الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة غرداية

Université de Ghardaïa

كلية العلوم والتكنولوجيا

Faculté des Sciences et de la Technologie

قسم الرياضيات والاعلام الالي

Département de Mathématique et d'Informatique

مخبر الرياضيات والعلوم التطبيقية

Laboratoire de Mathématiques et Sciences Appliquées

N° d'enregistrement

/ / / / /

Mémoire

Pour l'obtention du diplôme de Master

Domaine : *Mathématiques et Informatique*
Filière : *Informatique*
Spécialité : *Systèmes Intelligents pour l'Extraction des Connaissances (SIEC)*

Thème

Détection de plagiat du texte arabe à base d'incorporation de mots

Soutenue publiquement le 20 / 06 / 2022

Par
Mira MEREBBAH
Salima DINE

Devant le jury composé de :

Houssem Eddine DEGHA	**MAB**	**Université Ghardaia**	**Examinateur**
Slimane OULAD-NAOUI	**MCB**	**Université Ghardaia**	**Examinateur**
Slimane BELLAOUAR	**MCA**	**Université Ghardaia**	**Encadreur**

Année universitaire 2021/2022

Dédication

A mon père Mhammed

Tu as toujours été à mes côtés pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection.

A ma mère

Qui m'a soutenu et encouragé durant ces années d'études, qui n'a épargné aucun effort pour me rendre heureuse, je ne saurai point te remercier comme il se doit.

A ma chère petite sœur Meriem et mes frères Kaddour et Saleh, puisse Dieu vous donne santé, bonheur, courage et surtout réussite

Mon fiancé Djaber, qui n'a pas cessée de me conseiller, encourager et soutenir tout au long de mes études.

A mes cousines Bouchra, Yousra, Asma, Halima, Anfal et toute ma famille. Merci pour leurs amours et leurs encouragements.

A ma chère binôme Salima, pour sa entente, sa sympathie et ses efforts

Merci d'être toujours là pour moi.

Mira Merebbah

Dedication

Au sens de l'amour. Pour qui sa prière était le secret de ma réussite et de ma tendresse.

"Chère mère"

À mon soutien dans la vie. A celui qui m'a porté la robe de gloire et m'a comblé de son amour. A qui je porte fièrement son nom.

"Mon cher père"

À ceux qui m'ont soutenu et m'ont soutenu dans la vie, mes frères et sœurs et le mari de ma chère sœur, Mon amour la femme de mon frère.

A toute ma famille et mes proches, mes oncles, tantes, oncles, tantes et ma chère grand-mère, que Dieu vous donne longue vie

A tous ceux qui portent la famille "DINE" A mes amis "Nasira"."Asma"."Hanan"."Hasina"."Souria"."Naima"."Hasna"."Marwa".

A mes collègues de travail, vous avez été mon soutien et mon aide

À la chérie de mon cœur" Mira" la princesse la plus douce.

A mon professeur superviseur, Slimane bellaour .

A tous ceux qui m'ont connu, à tous ceux qui n'ont pas trouvé leur nom dans ma dédicace, à tous ceux qui ont atteint mon cœur et ne les ont pas écrits dans ma plume.

SALIMA DINE

Remerciements

Nous remercions tout d'abord le GRAND DIEU pour l'achèvement de ce modeste travail.

Nous remercions Mr S. BELLAOUR, notre encadreur, pour ses conseils et suggestions avisés qui nous aidés à mener à bien ce travail, et d'avoir rapporté à ce mémoire ces remarques et conseils.

Nous remercions infiniment tous le cadre enseignant de la spécialité SIEC à l'université de GHARDAIA, et spécialement à nos professeurs : Mr. Ouled naoui, Mr. Degha, Mr. Bouhanni, Mr. Mahjoub, Mr. Adjila et Mr. Moussaoui pour leurs enseignements, leurs indulgences et leurs dévouements.

