre 'ن' exemple 'فتح' devient 'نفتح'.
- le suffixe 'نت' ou bien 'تن' exemple 'أكل' devient 'نتكل' et 'قتل' devient 'تنقتل'.

C- Le dialecte algérien introduit la voyelle 'ي' entre la racine et les suffixes consonantique de la forme perfective du verbe géminé primaire, chose qui est inexistante dans la langue arabe fus'ha.

Par exemple, le verbe شدَّ (tirer) en langue arabe devient comme ceci quand on le conjugue ' أنا شددت' par contre en dialecte algérien il devient ainsi 'أنا شديت'.

D- Le dialecte algérien utilisent aussi des verbes de la langue française et les conjuguent avec les terminaisons de la langue arabe, exemple le verbe 'bouger' en français devient comme ceci en arabe :

- أنا بوجيت
- نتومابوجيتو
- هوما بوجاو-

1.2.2. Noms : En dialecte algérien, la flexion des noms différent un peu de celle en arabe :

A- En dialecte algérien seul le suffixe 'ين' est utilisé pour exprimer le pluriel régulier, et il élide l'avant dernière consonne avant d'ajouter le suffise 'ين'. Exemple pour le mot 'مهندس' en arabe il peut devenir soit 'مُهَنْدِسُونَ' soit 'مُهَنْدِسِينَ' mais en dialecte il existe seulement une transformation 'مُهَندْسِينْ'.

-Le pluriel irrégulier quant à lui est très diffèrent de l'arabe, pour quelques mots il change carrément et il reste le même pour quelques-uns (Ibrahimi, 2010).

Exemple :

Le mot en singulier	Pluriel en arabe	Pluriel en algérien	Traduction
باب	أبواب	بيبان	Portes
نافذة	نوافذ	تيقان	Fenêtres
كبير	كبر	كبار	Grands
بنك	بنوك	بنكات	Banques
حوت	حيتان	حوت	Baleines
أزق	أزقة	زنق	Rues
حذاء	أحذية	صبابط	Chaussures

Tableau 12 : La flexion des noms en arabe fusha et en AA

- À noter qu'il existe plusieurs mots qui n'existent pas dans l'arabe et qui sont présents dans le dialecte algérien cela est dû au lexique différent qu'on expliquera en détail dans la section qui suit.

Le dialecte algérien utilise en général le mot 'زوج' pour exprimer la forme duelle nominale contrairement à l'arabe ; par exemple pour dire deux livres en arabe 'كتابين' mais en dialecte algérien on dit 'زوج كتب'.

B- En ce qui concerne l'ensemble des noms dérivés d'une racine verbale il y'en a une forte présence de la langue française. Par exemple pour dire une gomme en arabe, on dit 'ممحاة' par contre en dialecte la plupart disent 'gomme'.

1.2.3. Pronom :

Les pronoms en dialecte algérien sont généralement différents de ceux de la langue arabe, dans ce qui suit on présentera des tableaux qui résument les différences existantes :

-Les pronoms démonstratifs :

A- Démonstratif de proximité :

Pronom	Langue arabe	Dialecte algérien
Singulier	هذا	هذا
	هذه	هذي
Duel	هذان	/
	هذين	/
	هتان	/
	هتين	/
Pluriel	هؤلاء	هذو

Tableau 13 : Comparaison des pronoms démonstratifs de proximités entre l'arabe fusha et AA

B- Démonstratif d'éloignement :

Pronom	Langue arabe	Dialecte algérien
Singulier	ذلك	هذاك
	تلك	هذيك
Duel	ذانك	/
	ذينك	/
	تانك	/
	تينك	/
Pluriel	أولئك	هذوك

Tableau 14 : Comparaison des pronoms démonstratifs d'éloignements entre l'arabe fusha et AA

- Les pronoms personnels :

A- Les pronoms personnels isolés :

Ils sont, presque tous, similaires à ceux de la langue arabe, excepté quelque uns :

Le tableau montre les différences existantes :

Arabe	أنا	نحن	أنتَ	أنتِ	أنتما	أنتم	أنتن	هو	هي	هما	هم	هن
Algérien	أنا	حنا	نتَ	نتِ	/	نتوما	/	هو	هي	/	هوما	/

Tableau 15 : Les pronoms personnels isolés en AA

B- Les pronoms personnels collés :

Ils sont les même que ceux de la langue arabe à part ceux de la forme duelle et de la forme pluriel féminin qui sont remplacées par celui du pluriel masculin.

Exemple : كتبتها, كتبتهم

- les pronoms relatifs :

Le tableau suivant recense les différences existantes :

Pronoms relatif	La langue arabe	Le dialecte algérien
Singulier	الذي	اللي
	التي	اللي
Duel	اللذان	/
	اللذين	/
	اللتان	/
	اللتين	/
Pluriel	اللذين	اللي
	اللاتي/اللواتي	اللي

Tableau 16 : Comparaison des pronoms relatifs entre l'arabe fusha et AA

1.2.4. Les mots outils :

A- Les particules : Mots invariables de la langue, elles sont totalement différentes en dialecte algérien, parmi ces dernières on a les interrogatifs qui sont résumés dans le tableau suivant :

La langue arabe	Le dialecte Algérien
من 'qui'	شكون
كيف 'comment'	كيفاش
ماذا 'quoi'	واش
لماذا 'pourquoi'	علاش
أين 'ou'	وين
متى 'quand'	وينت

Tableau 17 : Les mots outils en AA

B- Les adverbes : en dialecte algérien les adverbes sont parfois similaires et d'autres fois distincts de ceux de la langue arabe. Par exemple : en arabe on dit 'اول امس', tandis qu'en algérien on dit 'لولبارح'.

C- Pour ce qui reste des mots outils comme les conjonctions de coordination ainsi que les prépositions, ils sont généralement les mêmes, sauf pour quelques-uns qui ne sont jamais utilisés en dialecte comme la préposition 'بل'.

2- Variations lexiques :

Le dialecte algérien influencé par plusieurs facteurs cités précédemment, présente une différence importante comparé à celui de la langue arabe.

En plus des mots de langue arabe, on peut compter plusieurs mots empruntés de plusieurs langues étrangères ou bien formés par l'ajout de plusieurs suffixes et préfixes spécifiques à ce dialecte.

Le lexique représente alors l'axe où les deux langues divergent le plus.

Dans ce qui suit nous allons détailler les différences sur les deux axes : la dérivation et l'emprunt.

a- La dérivation :

Pour les dialectes, la régularité de la dérivation constitue la colonne vertébrale du système morphologique dialectal (SAADANE, 2015).

Le dialecte algérien comme les autres dialectes qui se dérivent de la langue arabe, possède des schémas de dérivation qui sont similaires à ceux de la langue arabe, néanmoins il est enrichi par un système affixale propre à lui.

En dialecte algérien on trouve le suffixe 'جي' qui indique la profession (Baccouche,1994) comme par exemple à partir du mot 'قهوة' (café) on obtient le nom de la profession 'قهواجي', on utilise un type de dérivation inexistant dans la langue arabe et cela se fait en combinant des schémas de dérivation aux affixes.

Il existe aussi des suffixes empruntés de la langue française comme le suffixe 'ist' qui s'ajoute à un mot afin d'exprimer une profession par exemple pour le mot 'حيط'on obtient 'حيطست'ce qui fait référence de manière ironique à un chômeur.

b- L'emprunt :

L'emprunt prend une place importante dans le lexique algérien, et cela est dû à plusieurs facteurs historiques cités précédemment, ce qui explique l'énorme divergence qui s'est instauré entre lui et la langue arabe (GUELLA, 2011).

Selon (SAADANE, 2015), il existe trois points à retenir en ce qui concerne l'emprunt :

-1 L'introduction de nouveau suffixes empruntés des autres langues, à l'exemple des deux suffixes cités précédemment, le suffixe turc 'جي' et le suffixe français 'يست' qui servent tous les deux à exprimer la profession.

-2 L'intégration systématique des unités empruntées dans les paradigmes construits par des schémas : À partir du mot emprunté, on peut utiliser la morphologie dérivationnelle pour dériver plusieurs autres mots. Par exemple, à partir mot 'business' emprunté de la langue anglaise, on peut dériver ce qui suit :

-le verbe 'بزنس' (faire du business).

-le participe actif 'بزناس' (celui qui fait du business).

-le nom d'action 'تبزنيس'.

Nous concluons cette partie par un tableau qui regroupe quelques mots empruntés des autres langues, et utilisés largement dans le dialecte algérien :

Mots	Traduction	Origine
فكرون	Tortue	Berbère
شلاغم	Moustache	
قرجومة	Gorge	
زرمومية	Lézard	
طبسي	Assiette	Turque
سكارجي	Ivrogne	
تقاشير	Chaussette	
زردة	Festin	
فيشطة	Fête	Italien
سوردي	Money	
زبلة	Faute	
سيمانة	Semaine	Espagnol
طابلة	Table	Français
تيليفون	Téléphones	
فرملي	Infirmier	

Tableau 18 : Les différents mots empruntés des autres langues

- À noter que les mots français que les algériens utilisent dans leur quotidien comme ‘salon’, ‘gomme’, ou bien d’autres mots, constitue un autre phénomène bien distinct qu’on trouve beaucoup plus chez les étudiants et dans le monde professionnel, ces derniers ont tendance à mélanger le dialecte algérien et la langue française et ceci est totalement diffèrent de l’emprunt. Cela s’appelle l’alternance codique (le code switching).

3- Variations syntaxiques :

Syntaxiquement, l’écart entre le dialecte algérien et la langue arabe est très marquant, et cela est dû en grande partie à la disparition des marqueurs flexionnels :

Les cas nominatif, accusatif, et génitif pour les noms, et la perte de la distinction entre l’indicatif, le subjonctif et l’impératif pour les verbes. Ceci pose un problème pour définir les différentes unités lexicales dans une phrase donnée.

Pour bien illustrer ces propos, on va s’appuyer sur ces différents exemples qui montrent bien les écarts existants :

Prenant cette phrase : ضرب الرجل الطّفل

En arabe cette phrase peut désigner plusieurs sens, selon la terminaison du verbe, comme suit :

Premier sens :

- ضربَ الرجلُ الطّفلَ => (l’homme a frappé l’enfant).

Deuxième sens :

- ضرب الرجلَ الطّفلُ => (l’enfant a frappé l’homme).

Par contre, au dialecte algérien cette phrase n’a qu’un seul sens, donc on ne peut pas inverser entre le sujet et le complément (ما بين الفاعل والمفعول به).

c) ضربْ الراجلْ الطّفل => (l’homme a frappé l’enfant).

4. Les problèmes liés au traitement automatique du dialecte algérien :

Le traitement automatique de l’arabe algérien fait face à plusieurs problèmes, pour bien les résumer, on va les classer selon deux grandes classes :

1- les problèmes hérités de la langue arabe :

Le dialecte algérien se dérive principalement de la langue arabe, ce qui fait que les problèmes rencontrés dans le traitement de la langue arabe vont aussi être rencontrés par le dialecte algérien, donc dans ce qui suit nous allons voir les problèmes liés à la langue arabe et leur influence sur le dialecte algérien.

a. L’absence de voyelles -voyellation-

L’absence des voyelles de la langue arabe qui sont remplacées par les voyelles courtes ou bien les diacritiques (ً) (ُ) (ِ) peut être source d’ambiguïté.

Pour bien comprendre ceci, on va étudier cet exemple :

On va prendre le radicale ‘علم’, ce radical peut avoir plusieurs formes grammaticales et plusieurs sens selon les diacritiques (علامات التشكيل) présentes sur lui.

عَلِمَ=>verbe (savoir) conjugué à l’accompli (forme active).

عُلِمَ=> verbe (savoir) conjugué à l’accompli (forme passive).

عَلَّمَ=>verbe (enseigner) conjugué à l'accompli (forme active).

عُلِّمَ=> verbe (enseigner) conjugué à l’accompli (forme passive).

عَلَمٌ=> nom (drapeau).

عِلْمٌ=>nom (le savoir).