Nous tenons à remercier le cadre administratif du département MI.

Enfin nos remerciements vont à tous les étudiants de notre promotion de l'année 2021/2022. Et à tous ceux qui ont contribués de près ou de loin dans ce travail.

ملخص

في الوقت الحاضر، أدى انفجار البيانات بسبب التطور التكنولوجي، من ناحية، وسهولة الوصول إلى هذا الكم الهائل من البيانات، من ناحية أخرى، إلى جعل ظاهرة الانتحال أكثر خطورة. علاوة على ذلك، تعد اللغة العربية واحدة من أكثر اللغات استخدامًا في العالم ويتزايد حضورها على الويب بشكل كبير. ومن هنا تأتي الحاجة إلى تطوير أدوات فعالة لكشف السرقات الأدبية للنص العربي. ندرس عدة تقنيات للكشف عن الانتحال في النص العربي مع التركيز على الجانب الدلالي، ولا سيما تضمين الكلمات. في منظور هذه الدراسة الببليوغرافية، نقترح نظامًا للكشف عن سرقة النص العربي يعتمد على استخدام تقنية المحولات الجديدة (AraBERT) في شبكة عصبية سيامية. من أجل تقييم نظامنا، نستخدم مجموعة ExAra. على الرغم من أن النتائج لا تزال سابقة لأوانها، فإننا نعتقد أنه يمكن تحسين نظامنا من خلال التفكير في استخدام مجموعات البيانات الأكبر الأخرى والضبط الدقيق لمحول AraBERT الخاص بنا.

الكلمات المفتاحية: كشف السرقة الأدبية، تضمين الكلمات، محول،AraBERT ، شبكة عصبية سيامية.

Résumé

De nos jours, l'explosion de données due au développement technologique, d'une part, et la facilité d'accès à cet énorme quantité de données, d'autre part, ont rendu le phénomène de plagiat plus grave. Par ailleurs la langue arabe compte parmi les langues les plus utilisées dans le monde et sa présence dans le Web croit d'une manière exponentielle. D'où la nécessité de développement d'outils efficaces pour la détection du plagiat du texte Arabe. Nous commençons par l'étude de plusieurs techniques de détection de plagiat du texte Arabe tout en se focalisant sur celles se basant sur l'aspect sémantique, notamment l'incorporation de mots (word embedding). Dans l'optique de cette étude bibliographique, nous proposons un système de détection de plagiat du texte Arabe se reposant sur l'utilisation de la nouvelle technologie des transformateurs (AraBERT) dans un réseau neuronal siamois. Dans le but d'évaluer notre système, nous utilisons le corpus ExAra. Malgré que les résultats sont encore prématurés, nous conjecturons que notre système peut être amélioré en considérant l'utilisation d'autres datasets plus larges et le réglage (fine-tuning) de notre transformateur AraBERT.

Mots clés : Détection du plagiat de texte arabe, Incorporation de mots, Transformateur, AraBERT, Réseau neuronal siamois.

Abrtract

Nowadays, the explosion of data due to technological development, on the one hand, and the ease of access to this huge amount of data, on the other hand, made the phenomenon of plagiarism more serious. Furthermore the Arabic language is one of the most used languages in the world and its presence on the Web is growing exponential way. Hence the need to develop effective tools for the detection of plagiarism of the Arabic text. We start by studying several techniques for detecting plagiarism of the Arabic text while focusing on those based on the semantic aspect, in particular word embedding. In the context of this bibliographic study, we propose a system for detecting plagiarism of the Arabic text based on the use of the new technology of transformers (AraBERT) in a Siamese neural network. In order to evaluate our system, we use the ExAra corpus. Although the results are still premature, we conjecture that our system can be improved consider the use of other larger datasets and the fine-tuning of our transformer AraBERT.

Keywords : Arabic plagiarism detection, word embedding, Transformer, AraBERT, Siamese neural network.

Table de matière

Liste des figures

Liste des tableaux

Introduction générale

De nos jours, avec le développement de la technologie, l'accès à l'information est devenu très facile, et donc le phénomène de plagiat est devenu plus dangereux. Dans le milieu universitaire, le plagiat scientifique est l'un des problèmes académiques les plus ennuyeux. en faite, c'est un acte incompatible avec l'objectif de la recherche scientifique, qui vise la promotion des domaines scientifiques. Cependant, ceci ne pourra avoir lieu qu'avec l'originalité des contributions. Dans cette optique, le ministère de l'enseignement supérieur et de la recherche scientifique a publié l'arrêté n°1082 du 27 décembre 2020 fixant les règles relatives à la prévention et la lutte contre le plagiat.

Par ailleurs, les statistiques de "Internet World Stats" de mars 2020 [1] estiment que la population mondiale de la langue arabe pour l'année 2021 s'élève à 447572891, parmi lesquels 237418349 sont des internautes avec une croissance de l'internet de 9348% entre les années 2000 et 2021. Ceci a incité la communauté des chercheurs du domaine a veiller au développement d'outils performants et efficaces pour la détection du plagiat dans le texte arabe.

Dans la littérature, la détection du plagiat du texte est abordée en considérant deux catégories, intrinsèque et extrinsèque. Le papier [Nath, 2021] se concentre sur la détection du changement de style du document en faisant recours aux réseaux de neurones siamois comportant essentiellement une couche d'incorporation de mot et une couche LSTM/GRU.

Dans la catégorie de la détection du plagiat extrinsèque, une autre contribution présentée par [Mahmoud et al., 2018] commence par le changement de représentation *Word2vec* des documents sources et suspects avant d'alimenter une architecture siamoise à base de convolution. Les auteurs de [El Moatez Billah Nagoudi et al., 2018] proposent un système de détection du plagiat du texte arabe à deux niveaux baptisé 2L-APD pour Two-Level Arabic Plagiarism Detection. Le premier niveau utilise l'empreinte digitale alors que le second implique la représentation *Word2vec* pour capturer les propriétés sémantiques du texte. Ensuite, le travail de

1. `https://www.axl.cefan.ulaval.ca/Langues/2vital_expansionINTERNET.htm`

[Mahmoud and Zrigui, 2019] commence par un changement de représentation des documents sources et suspects en combinant les techniques *Word2vec* et *Sen2vec* qui servent comme entrées dans un réseau neuronal convolutif (CNN). Finalement, le travail de [Mahmoud and Zrigui, 2021] combine à la fois l'incorporation des mots (GloVe), la convolution et le mécanisme d'attention pour la détection du plagiat du texte arabe.

Notre objectif dans ce mémoire consiste à prendre en charge l'aspect sémantique (incorporation de mots, ordre, contexte) des documents sources et suspects dans la détection du plagiat dans le texte arabe. Pour ce faire, nous proposons une architecture siamoise à base de transformateur. Dans notre cas, nous utilisons le célèbre transformateur AraBERT dédié à la langue arabe.

Pour la validation de notre système proposé, nous utilisons le corpus ExAra dédié à l'évaluation des systèmes de détection de plagiat du texte dans la tâche partagée AraPlagDet 2015. Les résultats, malgré prématurés, sont prometteuses et peuvent être améliorés si nous considérons le raffinement des différentes phases de notre système, notamment, celles liées au prétraitement du texte arabe.

Le reste du mémoire s'organise comme suit : Le Chapitre 1 comporte les concepts généraux nécessaires à la compréhension des chapitres suivants. Le Chapitre 2 se focalise sur les travaux connexes au détection du plagiat du texte arabe utilisant essentiellement l'incorporation de mots. Ensuite, notre proposition est décrite et validée dans le Chapitre 3. Enfin, la conclusion clôture notre mémoire par la présentation d'un bilan de notre travail et la suggestion des travaux futurs.