En dialecte algérien ça reste moins répandu que la langue arabe à cause du lexique différent, ainsi que la façon différente de représenter des catégories grammaticales, par exemple la forme

passive qui est exprimé par l'ajout d'un préfixe contrairement à l'arabe : Ainsi le verbe 'فتح' en forme passive devient 'تنفتح'. Ceci ne résout pas le problème, il diminue seulement la probabilité d'avoir une ambigüité.

À noter aussi que le dialecte algérien possède des complications inexistantes en langue arabe comme par exemple la conjugaison des verbes qui est identique entre la première personne du singulier 'أنا' est la deuxième personne du singulier 'أنت'.

=> أنا كليت

=> أنت كليت

- Ceci comme on le voit cause plusieurs d'ambigüité lexicale comparable à celle causée par les mots non accentués en français, comme le mot 'eleve'. Il peut être interprété comme *élève* (nom masculin ou Verbe, Présent de l'indicatif, Voix active, 1ère et 3ème personne, masculin/féminin, au singulier ou Verbe, Présent de l'impératif 2ème personne), ou *élevé* (Adjectif masculin ou participe passé du verbe 'élever').

b. Agglutination

La langue arabe est une langue fortement agglutinée, c'est-à-dire qu'on peut ajouter plusieurs préfixe ou suffixe pour exprimer plusieurs fonctions dans la phrase telle que le sujet, le complément (مفعول به) et plusieurs autres fonctions.

Cette image montre la puissance de l'agglutination en arabe :

Figure 6 : Exemple d'agglutination en langue arabe

Le dialecte algérien ne dispose pas de ce genre de problème, l'agglutination est moins forte que la langue arabe, elle se restreint aux pronoms possessifs. Par exemple :

كوارطها => ses => ها , papier => كوارط.

c. Irrégularité de l'ordre des mots dans la phrase :

La langue arabe et le dialecte algérien possèdent tous les deux un système de phrase très libre, on peut construire des phrases en inter-changeant des mots sans que le sens soit altéré.

Exemple :

- Verbe + sujet + complément :

تأهلت الجزائر إلى كأس العالم (- L'Algérie s'est qualifiée pour la coupe du monde)

- Sujet + verbe + complément :

الجزائر تأهلت إلى كأس العالم (- C'est l'Algérie qui s'est qualifiée en coupe du monde)

- Complément + verbe + sujet

إلى كأس العالم تأهلت الجزائر (- C'est pour la coupe du monde que l'Algérie s'est qualifiée).

Ceci montre les ambiguïtés syntaxiques rencontré lors d'un traitement automatique. En effet du moment que la structure n'est pas simple, on ne peut pas se prononcer sur le sens de la phrase sans étudier toutes les possibilités.

2- Les problèmes liés seulement au dialecte algérien :

a. L'absence de norme pour l'écriture du dialecte :

Comme le dialecte algérien est un moyen de communication spécifique à l'Algérie, ce dernier ne possède pas une académie qui fixe ses règles d'écriture. Il hérite de l'arabe en ce qui concerne la grammaire et le lexique, mais en ce qui concerne l'orthographe, il y a une absence totale de règles d'écriture et ça se remarque fortement sur les réseaux sociaux. Par exemple le prénom 'عمر' peut s'écrire de différente façon comme 'عومار' 'عمار' 'عومر'.

b. L'écriture du dialecte algérien en lettres latines :

On peut écrire le dialecte algérien avec les lettres latines. Il s'agit là d'une pratique très répandue dans les réseaux sociaux, surtout chez les étudiants. À titre d'exemple :

'Rani rayeh l la fac neqra cour' est une phrase écrite en dialecte algérien avec des lettres latines qui veut dire 'je pars à l'université pour assister à un cours'.

Par ailleurs, une pratique courante dans les réseaux sociaux consiste à utiliser des chiffres pour exprimer plusieurs lettres du dialecte algérien, comme par exemple '7=>h', ou bien 9=>'q' donc le mot =>'حمامة' peut s'écrire 'hamama' ou bien '7amama'.

Il existe aussi la technique d'abréviations qui est très utilisés lorsque les gens, envoient des messages, font des publications, …etc. Par exemple pour dire à quelqu'un 'Bonne nuit' ils écrivent tout simplement :'BN8'.

5. Conclusion

Au cours de ce chapitre, nous avons présenté le dialecte algérien, en commençant par l'étude de son histoire ainsi que l'exposition des différentes sphères ou parties qui forment ce dialecte. Ensuite nous sommes passés à l'étude détaillée de la grammaire et du lexique de ce dernier, puis nous avons exposé les différents problèmes liés au traitement automatique du dialecte algérien.

Dans le chapitre suivant nous présenterons les différentes approches existantes pour le traitement automatique de l'arabe dialectalisé qui vont nous aider à trouver la meilleure approche de traitement automatique du dialecte algérien.

Chapitre III

Traitement automatique de l'arabe dialectalisé :

Etat de l'Art

Introduction

Après avoir découvrir lors du chapitre précédent l'histoire du dialecte algérien ainsi que les différents problèmes liés à son traitement automatique, dans ce chapitre, nous exposerons un ensemble d'études récentes sur l'analyse et la compréhension du dialecte arabe algérien. Le traitement automatique de la langue arabe n'est pas nouveau, il a fait objet de plusieurs études et de plusieurs recherches. Ces études se sont essentiellement concentrées sur l'arabe standard (العربية الفصحى) laissant de côté les dialectes et les phénomènes liés à l'usage dialectal de la langue arabe. La prolifération de rédacteurs de blogs sur Internet et les contributions diverses et variées sur les forums de discussion en ligne a fait apparaître des usages langagiers de l'arabe standard fortement teintés de dialecte local, ou mixés avec une langue étrangère comme le français ou l'anglais, ou encore directement transcrits en lettres latines, ce qui nous conduit à nous poser des questions par rapport à l'état de la recherche en la matière. Dans ce qui suit, on s'intéressera à la méthodologie de traitement des dialectes arabes et spécialement le dialecte algérien.

1. Aperçu sur les études existantes :

L'étude du dialecte algérien à des fins d'informatisation est un domaine récent. Les quelques études que nous avons pu trouver dans la littérature datent de moins d'une dizaine d'années. Cet état de l'art focalisera sur 3 études récentes phares :

La première étude (SAADANE, 2015) intitulée « Traitement automatique de l'arabe dialectisé : aspects méthodologique et algorithmique est le résultat d'une thèse de doctorat soutenue publiquement le 14 décembre 2015 à l'université de Grenoble Alpes. Il s'agit d'une étude complète et approfondie de la morphologie dialectale de la langue arabe qui propose en outre, des algorithmes et méthodes pour traiter automatiquement l'arabe dialectisé.

La deuxième étude (Guellil Imane, Azouaou Faical, 2006) traite de la traduction automatique du dialecte algérien ainsi que l'analyse de sentiments dans les messages rédigés dans ce dialecte.

La troisième étude (Guellil Imane, 2018) est une approche hybride pour la translitération de l'arabizi algérien a l'aide d'un ensemble de règles permettant le passage de l'*arabizi* vers l'arabe.

1. Les travaux de (SAADANE, 2015):

L'auteur propose une approche de traitement de la langue arabe et explique chaque étape de cette dernière. (Cette méthode résout tous les problèmes, obstacles soulevés dans la première partie). Cette approche se résume dans le schéma suivant :

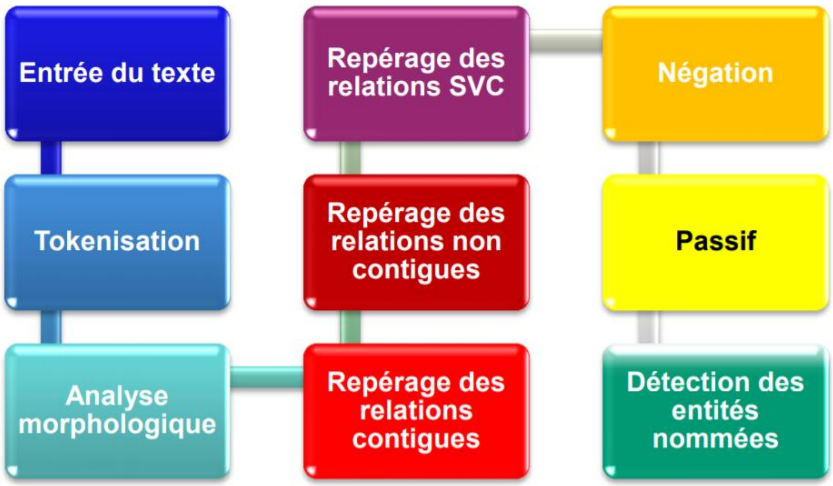


Figure 7 : Les étapes de l'analyse linguistique.

– Dans cette approche, l'auteur commence par la tokenisation du texte. La tokenisation c'est le fait de découper le texte en unité lexicales (segmentation) en se basant sur un ensemble de séparateurs (espace, ponctuation, …). Cette étape est nécessaire préalablement à tout traitement.

Pour bien illustrer cette étape, prenons comme exemple la phrase suivante en entrée :

'كل وعاء يضيق بما جعل فيه إلَ وعاء العلم؛ فإنه يتسع به'

Après l'étape de tokenisation, on a le résultat suivant :

'كل', 'وعاء', 'يضيق', 'بما', 'جعل', 'فيه', 'إلَ', 'وعاء', 'العلم', 'فإنه', 'يتسع', 'به'.

– Après cette étape, on passe à l'analyse morphologique. L'analyse morphologique est une étape qui consiste à traiter l'unité lexicale (le mot) obtenu à partir de la tokénisation afin de détecter le rôle et la structuration du mot. Dans l'analyse morphologique chaque mot est traité séparément (on verra un peu plus tard qu'il existe des relations entre les mots de la même phrase mais cela est de ressort de l'analyse syntaxique).

Dans cette analyse morphologique, chaque mot va être comparé à plusieurs mots sous forme canonique (infinitifs du verbe, singulier du nom …), présents dans un dictionnaire, afin de déterminer la fonction du mot traité en ce qui concerne sa catégories grammaticale (verbe, nom, particule ...) et ses traits morphologiques (genre, nombre, …). Par exemple si on prend le mot ‘شربتموها’, après analyse morphologique on va être capable de savoir que c’est le verbe ‘شرب’ conjugué avec le pronom ‘vous’ ‘أنتم’ et aussi que ces derniers ont bu quelque chose de féminin grâce au pronom ‘ها’.

Ceci bien évidemment passe par plusieurs d’étapes :

**La première étape de l’analyse morphologique est la segmentation des formes agglutinées. Une forme agglutinée est constituée d’une racine (lemme) à qui on ajoute des clitique (des pronoms ou bien des conjonctions attachées au mot). L’analyse des formes agglutinées se fait selon le processus décrit par le schéma suivant :

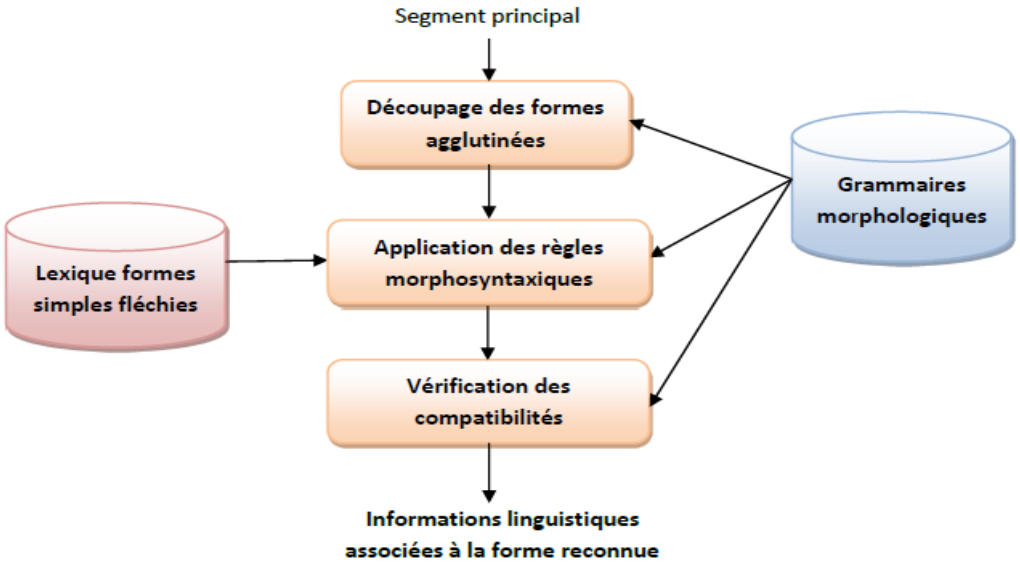


Figure 8 : Les étapes de l’analyse morphologique.