Chapitre 1

Préliminaires

Le présent chapitre est dédié à l'introduction des concepts de base abordés dans ce mémoire. Nous nous focalisons sur les notions de détection de plagiat, incorporation de mots, technologies des transformateurs et enfin sur quelques aspects de la langue arabe.

1.1 Phénomène de plagiat

Dans ce qui suit, nous mettons l'accent sur le phénomène du plagiat en général et le plagiat du texte en particulier.

1.1.1 Définition du plagiat

Le plagiat est l'utilisation d'idées, de concepts, de mots ou de structures sans reconnaître de manière appropriée la source à bénéficier dans un cadre où l'originalité est attendue ([Fishman, 2009]). Le plagiat est défini comme l'utilisation non autorisée ou l'imitation proche du langage, des idées et des illustrations de l'auteur en tant qu'œuvre originale. Cela inclut le vol littéraire, voler des paragraphes, des mots ou des idées de quelqu'un d'autre et les transmettre comme étant les sien sans en mentionner la source. Beaucoup considèrent le plagiat pour copier le travail de quelqu'un d'autre ou emprunter les idées originales de quelqu'un d'autre ([Abdelrahman et al., 2017]).

1.1.2 Définition du plagiat textuel

Le plagiat textuel, plagiat d'un texte, est un plagiat qui consiste à voler une œuvre écrite. Copier et coller tout ou partie du texte, sans citer la source, comme son propre travail. Le plagiat textuel est la forme de

plagiat la plus courante, il peut prendre de nombreuses formes différentes, copier textuellement un passage d'un livre, d'une revue ou d'une page web, résumer l'idée originale d'un auteur en l'exprimant dans ses propres mots, traduire partiellement ou totalement un texte, utiliser le travail d'une autre personne et le présenter comme le sien (et ce, même si cette personne a donné son accord), même si ces idées sont dites avec ses propres mots est un plagiat du moment où ces idées avaient déjà été mises à l'écrit ([Maurer et al., 2006]).

Les catégories plus larges de plagiat comprennent ([Maurer et al., 2006]) :

- Accidentel : en raison d'un manque de connaissances sur le plagiat et d'une compréhension du style de citation ou de référence pratiqué dans un institut
- Non intentionnel : l'immensité des informations disponibles influence les pensées et les mêmes idées peuvent sortir par des expressions parlées ou écrites comme les siennes.
- Intentionnel : un acte délibéré de copier tout ou partie du travail de quelqu'un d'autre sans donner le crédit approprié au créateur original
- Auto-plagiat : utilisation d'un travail auto-publié sous une autre forme sans faire référence à l'original

Il existe une longue liste de méthodes de plagiat couramment utilisées. Certaines de ces méthodologies incluent ([Maurer et al., 2006]) :

- copier-coller : copie mot à mot du contenu textuel.
- plagiat d'idées : utilisation d'un concept ou d'une opinion similaire qui n'est pas de notoriété publique.
- paraphrase : modification de la grammaire, mots de sens similaire, réorganisation des phrases dans l'œuvre originale. Ou reformuler le même contenu dans des mots différents.
- plagiat artistique : présenter le travail d'autrui à l'aide de différents médias, tels que le texte, l'image, la voix ou la vidéo.
- plagiat de code : utilisation de code de programme, d'algorithmes, de classes ou de fonctions sans autorisation ni référence.
- liens vers des ressources oubliés ou périmés : ajout de guillemets ou de références mais ne fournissant pas d'informations ou de liens à jour vers des sources.
- pas d'utilisation correcte des guillemets : incapacité à identifier des parties exactes du contenu emprunté.
- désinformation des références : ajout de références à des sources originales incorrectes ou inexistantes.
- plagiat traduit : traduction de contenu multilingue et utilisation sans référence à l'œuvre originale.