Pour commencer on recherche toutes les formes de clitiques qui peuvent être attaché au mot en utilisant un dictionnaire, après quoi on lance une recherche dans un dictionnaire à part afin de trouver la racine : si cette dernière est inexistante, des transformations morphologiques seront appliquées. Par exemple, pour le mot ‘بسيارته’, après extraction des clitiques ‘ب’ ‘ه’, on trouve la racine ‘سيارت’ qui est inexistante dans le dictionnaire, donc on applique une transformation morphologique afin qu’on le retrouve, donc le mot devient ‘سيارة’.

**L’étape suivante est la désambiguïsation. Cette étape sert à classer chaque mot selon une catégorie grammaticale bien définie.

- Une fois l’analyse morphologique terminée, on passe à l’analyse syntaxique. L’analyse syntaxique est très importante dans le traitement automatique des langages. En effet, pour pouvoir traiter une phrase, il ne suffit pas de comprendre ses mots individuellement, car la phrase à un sens global et chaque mot complète un autre mot de la même phrase.

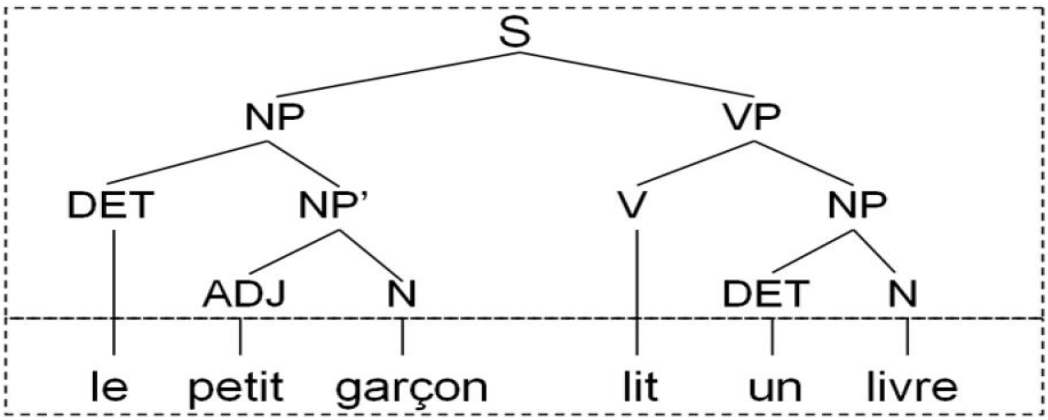


Figure 9 : Exemple montrant Les relations existantes entre les mots dans une phrase.

En effet, la majorité des linguistiques s’accordent sur le fait que les mots d’une phrase ne sont pas disposés de façon aléatoire, bien au contraire, ces derniers suivent un système d’organisations bien définies. La théorie la plus intéressante sur laquelle les linguistiques se basent c’est l’arbre de dépendance qui permet de mettre en avant les relations entre les mots d’une phrase. L’analyse syntaxique intervient pour expliciter ces relations entre mots dans une phrase

- La dernière étape du traitement automatique de la langue naturelle est la reconnaissance des entités nommées. Pour bien expliquer le concept des entités nommées, on s’appuie sur la classification faite lors de la conférence MUC-6.

- **NAMEX** : cette classe contient les noms propres de personne comme ‘كريم بلقاسم’ ‘Krim Belkacem’, d’**Organisation** comme par exemple يُونِيسْكُو ‘Unesco’, ou de l**ocalisations** de pays, villes, états, mers, océans, montagnes, fleuves, etc. Par exemple, الجَزَائِر. ‘Algérie’, بَارِيس ‘Paris’.

- **NUMEX** : contient les entités formalisées dans des expressions numériques de pourcentage, taille, expressions monétaires, etc.

- **TIMEX** : concerne les entités exprimant le temps, la date ou une durée.

En plus, comme il n'existe pas de norme commune ni de stratégie unifiée pour la transcription automatique du dialecte. La communauté scientifique, ont développé une convention d'écriture nommée CODA (convention orthographique des dialectes, la translittération des noms arabes en écriture latine et inversement.)

2. Les travaux de (Guellil Imane, Azouaou Faical, 2006) :

Dans ce document, les auteurs se sont intéressés à l'analyse des sentiments sur les réseaux sociaux, à partir de messages rédigés dans le dialecte arabe algérien. Pour ce faire, les auteurs ont d'abord construit un corpus initial (une sorte de corpus d'entrainement) issu de divers documents collectés à partir des réseaux sociaux. L'objectif est d'utiliser ce corpus pour apprendre le dialecte arabe algérien et ses caractéristiques.

Les auteurs ont procédé à une étude de l'ensemble des mots utilisés dans le corpus d'entrainement afin de construire un modèle de termes du dialecte cible. Pour ce faire :

- Une analyse lexicale a été mise en œuvre permettant de tokeniser chaque document du corpus d'entrainement. Pour chaque token (ou terme) obtenu, on calcule sa fréquence d'utilisation dans le corpus. Le but est bien de trier les termes par rapport à leurs fréquences d'apparition. Les termes sont ensuite filtrés pour ne garder que ceux qui appartiennent au dialecte algérien.
- Un prétraitement permet ensuite de « corriger » les termes du dialecte cible avant leur analyse syntaxique. Le prétraitement opéré consiste en 2 types d'analyses comme suit :

- La première permet de supprimer l'exagération (ie. Répétition volontaire de lettres dans un terme pour marquer une tonalité, une intensité, …). Par exemple, dans le terme : « bezzzzzaf », il y a exagération de la lettre « z ». Cette analyse permet de réduire le terme « bezzzzzaf » au terme « bezaf ».
- La deuxième, appelée analyse phonologique. Le but premier dans une *analyse phonologique* est d'identifier les sons qui créent des distinctions de sens. Le lexique du dialecte arabe algérien écrit est souvent enrichi avec des lettres alphabétiques latines et//ou des chiffres. Ainsi, pour maitriser certains sons, par exemple pour le son « غ », les utilisateurs font appel à la combinaison de deux lettres alphabétiques latines « gh », pour le son « ع » les utilisateurs utilisent « aa » ou bien « 3 ». L'analyse phonologique a pour objectif d'identifier, à partir des « enrichissements du lexique », le son correspondant dans le dialecte cible.

– Une analyse syntaxique du corpus est ensuite réalisée. Les auteurs ont proposé un analyseur syntaxique du dialecte algérien (ASDA). L'analyseur réalise une sorte d'étiquetage Pos (*Part of speech*), et associe à chaque terme d'un corpus de données, une étiquette définissant sa catégorie syntaxique (verbe, nom, adjectif, conjonction, pronoms personnels avec lesquels est conjugué un verbe, COI, COD, …). Par exemple, dans le terme « Rabbi » il y a deux parties, la première « Rabb » et la deuxième « i ». Cette décomposition servira dans la traduction du corpus vers la langue française. Dans l'exemple précédent, le terme traduit donne « mon Dieu » comme résultat.

– Enfin après avoir fait l'analyseur syntaxique, une table d'étiquetage qui permet la traduction d'un corpus du dialecte algérien vers une autre langue a été réalisée. Cette table a été utilisée dans l'analyse de sentiments des utilisateurs des médias sociaux vis-à-vis d'un sujet donné.

La figure suivante résume le traitement du dialecte algérien réalisé (ASDA : Analyseur Syntaxique du Dialecte Algérien dans un but d'analyse sémantique).

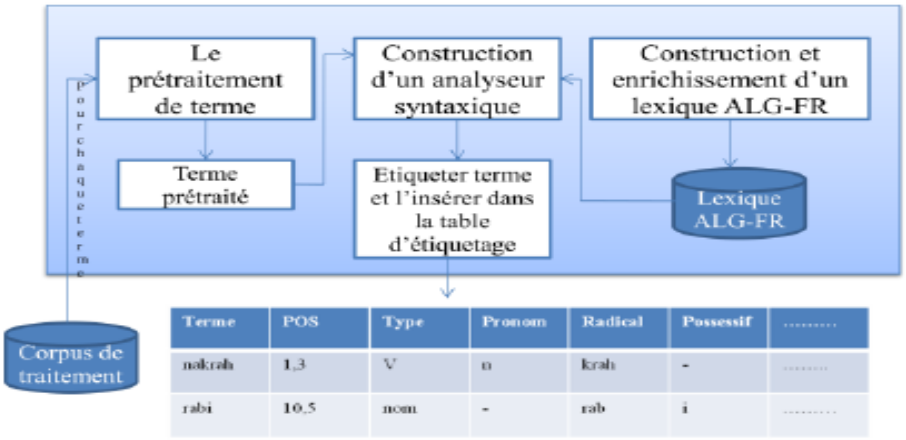


Figure 10 : Analyseur Syntaxique du Dialecte Algérien dans un but d'analyse sémantique]

3. Les travaux de (Guellil Imane, 2018) :

L'utilisation popularisée des réseaux sociaux, des blogs et des forums, a donné naissance à une nouvelle forme de l'arabe : L'*arabizi*. Il s'agit de l'arabe écrit avec l'alphabet latine. Les auteurs se sont intéressés à la translitération de l'arabizi. La translitération est un processus de passage d'un texte écrit en un script ou alphabet donné vers un autre, dans ce cas de l'arabizi algérien

vers l'arabe. En particulier, les auteurs ont élaboré un ensemble de règles permettant cette translitération.

L'approche de translittération proposée se base sur un ensemble d'étapes comme suit :

- La première étape consiste en l'extraction et le prétraitement d'un corpus arabe d'entrainement. Les auteurs ont d'abord extrait à partir de différentes pages sur les réseaux sociaux, des ensembles de commentaires de locuteurs algériens. Ces « documents » ont permis de constituer un corpus d'entrainement. Dans ce corpus, les auteurs se sont focalisés sur les messages écrits uniquement en arabizi, ensuite ils ont supprimé l'exagération dans ces mots.
- En deuxième étape, les auteurs ont proposé et utilisé un ensemble de règles pour l'arabizi algérien, qui définissent d'une part les différentes possibilités de remplacement de chaque caractère arabizi algérien, et d'autre part les règles de passage de l'arabizi vers l'arabe. La transformation des différents caractères arabizi en arabe par leurs Unicode grâce à un ensemble de règles résumées dans le tableau suivant :

Lettre en Arabizi	Lettre en arabe	Lettre en Arabizi	Lettre en Arabe	Lettre en Arabizi	Lettre en Arabe
A	ع ,أ ,ى ,ة ,ا ,"	k	ك,ق	U	أ ,و,"
B	ب	l	ل	V	ف
C	س ,ك	m	م	W	و
D	د, ض, ظ	n	ن	X	كس
E	ا,"	o	و ,"أ	Y	ي ,إ,"
F	ف	p	ب	Z	ز
G	ق	q	ك	7	ح
H	ح ,ه	r	ر,غ	5	خ
I	ي,"	s	س ,ص	3	ع
J	ج	t	ت ط	9	ق

Table 1 : Lettres de passage de l'arabizi vers l'Arabe

Après avoir appliqué les différentes règles de passages, un ensemble de mots candidats à la translittération est généré.