1.2 Détection du plagiat

La présente section est consacrée aux types de détection de plagiat intrinsèques et extrinsèques, ainsi qu'aux mesures de similarité utilisées dans le domaine. Elle se clôture en mentionnant quelques logiciels de détection de plagiat.

1.2.1 Détection de plagiat intrinsèque

Le plagiat intrinsèque vise à déterminer la possibilité de plagiat en analysant le document pour des changements subtils dans le style ([Alzahrani et al., 2012]).

L'analyse intrinsèque du plagiat est étroitement liée à la vérification de la paternité : l'objectif de la méthode était de déterminer la possibilité de plagiat en analysant un document pour des changements subtils dans l'orthographe ([Eissen and Stein, 2006]).

La détection du plagiat de façon intrinsèque utilise des approches dites stylo-métriques. Ces dernières suggèrent qu'en analysant les caractéristiques d'un texte, on peut en reconnaître l'auteur, et ainsi, si un passage du document ne possède pas les mêmes caractéristiques que le reste du document, on peut en déduire que ce passage aura été emprunté à un autre auteur ([Ferrero, 2017]).

1.2.2 Détection de plagiat extrinsèque

Dans la détection du plagiat extrinsèque, un document suspect donné est comparé à un corpus ou un ensemble de sources de référence préexistantes. Cette collection de référence peut être en ligne ou hors ligne, c'est-à-dire des sources en ligne sur le WWW ou une base de données hors ligne dans laquelle les documents sources sont stockés ([Gupta et al., 2016])

Chaque document suspect d'entrée est vérifié par rapport aux sources disponibles pour détecter s'il a été copié ou manipulé à partir de l'une de ces références (Figure 2.1).

La tâche de détection de plagiat extrinsèque est souvent découpée en deux sous-tâches, la tâche de collecte de documents sources candidats et la tâche de comparaison entre le document suspect en cours d'analyse et chacune des sources renvoyées par la tâche de collecte.

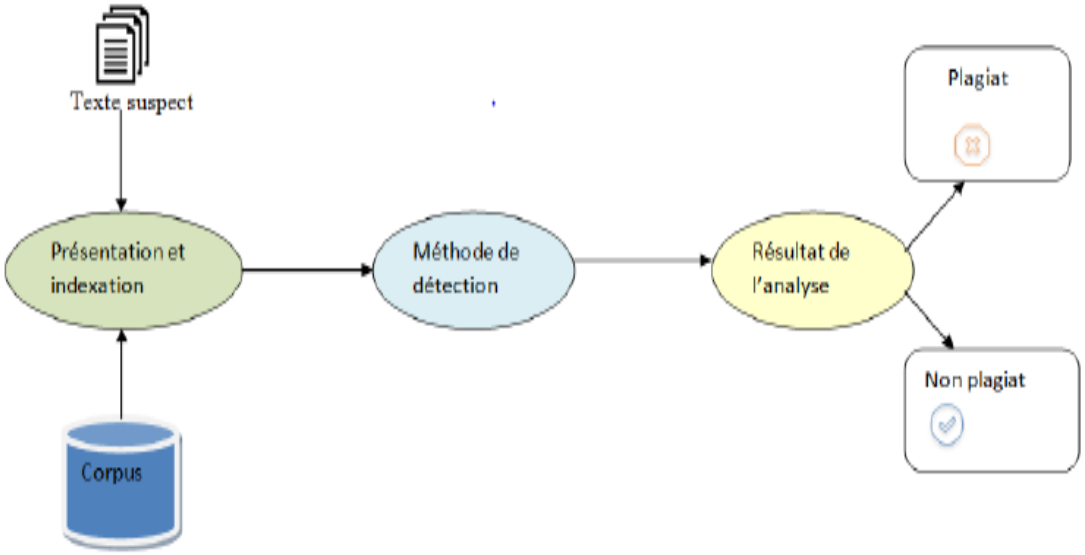

FIGURE 1.1 – Architecture générale de la détection de plagiat extrinsèque

1.2.3 Différences entre la détection de plagiat extrinsèque et intrinsèque

La Table représente les principales différences entre la détection de plagiat extrinsèque et intrinsèque :

plagiat extrinsèque	**plagiat intrinsèque**
Besoin d'une base de donné externe	Ne nécessite pas une base de donné externe
Vérifier mot à mot ou passage à passage les documents suspects avec documents sources	Analyse la stylo-métrie d'un document avec lui-même dans