- Enfin la dernière étape est l'extraction du meilleur candidat. L'objectif de cette étape est de trouver le meilleur mot translitéré d'un mot arabizi. Cette étape est décomposée en deux sous-étapes :

- Dans la première sous-étape, on effectue recherche simple de chaque candidat au sein du corpus arabe permet de récupérer le nombre d'occurrences de chaque candidat dans le corpus.

- Dans la deuxième sous-étape, on effectue une recherche basée sur un modèle de langue appliqué au corpus arabe. Dans cette étape, on cherche chaque candidat au sein du modèle en extrayant la probabilité de chacun. Le candidat ayant la probabilité la plus élevée est retourné.

Les auteurs de cette étude ont signalé d'une part que leur approche ainsi proposée engendrait de nombreuses erreurs, principalement à cause de :

- La non prise en considération du contexte du mot dans la phrase,
- Le non traitement des mots ayant comme signification une langue étrangère,
- Le fait que le dialecte algérien est influencé par 3 langues : berbères, arabe, français et il est enrichi par des mots espagnols, turcs, phéniciens, romains.

Conclusion

Les études existantes sur le traitement automatique de l'arabe dialectalisé ne sont pas nombreuses. Les quelques études menées dans ce domaine sont éparses, et chacune s'intéressant à une problématique bien définie comme la translitération de l'arabizi : Approche Hybride pour la translitération de l'arabizi algérien (Guellil Imane, 2018), l'analyse syntaxique ASDA (Guellil Imane, Azouaou Faical, 2006), l'analyse morphologique du dialecte arabe algérien (SAADANE, 2015).

L'apprentissage automatique du dialecte arabe algérien n'a donc pas été abordé dans les études rencontrées. Néanmoins chacune de ces études a contribué à éclaircir beaucoup de zone d'ombres autour du dialecte arabe algérien et autour de son traitement automatique apportant ainsi une partie des éléments de réponse à notre problématique. Nous pourrons exploiter tous ces éléments dans notre travail.

Dans le chapitre suivant, nous allons exposer notre approche pour l'apprentissage automatique du dialecte algérien.

Chapitre IV

Apprentissage automatique du dialecte arabe algérien :

Approche proposée

Introduction

L’apprentissage automatique du dialecte algérien est une tâche difficile mais très intéressante pour les systèmes de recherche d’informations localisés en Algérie. Notre travail consiste à proposer et à mettre en place un système d’apprentissage automatique du dialecte arabe algérien. Dans ce chapitre, on va définir notre approche d’apprentissage et l’expliquer en détail, étape par étape.

Approche proposée

Afin de traiter le dialecte algérien et pouvoir comprendre la sémantique des mots, nous nous sommes basés sur la théorie de linguistique du philosophe Ludwig Wittgenstein qui disait que : « les mots n'ont de sens qu'à travers les mots qui les entourent ». L’idée est donc qu’un mot ne peut être compris qu’à travers son contexte. Nous avons donc pensé à exploiter cette théorie et l’adapter afin d’arriver à traiter le dialecte algérien. L’approche que nous proposons se base sur les réseaux de neurones. A cet effet, le *deep learning* nous semble une approche pertinente pour un apprentissage automatique performant du dialecte arabe algérien. Notre approche s’articule autour de 3 étapes principales (Voir figure 11) comme suit :

- Collecte de données
- Traitement des données
- Génération de la représentation contextuelle.

Etape 1 : La collecte de données

Notre approche commence par collecter de grands volumes de données du dialecte algérien pour avoir la multitude d’utilisations des mots dans différents contextes. À l’issue de cette étape, deux bases de données différentes sont créées, dont l’une contiendra les données écrites en arabizi, et l’autre les données écrites en arabe.

Etape 2 : Traitement des données

Dans cette étape, il s’agit de nettoyer les données brutes afin de construire un corpus épuré en vue de la représentation contextuelle. Cette étape s’appuie sur une succession de traitements que nous décrivons ci-après.

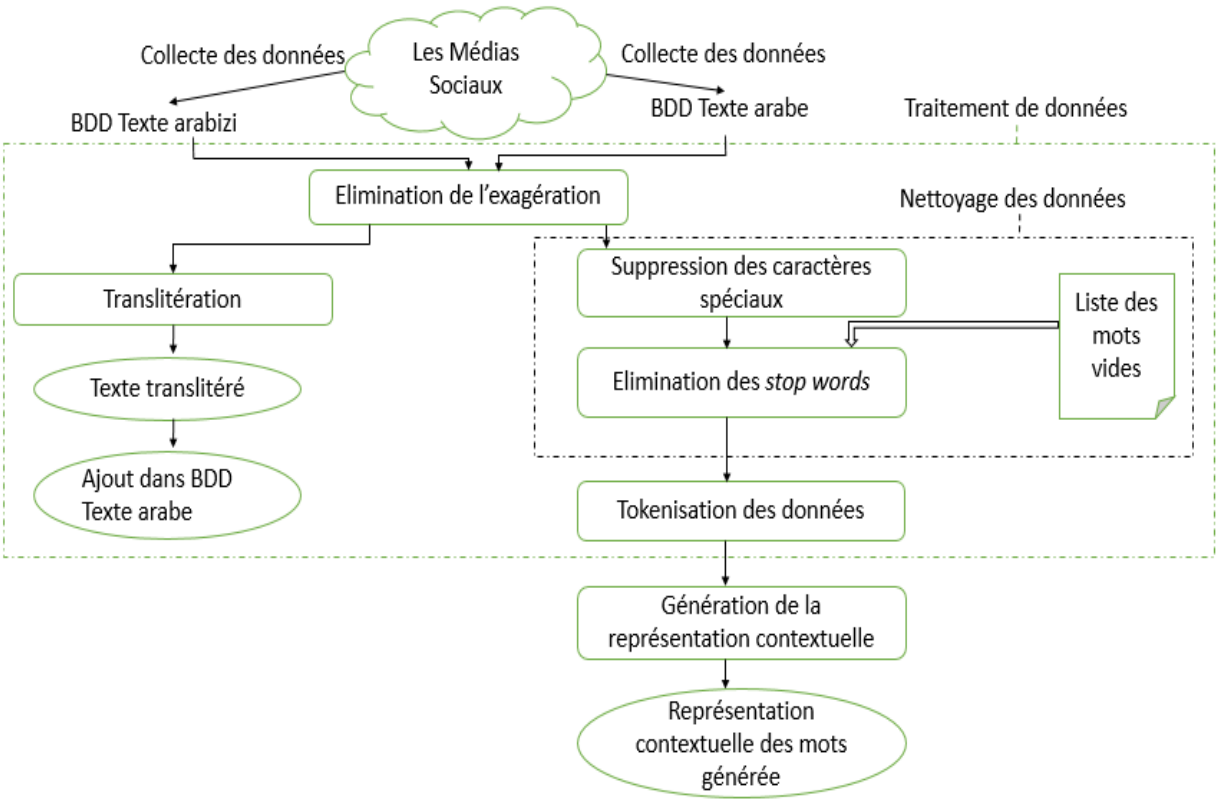


Figure 11 : Schéma générale de l'approche

a) Elimination de l'exagération

Dans cette étape, on considère respectivement les textes écrits en arabe et ceux écrits en arabizi. Il s'agit alors, pour chaque mot d'un texte, d'éliminer l'exagération dans ce mot.

b) La translitération

Dans cette étape, il s'agit de transcrire les textes arabizi en arabe. Pour ce faire, nous proposons d'utiliser l'approche de translitération de (Guellil Imane, 2018). Cette approche se base sur un ensemble de règles de passage de l'arabizi vers l'arabe telles que définies dans le tableau suivant :

Lettre en Arabizi	Lettre en arabe	Lettre en Arabizi	Lettre en Arabe	Lettre en Arabizi	Lettre en Arabe
A	ا,"ة,ى,أ,ع	k	ك,ق	U	و,أ,"
B	ب	l	ل	V	ف
C	س,ك	m	م	W	و
D	د,ض,ظ	n	ن	X	كس
E	ا,"	o	و,أ,"	Y	ي,إ,"
F	ف	p	ب	Z	ز
G	ق	q	ك	7	ح
H	ه,ح	r	ر,غ	5	خ
I	ي,"	s	س,ص	3	ع
J	ج	t	ت,ط	9	ق

Figure 12 : Lettres de passage de l'arabizi vers l'arabe

À l'issue de cette étape, on obtient des textes translitérés écrits en arabe. Ces textes sont ensuite insérés dans une base de données unifiée qui ne contient que des textes écrits en arabe, sans exagération des mots.

Exemple :

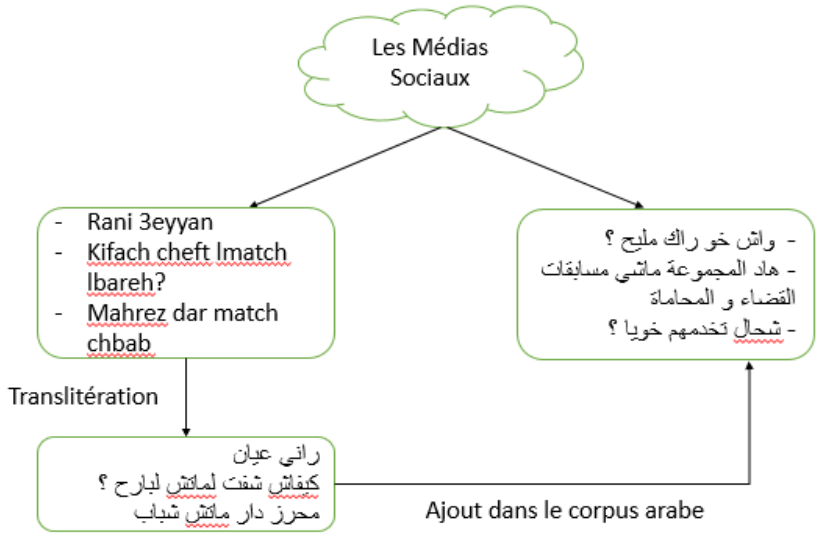


Figure 13 : Exemple de translitération de l'arabizi vers l'arabe

c) Nettoyage des données

Le nettoyage des données se fait en éliminant les caractères spéciaux et les mots vides. Les mots vides (*stop words*) sont des mots fonctionnels de la langue (on les utilise afin de relier des

phrases entre elles comme par exemple la conjonction de coordination ’car’ qui exprime la cause), car ce qui intéresse le traitement ce sont les mots non vides du contexte. Pour reconnaître les mots vides, nous proposons la liste suivante des mots vides (ou *stopwords*) du dialecte arabe algérien (Voir figure 14).

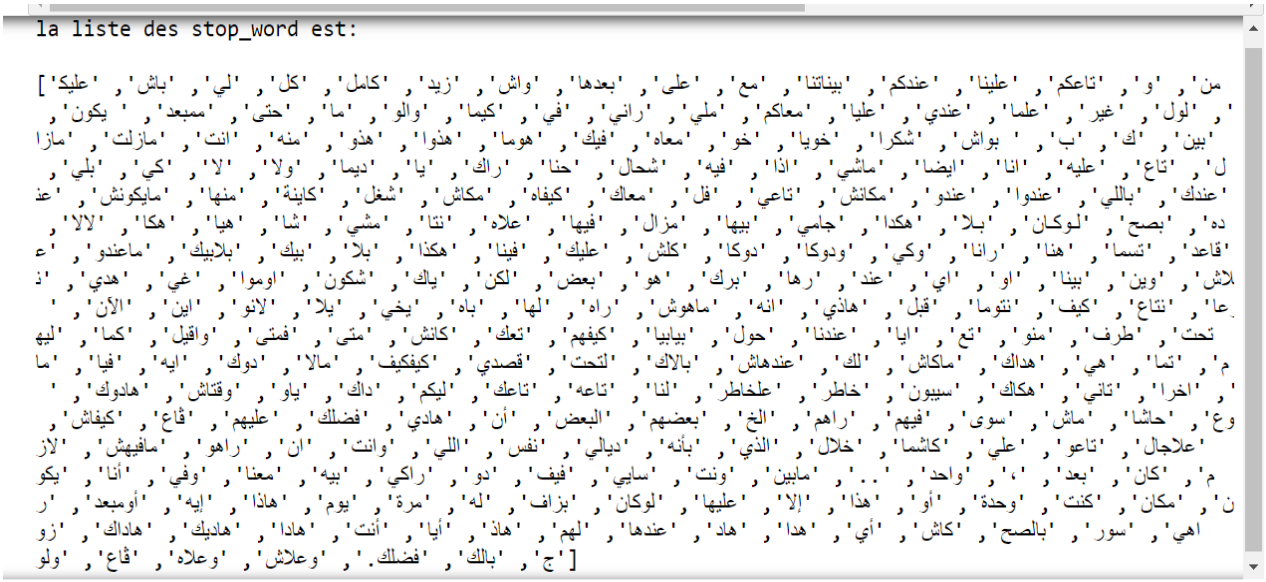


Figure 14 : Liste des stop words

d) La Tokénisation des données

Cette étape sert à séparer les mots, de notre corpus nettoyé, les uns des autres et les mettre dans une liste en respectant le même ordre dans leurs phrases. Le but de cette étape est ainsi de préparer les mots afin de les représenter contextuellement.

Etape 3 : Génération de la représentation contextuelle

Dans cette partie nous proposons une structure capable de produire une représentation intelligente des mots. Cette structure prend en entrée une grande quantité d’information, pour en extraire des caractéristiques (*features*), ce qui nous a conduit à l’utilisation des réseaux de neurones.

Notre réseau de neurones s’inspire de celui préconisé dans le modèle Word2vec, et se compose de 3 couches : une couche d’entrée, une couche cachée et une couche de sortie. Le fonctionnement de notre réseau de neurones est expliqué dans le schéma suivant :

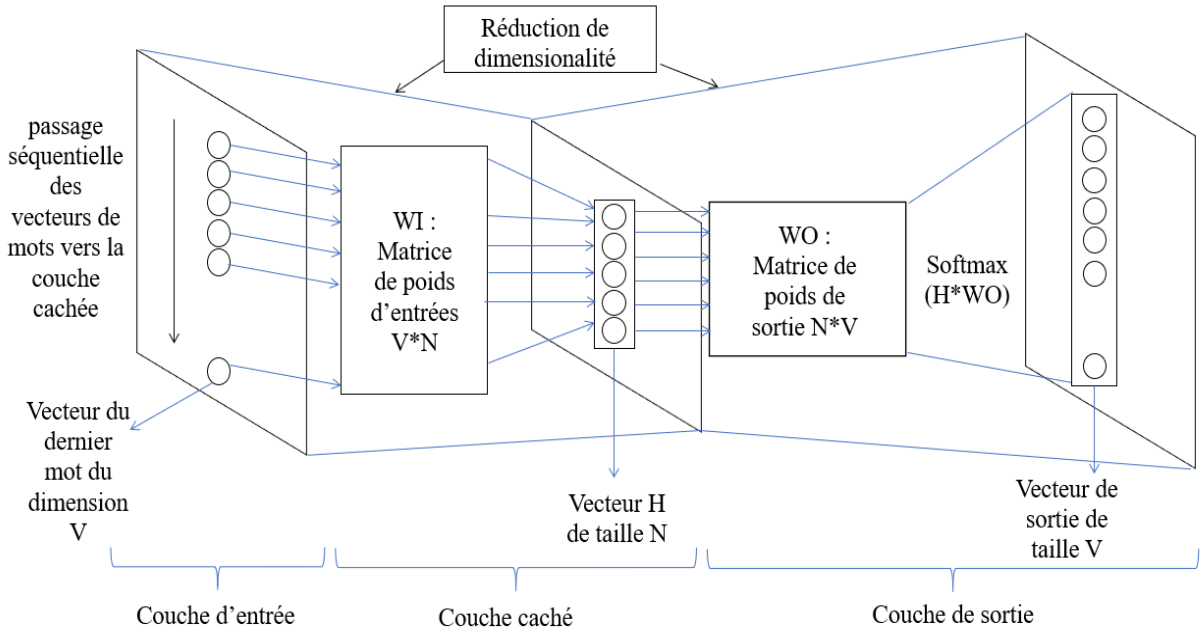


Figure 15 : Schéma montrant la génération de la représentation contextuelle

- La couche d'entrée va avoir comme entrée un vecteur de dimension V, tel que V est le nombre de mots contenus dans notre vocabulaire. Ce vecteur va contenir des 0 partout sauf 1 à la position correspondante à notre mot. Les neurones de cette couche représentent le vecteur à chaud d'un mot. Ces neurones ne sont pas interconnectés entre eux. Le rôle de cette couche est de faire passer séquentiellement les vecteurs à chaud de chaque mot vers la couche cachée.

- La couche cachée va être de dimension N, N représente le nombre de neurones que va contenir la couche cachée ainsi que la taille de vecteur H. En effet cette couche est une couche de projection, car elle sert à projeter notre vecteur de dimension V sur un vecteur H de dimension N tel que N << V : le processus se présente sous le nom de la réduction de la dimensionnalité. Comme le montre le schéma ci-dessus, cette réduction se fait en multipliant notre vecteur initial de dimension V par la matrice de poids d'entrées Wi de dimension V*N.
Chaque neurone de cette couche représente une dimension, ces neurones ne sont pas interconnectés entre eux.

- La couche de sortie va nous fournir comme résultat un vecteur de taille V, Ce vecteur va être le résultat d'une fonction d'activation qui calcule la probabilité d'appartenance de chaque mot du corpus au contexte du mot cible, ainsi on pourra calculer l'erreur et effectuer le processus de retro-propagation afin d'ajuster les poids de nos matrices. L'erreur est détectée grâce à un

ensemble de données étiquetées qui va être construit par notre réseau. Pour bien illustrer ce point, nous nous présentons cet exemple :

Phrase1 = 'حسبي الله ونعم الوكيل', phrase2= 'بسم الله الرحمان الرحيم', phrase3= 'سبحان الله العظيم',

Phrase4 = 'والله الطوموبيل هذيك طارت'

En prenant ces quatre phrases en entrées, on aura comme résultats dans la phase prétraitement à titre d'exemple pour le mot 'الله' les résultats suivants :

('حسبي','الله'), ('سبحان','الله'), ('العظيم','الله'), ('بسم','الله'), ('الرحمان','الله'), ('نعم','الله'), ('الطوموبيل','الله')

Donc, dans le passage de ces données dans notre réseau de neurones, ce dernier va essayer d'ajuster nos matrices de poids afin d'obtenir exactement le même résultat.

On remarque que dans notre cas, on a un résultat incohérent qui fait lier le mot 'الله', et le mot 'الطوموبيل', sauf que ce résultat ne sera pas pris en compte en fin de notre phase d'entrainement, et cela pour la simple raison que les autres mots cités précédemment, on aura souvent l'occasion de les donner comme entrées à notre réseau de neurones ce qui augmentera la probabilité de voir le mot 'الرحمان', ou bien 'العظيم' après le mot 'الله', et cela n'est pas le cas entre le mot 'الله' et 'الطوموبيل'.

On voit bien dans cette exemple qu'on prend juste un mot à droite et un mot à gauche, ce qui empêchera notre réseau de neurones de donner le résultat 'الرحيم' pour le mot 'الله', malgré leurs relations. De ce fait on a pensé à agrandir la taille de la fenêtre (nombre de mots à prendre adroite et à gauche du mot) selon notre besoin. Exemple :

('نعم', 'الله', 'حسبي'), ('الرحيم', 'الرحمان', 'الله').

Conclusion :

Dans ce chapitre nous avons proposé une approche pour l'apprentissage automatique du dialecte arabe algérien. Cette approche s'appuie sur les réseaux de neurones, elle prend le corpus de données massives de notre dialecte en entrée et donne les représentations vectorielles pour chaque mot en sortie. Dans le chapitre suivant on va implémenter notre approche et voir les résultats qu'elle va donner et faire des différents tests et finir par une évaluation de notre system.

Chapitre V

Réalisation

Introduction

Dans ce chapitre nous allons implémenter notre approche. Comme le modèle d'apprentissage que nous avons conçu s'inspire du modèle Word2vec pour la représentation contextuelle des mots, et comme ce dernier possède déjà une implémentation libre paramétrable, nous allons utiliser cette implémentation pour la réalisation de notre système d'apprentissage automatique, mais avant cela on va parler sur les différents outils que nous avons utilisés.

1. Les outils utilisés :

Java : Est un langage de programmation orienté objet très bien structuré qui utilise principalement des objets de type class., qui a su se faire une place primordiale dans le monde de l'informatique et a réussi à se hisser au premier rang ; ceci grâce à sa constante évolution et à ses particularités.

Python : est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.

Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Gensim : Est une bibliothèque libre de Python conçue pour extraire automatiquement des sujets sémantiques des documents, pour traiter les textes numériques bruts et non-structurés. Les algorithmes de gensim, tels que Word2vec, n'ont besoin que d'un corpus de documents en texte brut.

La bibliothèque Python **xlrd** consiste à extraire des données de fichiers de feuille de calcul Microsoft Excel.

Anaconda : Est une distribution gratuite et open-source des langages de programmation Python et R et pour le calcul scientifique (science des données, applications d'apprentissage automatique, traitement de données à grande échelle, etc.), qui vise à simplifier la gestion des paquets et déploiement. En effet, cette dernière permet de créer un environnement virtuel sur une machine, cela permet de travailler au même temps sur des différents projets sollicitant des outils ou des technologies avec des versions différentes sans avoir des interférences ou de problèmes de versions.

Cette dernière offre aussi des outils très intéressant comme :

Jupyter notebook : Est un environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données Jupyter.

Word2Vec : Est l'une des techniques les plus populaires pour apprendre l'intégration de mots à l'aide d'un réseau neuronal peu profond. On utilisera cet outil dans la représentation contextuelle des mots, on a défini ce modèle ainsi que son fonctionnement dans le chapitre 1**.**

2. Mise en œuvre :

Dans cette étape on va implémenter notre approche proposée dans le chapitre précèdent

1- Collecte de données :

Au départ nous avons trouvé en face de nous un grand obstacle qui est l'absence de corpus de données bien défini du dialecte algérien alors nous l'avons construit manuellement. Ce corpus contient plus de 10 000 phrases, on les a récupérés à partir de différentes pages Facebook où la communauté algérienne interagisse, citons : les pages des opérateurs : Djezzy, Ooredoo et Mobilis. Autres pages comme : Condor, Brandt et les pages de vente et achats mobiles comme : DZ Electronics et d'autres pages officielles.

On a créé deux bases de données, une pour les données écrites en alphabet arabe et l'autre pour les données écrites en alphabet latine (arabizi).