TABLE 1.1 – Différences entre la détection de plagiat extrinsèque et intrinsèque

1.2.4 Mesures de similarité

La mesure de similarité est une métrique permettant de calculer la similarité des objets. L'objectif dans ce qui suit est de savoir comment identifier le taux de similarité entre les mots ou les phrases. Donc, la similarité est calculée en utilisant la fréquence des termes et la fréquence inverse des documents ou des techniques plus avancées décrites dans cette section.

Similarité Cosinus

La similarité en cosinus est une mesure de similarité entre deux vecteurs non nuls. Elle détermine le cosinus de leur angle. La valeur d'un cosinus est comprise dans l'intervalle $[-1, 1]$. La valeur de -1 indique des vecteurs opposés, la valeur de 0 des vecteurs indépendants (orthogonaux) et la valeur de 1 des vecteurs colinéaires de coefficient positif. Les valeurs intermédiaires permettent d'évaluer le degré de similarité. Étant donné deux vecteurs S1 et S2, leur similarité est calculée par l'Équation 1.1 :

$$\cos(t_1, t_2) = \frac{< t_1, t_2 >}{||t_1||.||t_2||} \tag{1.1}$$

où $< t_1, t_2 >$ est le produit scalaire des vecteurs t_1 et t_2 et $||t||$ est la norme du vecteur t. La Figure 1.2 illustre la similarité cosinus des vecteurs t_1 et t_2.

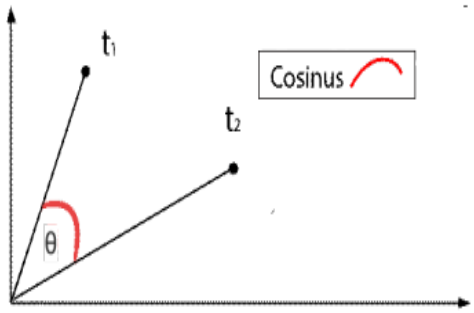

FIGURE 1.2 – Similarité cosinus entre deux vecteurs.

Distance euclidienne

La distance euclidienne est une mesure standard pour les problèmes de géométrie. C'est la distance normale entre deux points. En mesurant la distance entre les documents textuels, la distance est calculée à l'aide des vecteurs TF-IDF des documents. Étant donné que deux documents d_a et d_b sont représentés, respectivement, par leurs vecteurs de terme t_a et t_b, la distance euclidienne des deux documents est déterminée par l'Équation 1.2 ([Belattar, 2021]) :

$$D_e(d_a, d_b) = \sqrt{\sum_{t=1}^{m} |t_a - t_b|^2} \tag{1.2}$$

où m est la taille du vocabulaire.

Distance Manhattan

Nommée ainsi car elle sert à « mesurer » la distance parcourue par une voiture dans la ville de Manhattan. Elle est la plus simple. La distance de Manhattan calcule les différences absolues entre les coordonnées d'une paire d'objets. l'Équation 1.3 peut être généralisé en définissant la distance de Manhattan entre a et b comme [Vadivel et al., 2003] :

$$MH(a,b) = \sum_{i=1}^{n} |x_i - y_i| \tag{1.3}$$

Distance Jaccard

La similarité Jaccard mesure la similarité entre deux ensembles de données pour voir quels membres sont partagés et distincts. Elle est calculée en divisant le nombre d'observations dans les deux ensembles par le nombre d'observations dans l'un ou l'autre ensemble. En d'autres termes, la similarité de Jaccard peut être calculée comme la taille de l'intersection divisée par la taille de l'union de deux ensembles,comme le montre dans l'Équation 1.4 ([Niwattanakul et al., 2013]) :

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1.4}$$

Où |A ∩ B| donne le nombre de membres partagés entre les deux ensembles et |A ∪ B| donne le nombre total de membres dans les deux ensembles (partagés et non partagés). La similarité Jaccard sera de 0 si les deux ensembles ne partagent aucune valeur et de 1 si les deux ensembles sont identiques. De plus, cette fonction peut être utilisée pour trouver la dissemblance entre deux ensembles en calculant est déterminée par l'Équation 1.5 :

$$d(A,B) = 1 - J(A,B) \tag{1.5}$$

Distance de déplacement des mots

Une mesure appelée distance de déplacement des mots (Word Mover's Distance, WMD) [Kusner et al., 2015] permet de surmonter les problèmes liés aux mesures de distance traditionnelles. Elle utilise l'encodage dans un plongement lexical (*Word2vec*) pour comprendre la similarité ou la distance entre les mots. La distance entre deux mots w_0 et w_1 est calculée par l'Équation 1.6 :

$$c(w_0, w_1) = ||v(w_0) - v(w_1)|| \tag{1.6}$$

où la fonction v prend un mot et renvoie le vecteur correspondant à partir d'une matrice de word embeddings.

Cette distance C est appelée coût de déplacement du mot, qui est un élément constitutif des distances entre les documents. Soit T une matrice de flux où (T_{ij}) indique la quantité de mots i dans un document d qui circule vers le mot j en d' [Kusner et al., 2015]. Le calcul de WMD est équivalent au problème d'optimisation suivant (Équation 1.7) :

$$\begin{aligned} &\min T \geq 0 \sum_{i,j=1}^{n} T_{i,j} c(i,j) \\ &\sum_{j=1}^{n} T_{i,j} = d_i, \forall i \in \{1\ldots,n\} \\ &\sum_{i=1}^{n} T_{i,j} = d'_j, \forall j \in \{1\ldots,n\} \end{aligned} \tag{1.7}$$

Où $d_k = \frac{tf(k,d)}{\sum_{i=1}^{n} tf(i,j)}$ et $tf(k,d)$ est le terme fréquence du mot k dans le document d.

1.2.5 Quelques logiciels de détection du plagiat

En terme de plagiat, la technologie a facilité l'accès à l'information et de faire le copier-coller et à la fois elle la rendue aussi beaucoup plus facile à le détecter grâce aux programmes de détection de plagiat en ligne, bien qu'il existe d'autres versions gratuites [Mahmoud and Zrigui, 2017]. Ci-après les plus célèbres :

PlagScan

Le système est un système de détection plagiat destiné aux universités, écoles et entreprises. Pour l'utiliser, un compte payant doit être ouvert. En cas d'insatisfaction avec ce service, l'adhésion peut être annulée et l'argent remboursé. [Chowdhury and Bhattacharyya, 2018].

Turnitin

Il s'agit d'un autre outil Web puissant proposé par iParadigms. L'utilisateur doit télécharger un test dans la base de données, puis ce système crée une empreinte digitale du document et le stocke. Là où il est détecté et un rapport généré à distance, il est considéré comme l'un des meilleurs jeux de dames pour les enseignants. ([Chowdhury and Bhattacharyya, 2018]).

PlagTracker

Le vérificateur de plagiat pour les étudiants, les enseignants et les éditeurs dispose d'une grande base de données et fournit un rapport détaillé sur le travail autorisé. Cet outil s'est avéré utile pour vérifier si le papier de test est plagié ou non ([Chowdhury and Bhattacharyya, 2018]).