Voici un extrait de données écrites en arabe (figure 16) suivi d'un extrait de données écrites en arabizi (Figure 17) :

10190	ماكاش لي يغطي عليك ماكاش لي معاك مليحة وزيد لونكادرور مليح قاع
10191	كلش مونوم حنا
10192	إيه واش كنت رايح تقول
10193	أي واحد قاري أنفورماتيك قادر يفهموا
10194	أنت تفهمه
10195	نعطيك واحد يفهمو صح تعرفه الطاهر
10196	لالا نتكل على الأولاد
10197	عندي تاست الخميس
10198	أول ماي بالخميس على شحال
10199	تاست من الثلاثة للربعة و نص
10200	ميكرو قدامك تعطيك بروڤرام غالط و أنت تسڤمه من الثلاثة للربعة ونص
10201	و يتحسب التاست تاعكم ولا واش
10202	أووليمس ما نعرف حتى واحد فهمت
10203	الخميس هدا حتى كون جيت قاعد نورمال نجوز الإمتحانات
10204	سيبون رايح نكمل مازالولي غير هدو الإمتحانات
10205	حسيت بالعيا تاعك
10206	صح

Figure 16 : Extrait des données du dialecte algérien écrits en arabe

1	Ya kho rani habet
2	Ya kho kheli ki nweli ngesro
3	Raho m3a lhaj
4	Apres nfehmek ou fehemli hadi ta3 sécurité sociale
5	Hhhh nn ana bakayaa nabkiii 3la mama w paapa si pour Sa nag3d bayra
6	Hna 3 bnat (bant mazja 3wd ana 3wd khti sghira 16ans) wkhoya
7	Mnbghich ndirha profil
8	Wesh a hsa mazalk raqed
9	Meme problème ya khoya wch n9olk alah yferdjha 3lina wla rebi ymedna refda mn hedi lbled

Figure 17 : Extrait des données du dialecte algérien écrits en arabizi

2- Traitement des données :

a) - Elimination de l'exagération

Voici le programme (en java) qui permet d'éliminer l'exagération :

```
11
12    public static String del_Rep(String x ){
13
14    int i=0;
15
16     ArrayList <Character> k = new ArrayList <Character> ();
17
18        char [] a = x.toCharArray();
19
20         k.add(a[i]);
21
22 while (i < a.length-1) {
23
24        if (a[i] == a[i+1]){
25
26            i++;
27          } else{
28
29               k.add(a[i+1]);
30
31              i++;
32
33          }
34
35        }
36
```

Figure 18 : programme JAVA : Elimination de l' exagération

b) - Translitération

Le programme de translitération :

```
	public static String trans (String x ){
		char [] a = x.toCharArray();
		 ArrayList <Character> k = new ArrayList <Character> ();
		for (int i=0 ; i < a.length ; i++ ){
			switch (a[i]){
			case ' ' :
				k.add(' ');
			    break;
			case ',' :
				k.add('،');
			    break;
			case '.' :
				k.add('.');
			break;
			case '%' :
				k.add('%');
			    break;
			case ':' :
				k.add(':');
			    break;
			case ';' :
				k.add('؛');
			    break;
			…
			case 'c' :if ( (a[i+1] == 'a') || (a[i+1] == 'o') || (a[i+1] == 
'u') || (a[i+1] == ' ') ) {
				k.add('ك');
					i=i+1;
			} else		if (a[i+1] == 'h'){
				k.add('ش');
				 i=i+1;
			} else {
				k.add('س');
			}
			break;
			case 'd' : if (a[i+1] == 'h'){
				k.add('ض');
				i=i+1;
				}else if (a[i+1] == 'j'){
						k.add('ج');
						i=i+1;
					}else{
					k.add('د');
				    }
			break;
			case 'e' : if (i == 0){
				k.add('أ');
			}
			break;
			case 'é' :
				k.add('ي');
			break;
			case 'è' :
				k.add('ي');
```

```
                    break;
                    case 'f' :
                          k.add('ف');
                    break;
                    case 'g' :  if (a[i+1] == 'h'){
                          k.add('غ');
                          i=i+1;
                    }
                    else {
                          if ( (a[i+1] == 'e') || (a[i+1] == 'é') || (a[i+1] ==
'è')  ) {
                                k.add('ج');
                                i=i+1;
                          } else {
                          k.add('ق');
                           }
                    }
                    break;
                    case 'h' : k.add('ح');
                    break;
                    case 'i' : k.add('ي');
                    break;
                    case 'j' :
                          k.add('ج');
                    break;
                    }
              }
              String listchar="";
               for (Character s : k)
               {
                   listchar += s ;
               }
        return listchar;
        }
```

Exemple : On prend ce texte (Voir figure 19) :

```
394
395
396         String nom = ("t3echit wella mazal ? , koul matehchemch rak fi darek ");
397
398
399
```

Figure 19 : Exemple à translitérer de l'arabizi vers l'arabe

Le résultat de la translitération de ce texte est montré dans cette figure ci-dessous :

```
le texte apres transleteration est : تعشيت ولا مازال ؟ كول ماتحشمش راك في دارك
```

Figure 20 : Résultat de translitération de l'arabizi vers l'arabe

c) – Nettoyage des données

- L’élimination des caractères spéciaux :

Exemple 1 :

```
#مكاش الانترنت!!!!!!!?????
la phrase avant d'enlever les caractéres spéciaux et les chiffres

_____________________________________________________________
_____________________________________________________________
la phrase aprés avoir enlever les caractéres spéciaux et les chiffres

مكاش الانترنت
```

Exemple 1 : Elimination des caractères spéciaux

Exemple 2 :

```
41لازم تزيدوهم 985 في تطبيق جازي!!
la phrase avant d'enlever les caractéres spéciaux et les chiffres

_____________________________________________________________
_____________________________________________________________
la phrase aprés avoir enlever les caractéres spéciaux et les chiffres

لازم تزيدوهم في تطبيق جازي
```

Exemple 2 : Elimination des caractères spéciaux

Le code suivant nous permet d’éliminer les mots vides :

```
for i in list1:
    if((i in swgen) ):
        list1.remove(i)
list2.append(list1)
```

La figure ci-dessous montre un exemple sur la phase de nettoyage de données (Eliminer les mots vides) :

```
----------------------------------------
la liste avant d'enlever les mots vides

['تأمين' ,'بشركة' ,'نتصل' ,'حاب' ,'راني']
la liste aprés avoir enlever les mots vides

['تأمين' ,'بشركة' ,'نتصل' ,'حاب']
----------------------------------------
la liste avant d'enlever les mots vides

['الوجع' ,'انقص' ,'باش' ,'باردة' ,'كمادة' ,'ندير' ,'دوك']
la liste aprés avoir enlever les mots vides

['الوجع' ,'انقص' ,'باردة' ,'كمادة' ,'ندير']
----------------------------------------
```

Figure 21 : Exemple sur l'élimination des mots vides

d) – Tokenisation des données

Lors du traitement automatique du texte, l'étape de tokenisation est une étape primordiale pour le bien fonctionnement de notre programme

Pour la représentation contextuelle de nos données, on va créer un modèle *Word2Vec*. Pour nous, notre modèle Word2vec, exige que les données en entrées soient sous forme d'une liste de listes où chaque liste comportera une phrase (dans notre cas une ligne de notre corpus sous Excel). Donc dans cette phase on crée une liste de liste.

Le code suivant nous permet de faire la tokenisation :

```
while curr_cell < num_cells:
      #récuprer les phrases depuis excel
      cell_type = worksheet.cell_type(curr_row, curr_cell)
      cell_value = worksheet.cell_value(curr_row, curr_cell)
      #faire des prétraitement netoyage
      string = cell_value.replace(u'\xa0', u' ')
      #mettre la phrases dans une liste et faire des préraitement (du netoyage)
      list1=string.split(' ')
      #ajouter la sous-liste contenant la ligne numéro n à la liste pricipale
      list2.append(list1)
```

Figure 22 : Un extrait du code de Tokenisation

3- Génération de la représentation contextuelle

Création du modèle Word2vec : (paramétrage de Word2vec)

Min count = 2 => c'est-à-dire qu'il va prendre en considération juste les mots qui se sont répétés au moins 2 fois dans le corpus. Nous avons choisi cette valeur pour avoir un vocabulaire important. C'est une valeur prouvée dans la partie test.

Taille de fenêtre = 5 => le programme lors de son exécution va prendre 5 mots avant le mot cible et 5 mots après le mot cible. C'est une valeur prouvée dans la partie test.

Taille du vecteur = 100 => Nombre de neurones de la couche cachée est 100. Donc le vecteur de la représentation contextuelle va être de dimension 100.

Epochs = 200 => Vu la taille très faible de notre corpus, on a mis un epochs élevé pour assurer la convergence du réseau. Cette valeur nous permet d'obtenir de meilleurs résultats. C'est une valeur prouvée dans la partie test.

Alpha = 0.01 => C'est une valeur recommandée par la communauté scientifique

À l'issue de cette étape, le modèle est créé donc on lance la phase '*Training*' pour entrainer notre modèle :

```
word2vec1.train(list3, total_examples=word2vec.corpus_count, epochs=30)
words = list(word2vec.wv.vocab)
print(words)
```

- Avec notre modèle on peut trouver :

1- Le degré de similarité entre deux mots (Voir figure 23) :

```
#mincount=2
word2vec1 = Word2Vec(min_count=2, size=100, window=5 )
word2vec1.build_vocab(list3)

word2vec1.train(list3, total_examples=word2vec.corpus_count, epochs=30)

print(word2vec.similarity("الاحرار","خراطة"))
```

```
0.9953871
```

Figure 23 : Degré de similarité entre deux mots

- On a un résultat qui montre un degré de similarité élevé entre le mot 'الاحرار' et le mot 'خراطة' qui est de 0.995.

2- Les 10 mots les plus proches du mot cible (Voir figure 24) :

```
Entrée [20]: sim_words = word2vec.wv.most_similar("المجتمع")
             print(sim_words)

[('الفرد', 0.9264086484909058), ('الجزائري', 0.9100626707077026), ('المجتمعات', 0.9092104434967041), ('قوانين', 0.9014225006103516), ('الأخلا
بية', 0.894662618637085), ('الغرب', 0.8926489353179932), ('النظام', 0.8849635720252991), ('بالنسبة', 0.883112907409668), ('الأفكار', 0.8803
[('البرلمان', 0.8770604729652405), (635835647583
```

```
Entrée [9]: sim_words = word2vec.wv.most_similar("الله")
            print(sim_words)

[('الوكيل', 0.9068782329559326), ('يرحم', 0.8921998739242554), ('بارك', 0.8853878974914551), ('سبحان', 0.8795993924140930), ('التوا', 0.8
0.8715043663970000), ('اهلو', 0.8720219135284424), ('المختار', 0.8738305568695068), ('ونعم', 0.8745311498641968), ('غالب', 765010833740234),
[('استغفر', 0.8711918592453003), (8577
```

Figure 24 : Exemple d'extraction des mots les plus proches du mot « "الله" et "المجتمع" »

3- Le degré de similarité entre deux contextes (Voir figure 25) :

```
Entrée [23]: word2vec.n_similarity(["الجزائري","المجتمع"],["فن","ثقافة"])

<ipython-input-23-9a7af02c835d>:1: DeprecationWarning: Call to deprecated `n_similarity` (Method will be removed in 4.0.0, use
self.wv.n_similarity() instead).
  word2vec.n_similarity(["الجزائري","المجتمع"],["فن","ثقافة"])

Out[23]: 0.7980373
```

Figure 25 : Degré de similarité entre deux contextes

- Le résultat montre que le degré de similarité entre les deux contextes ["الجزائري", "المجتمع"], ["فن", "ثقافة"] est : 0.798

4- Le mot le plus proche d'un contexte (Voir figure 26) :

```
Entrée [142]: word2vec.predict_output_word("حسبي الله ونعم ".split(),topn=5)

Out[142]: ,(0.2483142 ,'الوكيل')]
          ,(0.20659083 ,'حسبنا')
          ,(0.19348304 ,'شاء')
          ,(0.10758666 ,'ونعم')
          [(0.025205469 ,'غالب')
```

Figure 26 : Extraction des mots les plus proches d'un contexte

- Les résultats montrent que mot le plus pertinent du contexte « حسبي الله و نعم » est :' الوكيل'

5- Le mot hors contexte (Voir figure 27) :

```
ntrée [25]: word2vec.doesnt_match(["كورونا","الفرد","الشعب","الأمة","المجتمع"])

<ipython-input-25-fc9b3904a019>:1: DeprecationWarning: Call to deprecated `doesnt_match` (Method will be removed in 4.0.0, use
self.wv.doesnt_match() instead).
  word2vec.doesnt_match(["كورونا","الفرد","الشعب","الأمة","المجتمع"])
C:\Users\ANIS\anaconda3\envs\skipgram\lib\site-packages\gensim\models\keyedvectors.py:877: FutureWarning: arrays to stack must
be passed as a "sequence" type such as list or tuple. Support for non-sequence iterables such as generators is deprecated as of
NumPy 1.16 and will raise an error in the future.
  vectors = vstack(self.word_vec(word, use_norm=True) for word in used_words).astype(REAL)

Out[25]: 'كورونا'
```

Figure 27 : Extraction du mot hors contexte

- Le résultat montre que le mot 'كورونا ' est le mot hors contexte de l'ensemble : [المجتمع، الأمة، الشعب، الفرد، كورونا]

Conclusion :

À l'issue de ce chapitre, nous avons réussi à implémenter notre approche d'apprentissage du dialecte arabe algérien, en utilisant le modèle word2Vec pour la génération de la représentation contextuelle. Ceci nous a permis de réaliser plusieurs opérations comme la déduction du mot hors contexte dans un ensemble de mots.

Dans ce qui suit, nous décrivons une étape complémentaire de notre travail, qui est l'évaluation de notre approche.

Chapitre VI

Tests & Evaluation

Introduction

Après avoir implémenté, et mettre en œuvre notre approche, dans ce chapitre nous allons faire différents tests, en modifiant les paramètres de configuration de Word2Vec afin de trouver les paramètres adéquats retournant les meilleurs résultats.

1 - Test :

Dans cette partie, on fait des tests sur les paramètres de configuration de word2Vec

```
: from gensim.models import Word2Vec
  #mincount=1
  word2vec = Word2Vec(min_count=1, size=100, window=5 )
  word2vec.build_vocab(list3)
  print(word2vec)
  word2vec.train(list3, total_examples=word2vec.corpus_count, epochs=30)
  print(word2vec.similarity("لانوت","ديبلوم"))
  #mincount=2
  word2vec1 = Word2Vec(min_count=2, size=100, window=5 )
  word2vec1.build_vocab(list3)
  print(word2vec1)
  word2vec1.train(list3, total_examples=word2vec.corpus_count, epochs=30)
  print(word2vec1.similarity("لانوت","ديبلوم"))
  #mincount=10
  word2vec2 = Word2Vec(min_count=10, size=100, window=5 )
  word2vec2.build_vocab(list3)
  print(word2vec2)
  word2vec2.train(list3, total_examples=word2vec.corpus_count, epochs=30)
  print(word2vec2.similarity("لانوت","ديبلوم"))
```

Figure 28 : Test avec changement du paramètre min count

```
Word2Vec(vocab=18174, size=100, alpha=0.025)
<ipython-input-92-540ca33e31a1>:7: DeprecationWarning: Call to deprecated `similarity` (Me
lf.wv.similarity() instead).
  print(word2vec.similarity("لانوت","ديبلوم"))
0.9604699
Word2Vec(vocab=7334, size=100, alpha=0.025)
<ipython-input-92-540ca33e31a1>:13: DeprecationWarning: Call to deprecated `similarity` (M
elf.wv.similarity() instead).
  print(word2vec1.similarity("لانوت","ديبلوم"))
0.99641025
Word2Vec(vocab=933, size=100, alpha=0.025)
<ipython-input-92-540ca33e31a1>:19: DeprecationWarning: Call to deprecated `similarity` (M
elf.wv.similarity() instead).
  print(word2vec2.similarity("لانوت","ديبلوم"))
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-92-540ca33e31a1> in <module>
     17 print(word2vec2)
     18 word2vec2.train(list3, total_examples=word2vec.corpus_count, epochs=30)
---> 19 print(word2vec2.similarity("لانوت","ديبلوم"))

~\anaconda3\envs\skipgram\lib\site-packages\gensim\utils.py in new_func1(*args, **kwargs)
   1459                       stacklevel=2
```

Figure 29 : Les Résultats obtenus lors du test

Ces deux figures montrent donc, qu'on a créé trois modèles similaires, puis on fait varier le paramètre **min-count**. Par la suite, on a testé le degré de similarité entre les deux mots "لانوت","ديپلوم" (diplôme et notes) et les résultats sont les suivants :

Pour min-count = 1, le degré de similarité est de 0.96.

Pour min-count = 2, le degré de similarité est de 0.99.

Pour min-count=10, le programme échoue est nous signale que le mot "ديپلوم" n'existe pas dans notre corpus (nombre de répétition du mot "ديپلوم " dans notre corpus est inférieur à 10).

- Dans la suite, on considère que :

- les paramètres : *vector_size*, *min_count*, et *rate Learning* ont des valeurs fixes qui sont respectivement : 100, 2, 0.01.

- Relativement à notre corpus qui est petit, le paramètre *min-count* est affecté de la valeur 2 (voir figure 30). À noter qu'on condition réelle, le min-count ne doit pas être inférieur à 20.

```
: from gensim.models import Word2Vec
  #mincount=1
  word2vec = Word2Vec(min_count=1, size=100, window=5 )
  word2vec.build_vocab(list3)
  print(word2vec)

  #mincount=2
  word2vec1 = Word2Vec(min_count=2, size=100, window=5 )
  word2vec1.build_vocab(list3)
  print(word2vec1)

  #mincount=10
  word2vec2 = Word2Vec(min_count=10, size=100, window=5 )
  word2vec2.build_vocab(list3)
  print(word2vec2)

  Word2Vec(vocab=18174, size=100, alpha=0.025)
  Word2Vec(vocab=7334, size=100, alpha=0.025)
  Word2Vec(vocab=933, size=100, alpha=0.025)
```

Figure 30 : Relation entre min_count et taille vocabulaire

Vector size : on a choisi de le mettre à 100 parce que notre corpus est relativement petit. Cette valeur est la mieux adapté pour cela.

Rate Learning : celui-ci on le prend à 0.01, c'est une valeur recommandée par la communauté scientifique.

Pour les deux paramètres restants, " *window* et *epochs*", on a décidé de faire des tests sur un ensemble de mots en attribuant 4 valeurs différentes pour la taille de la fenêtre et 3 valeurs différentes pour l'epochs.

Les résultats sont présentés dans la section " evaluation ".

2 - Evaluation :

L'évaluation de notre système d'apprentissage se fait sur la base de deux métriques suivantes : la précision et le rappel, définies comme suit :

$$\textbf{Précision} = \frac{\textbf{\textit{Nombre de mots pertinent retourné}}}{\textbf{\textit{Nombre de mots retournés}}}$$

$$\textbf{Rappel} = \frac{\textbf{\textit{Nombre de mots pertinents retournés}}}{\textbf{\textit{Nombre de mots pertinents attendus}}}$$

Pour avoir le nombre de mots pertinents attendus, on a créé une base de données qui va contenir le contexte de quelques mots qu'on va tester. Cette base va nous aider pour calculer le rappel.

Les résultats des tests sont présentés dans les sections suivantes.

1- Pour le mot 'الله' :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	2/10	2/15	5/10	5/15	6/10	6/15
Taille de fenêtre = 5	3/10	3/15	7/10	7/15	7/10	7/15
Taille de fenêtre = 7	2/10	2/15	7/10	7/15	6/10	6/15
Taille de fenêtre=10	2/10	2/15	6/10	6/15	8/10	8/15

2- Pour le mot ‘ الجزائر ’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/11	5/10	5/11	4/10	4/11
Taille de fenêtre = 5	2/10	2/11	6/10	6/11	5/10	5/11
Taille de fenêtre = 7	2/10	2/11	6/10	6/11	5/10	5/11
Taille de fenêtre=10	3/10	3/11	5/10	5/11	5/10	5/11

3- Pour le mot ‘القانون ’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/13	5/10	5/13	5/10	5/13
Taille de fenêtre = 5	1/10	1/13	9/10	9/13	9/10	9/13
Taille de fenêtre = 7	2/10	2/13	8/10	8/13	8/10	8/13
Taille de fenêtre=10	2/10	2/13	7/10	7/13	8/10	8/13

4- Pour le mot ‘حقوق’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	2/10	2/13	4/10	4/13	5/10	5/13
Taille de fenêtre = 5	3/10	3/13	8/10	8/13	8/10	8/13
Taille de fenêtre = 7	3/10	3/13	8/10	8/13	8/10	8/13
Taille de fenêtre=10	3/10	3/13	7/10	7/13	8/10	8/13

5- Pour le mot ‘ مسابقة ’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/10	5/10	5/10	4/10	4/10
Taille de fenêtre = 5	2/10	2/10	8/10	8/10	8/10	8/10
Taille de fenêtre = 7	2/10	2/10	6/10	6/10	8/10	8/10
Taille de fenêtre=10	2/10	2/10	6/10	6/10	2/10	2/10

6- Pour le mot ‘ دولة ’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/15	6/10	6/15	6/10	6/15
Taille de fenêtre = 5	2/10	2/15	9/10	9/15	9/10	9/15
Taille de fenêtre = 7	1/10	1/15	7/10	7/15	6/10	6/15
Taille de fenêtre=10	1/10	1/15	6/10	6/15	8/10	8/15

7- Pour le mot ‘محمد’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	3/10	3/12	3/10	3/12	3/10	3/12
Taille de fenêtre = 5	4/10	4/12	6/10	6/12	5/10	5/12
Taille de fenêtre = 7	3/10	3/12	5/10	5/12	5/10	5/12
Taille de fenêtre=10	2/10	2/12	5/10	5/12	5/10	5/12

8- Pour le mot ‘طبيب ’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/13	3/10	3/13	3/10	3/13
Taille de fenêtre = 5	1/10	1/13	6/10	6/13	5/10	5/13
Taille de fenêtre = 7	1/10	1/13	5/10	5/13	5/10	5/13
Taille de fenêtre=10	1/10	1/13	5/10	5/13	4/10	4/13

9- Pour le mot ‘ حياة’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	1/10	1/12	3/10	3/12	4/10	4/12
Taille de fenêtre = 5	2/10	2/12	7/10	7/12	7/10	7/12
Taille de fenêtre = 7	2/10	2/12	6/10	6/12	5/10	5/12
Taille de fenêtre=10	1/10	1/12	5/10	5/12	5/10	5/12

10- Pour le mot ‘المجتمع’ :

	Epochs = 30		Epochs = 200		Epochs = 300	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
Taille de fenêtre = 2	2/10	2/13	6/10	6/13	6/10	6/13
Taille de fenêtre = 5	3/10	3/13	8/10	8/13	7/10	7/13
Taille de fenêtre = 7	3/10	3/13	7/10	7/13	6/10	6/13
Taille de fenêtre=10	4/10	4/13	7/10	7/13	7/10	7/13

Dans ce qui suit, nous calculons la moyenne de précision et de rappel pour chaque paramètre de configuration afin de trouver la meilleure configuration pour notre modèle.

- La moyenne de précision et de rappel pour epochs = 30 :

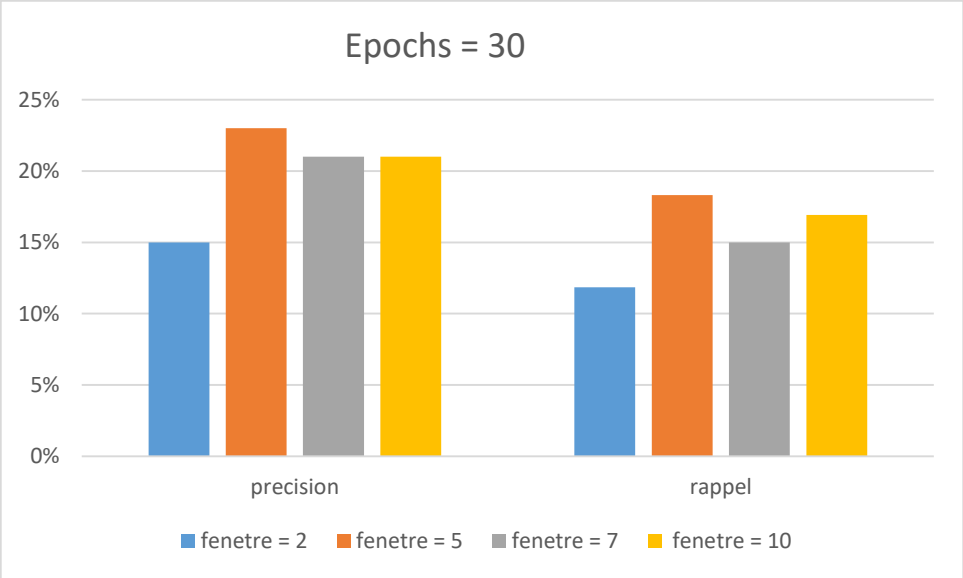


Figure 31 : Graphe montrant la moyenne de précision et de rappel (epochs = 30)

- La moyenne de précision et de rappel pour epochs = 200 :

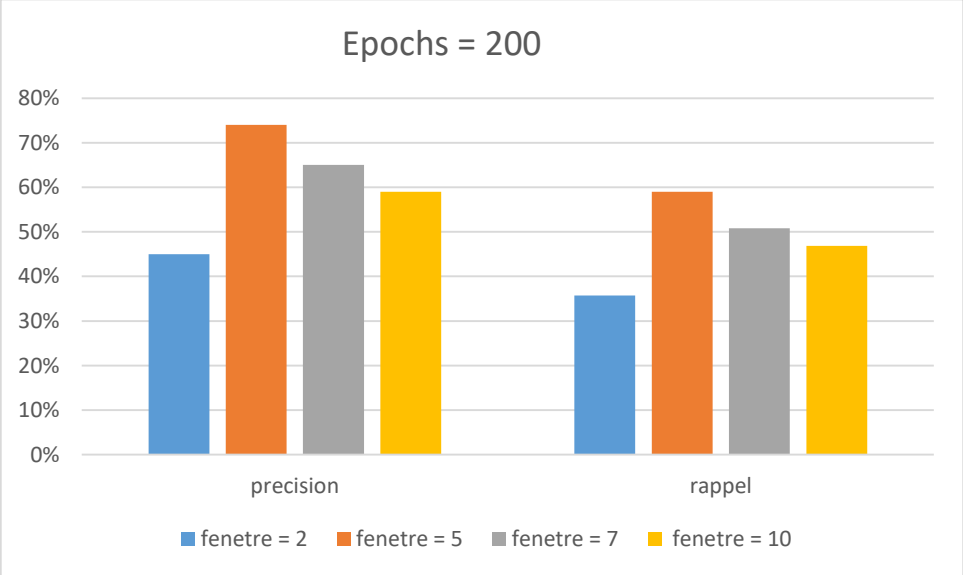


Figure 32 : Graphe montrant la moyenne de précision et de rappel (epochs = 200)

- La moyenne de précision et de rappel pour epochs = 300 :

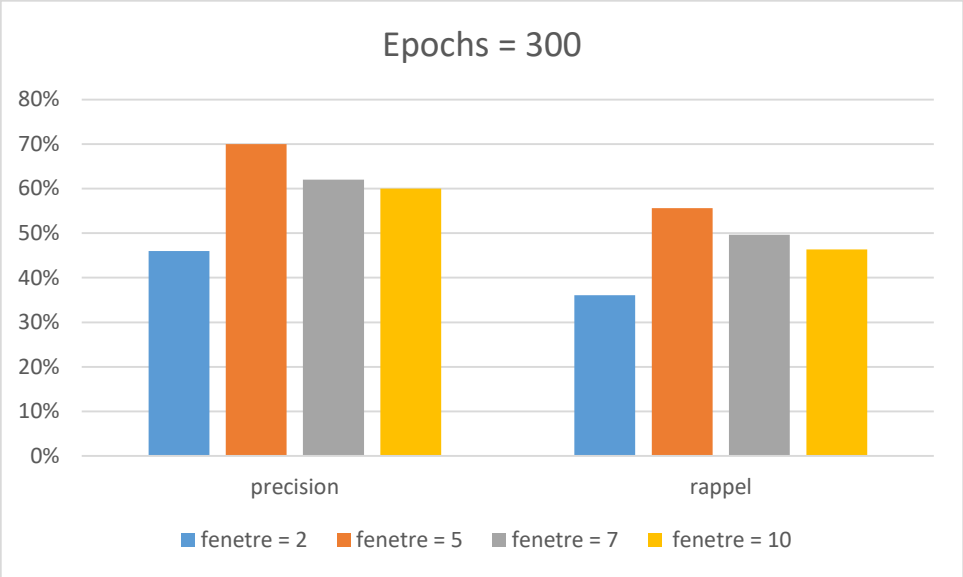


Figure 33 : Graphe montrant la moyenne de précision et de rappel (epochs = 300)

D'après l'analyse de ces graphes, les meilleurs taux de rappel et de précision de notre système sont obtenus pour la configuration suivante de notre modèle :

Taille de la fenêtre = 5, Epochs = 200

Conclusion :

Dans ce chapitre nous avons présenté plusieurs tests sur notre approche, on a trouvé sa meilleure configuration étant donné notre corpus. Cette configuration a donné des résultats pertinents avec une moyenne de précision et de rappel : 74%, et 59% respectivement.

Conclusion générale

Le traitement automatique des langues (NLP) prend de plus en plus de l'ampleur, et devient une discipline incontournable pour les différentes applications du quotidien humain. Les travaux présentés donc dans ce mémoire, se basent sur l'apprentissage automatique afin de traiter une variété linguistique qui est le dialecte arabe algérien. En effet ce dernier, est le moyen de communication utilisé dans notre pays, ce qui rend sans doute son traitement une nécessité voire même une obligation, afin de le placer par la suite dans plusieurs applications tous domaine confondus (comme l'économie pour le profilage et la collecte d'informations en langue locale).

Notre système a permis d'avoir un contexte bien défini pour chaque mot. En effet, on a réussi grâce à notre approche à construire des représentations des mots sous formes vectorielles, qui comporte et englobe les différentes relations existantes entre les mots du corpus.

Notre système a donné des résultats satisfaisant malgré le corpus de données qui est très faible, les mesures de performance utilisée pour évaluer notre système en calculant la moyenne de précision et de rappelle ont donné 74% et 59% respectivement. On estime que ces résultats seront meilleurs avec un corpus de données de plus grande taille.

Notre système compte néanmoins des points non abordés et qui seront des perspectives à implémenter prochainement :

-Intégrer un algorithme de reconnaissance des entités nommées afin de détecter toutes ces dernières.

- Mettre en œuvre une méthode qui permettra de considérer tous les mots dérivés d'une même racine comme un seul mot, car en effet les mots dérivés de la même racine ont pratiquement le même sens.

Bibliographie

Bousquet, O. a. (2002). Stability and generalization . *Journal of machine learning* , 499-526.

GUELLA, N. (2011). *Emprunts Lexicaux dans les Dialectes arabes Algériens.* Riyad: King Saud University, Faculté de langues et traduction , Synergies Mondes arabe.

Guellil Imane, A. F. (2018). *Approche Hybride pour la translitérartion de l'arabizi algérien : une étude préliminaire.*

Guellil Imane, Azouaou Faical. (2006). *Analyseur Syntaxique du Dialecte Algérien dans un but d'analyse sémantique.* Alger.

Ibrahimi, K. T. (2010, juillet 08). *L'Algérie: coexistance et concurrence des langues*. Récupéré sur https://journals.openedition.org/anneemaghreb/305

Kerras Nassima, M. B. (2019). *L'arabe standard et l'algérien : une approche sociolinguistique et une analyse grammaticale.* Granada: Íkala, Revista de Lenguaje y Cultura,.

Kim, P. (2017). *MATLAB deep learning: with machine learning , Neural Network and artificial intelligence* . Library of Congress Control Number : 2017944429.

MESFAR, S. (2008). *Analyse morpho-syntaxique automatique et reconnaisance des entités nommées en arabe standart.* Université de FRACHES_COMTE .Ecole doctorale.

Palash Goyal, S. P. (2018). *Deep learning for natural language processing.* Library of Congress Control Number : 2018947502 , ISBN-13.

SAADANE, H. (2015). *le traitement automatique de l'arabe dialectalisé: aspects méthodologiques et algorithmiques. Linguistique.* Grenouble: Université Grenouble Alpes.

Tomas Mikolov, I. S. (2013). *Distributed representation of words and phrases and their compositionality.* USA: Curran Associate, Red Hook ,Nk,.

ZENNATI, J. (2020, 05 08). *L'Algérie à l'épreuve de ses langues et de ses identités : histoire d'un échec répété*. Récupéré sur journals.openedition: http://journals.openedition.org/mots/4993

Annexe 1: Paramétres Word2Vec

− **Taille du vecteur :** la taille du vecteur est directement liée avec le nombre d'information que ce dernier peut contenir. Détermine le nombre de neurones (la dimensionnalité de la représentation) dans la couche cachée du réseau.

− **min_count :** indique la fréquence minimale d'un terme pour être inclus dans le calcul

− **Taille de la fenêtre :** indique la taille du voisinage à prendre en compte. Le programme lors de son exécution va prendre X mots avant le mot cible et X mots après le mot cible, en d'autre manière la fenêtres va se glisser sur une longueur de 2X. Il faut spécifier la ''bonne'' valeur : trop faible, on risque de ne pas capter les influences croisées entre un terme et ses voisins ; trop élevée, nous risquons de diluer l'information.

− **epochs** : indique le nombre d'itérations sur la base de données. Ce paramètre va être influencé par le taille du corpus et le pas d'apprentissage, ce qui est évident car plus on a des données plus notre modèle atteint une performance maximale. Il est recommandé d'augmenté ce paramètre lorsque le corpus de données est petit.

− **Workers =** cores => ce paramètre permet d'exploiter l'ordinateur d'une manière optimisé car il envoie les traitements à tous les cœurs du processeur.

− **Alpha =** 0.01 => ce paramètre fait référence au pas d'apprentissage (*the Learning rate*). Ce paramètre est très important pour suivre l'évolution de l'entraînement de notre modèle, car en effet ce dernier est en relation directe avec la mise à jour des poids paramétrant notre réseau de neurones. Pour bien comprendre ce paramètre, détaillons un peu la formule générale de la correction des poids de notre réseau :

$W^* = W - (n^* \frac{\partial f(x)}{\partial W})$. Où : - W* est le nouveau poids mis à jour.

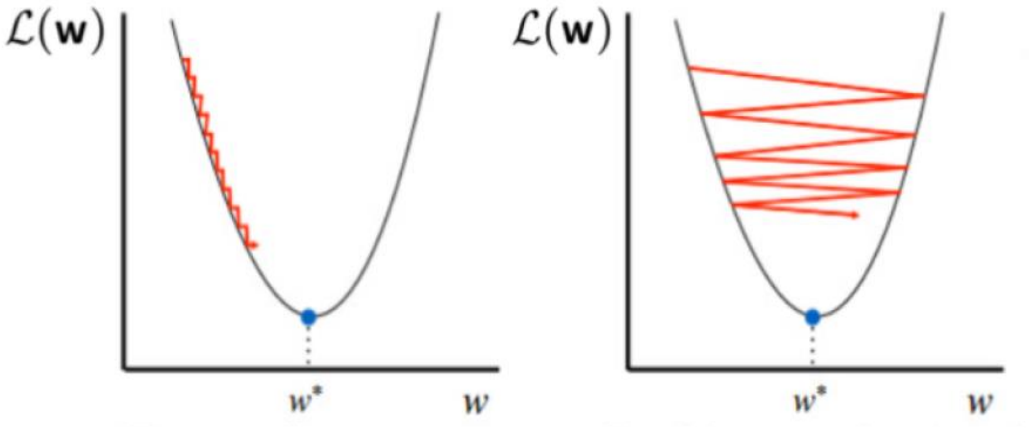
- f(x) est la fonction d'erreur.

- W est l'ancien poids.

- n est le pas d'apprentissage.

Par cette équation on pourra déduire donc que lorsque n est grand, cela influence sur le poids mis à jour (ou bien la pente calculée par les dérivées partiel), ce qui fait que notre paramètre diverge et donc un résultat erroné. Par contre lorsque ce dernier est trop petit cela rend la convergence très lente ce qui pose aussi problème.

Pour bien comprendre, voici un schéma qui résume le tout :



Très petit : Convergence très lente

Très grand : Résultat erroné et divergence

Figure 34 : Représentation du pas d'apprentissage avec différentes valeurs