Quetext

utilise le traitement du langage naturel et l'apprentissage automatique pour détecter le plagiat. il est Il effectue d'abord une analyse interne pour vérifier le plagiat, puis une analyse externe est effectuée. Cet outil gratuit Le principal inconvénient de cet outil est qu'il ne fournit pas de rapport détaillé. Il n'est pas facile à utiliser. ([Chowdhury and Bhattacharyya, 2018]).

Copyscape

Il s'appuie principalement sur une adresse URLcomme entrée et recherche des copies du Web. Il aide également à trouver les fichiers des sites Web qui ont copié la page sans avoir à le faire, ce système a une version gratuite. ([Chowdhury and Bhattacharyya, 2018]).

Malgré l'étude approfondie des avantages et des inconvénients du vol littéraire, tout cela montre qu'il n'existe aucun outil capable de détecter ou de prouver l'existence du document 100% plagié, car chaque programme et outil a des avantages, des inconvénients et il n'est pas adapté pour traiter toutes les sortes de plagiat qui existent.

1.3 Incorporation de mots (Word Embedding)

Le word embedding est un ensemble de techniques d'apprentissage automatique, qui visent à mapper les mots ou les phrases dans des données textuelles à un espace vectoriel, afin de capturer les informations sémantiques et syntaxiques internes [Li and Yang, 2018]. Ainsi, les mots avec des contextes similaires ont des significations similaires. L'incorporation de mots (Word Embedding) possède différentes appellations, neural Emebeddings ou prédiction based Emebeddings, ou en français représentation vectorielle du mot, représentation continue du mot [Bengio et al., 2000].

Ce modèle consiste à apprendre un réseau de neurones (feed-forward) pour estimer la probabilité du prochain mot, en s'appuyant sur la représentation continue des mots précédents. Cette représentation est apprise au fur et à mesure au moment de l'apprentissage du réseau de neurones. Cette incorporation de mots permet de transformer le langage humain en une forme numérique. L'idée principale est que chaque mot peut être converti en un vecteur de nombres à N dimensions. Bien que chaque mot reçoit un vecteur par incorporation unique, des mots similaires finissent par avoir des valeurs plus proches les uns des autres [Balikas, 2017].

Il existe plusieurs approches de Word embedding. Le premier remonte aux années 1960 et est basé sur la technologie de réduction de dimensionnalité [Harris, 1954]. Plus récemment, de nouvelles méthodes basées sur des modèles probabilistes et des réseaux de neurones, comme *Word2vec*, peuvent améliorer les performances [Abbas and Hamdad, 2020].

1.3.1 Méthode Word2vec

Word2vec est une méthode d'apprentissage automatique non supervisée qui génère une représentation distribuée de mots et de phrases dans un espace vectoriel de grande dimension [Jansen, 2017]. Word2vec utilise un réseau de neurones formé pour capturer la relation entre les éléments du langage et le contexte. Cette forme de représentation capture des informations sur les relations syntaxiques et sémantiques entre les mots et les phrases.

Word2vec possède deux architectures : le modèle CBOW (continuous bag of words) et le modèle Skip-gram (Figure 1.3).

Continous Bag-of-Word

Le modèle CBOW est l'un des techniques de word embedding. Le CBOW vise à prédire un mot étant donné son contexte. Autrement dit, les termes qui l'entourent dans une phrase, et essaye de prédire le mot en question.

Skip-gram

Le skip-gram a une architecture visant à prédire les mots du contexte étant donné un mot en entrée. Skip-Gram fait exactement le contraire de CBOW.

Le modèle CBOW prédit le mot du milieu en combinant des représentations distribuées des contextes (ou des mots environnants). Le modèle Skip-gram utilise la représentation distribuée de mot d'entrée pour prédire le contexte.

En pratique, les modèles CBOW apprennent plus rapidement, mais les modèles de skip-gram donnent généralement de meilleurs résultats. [Mikolov et al., 2013]

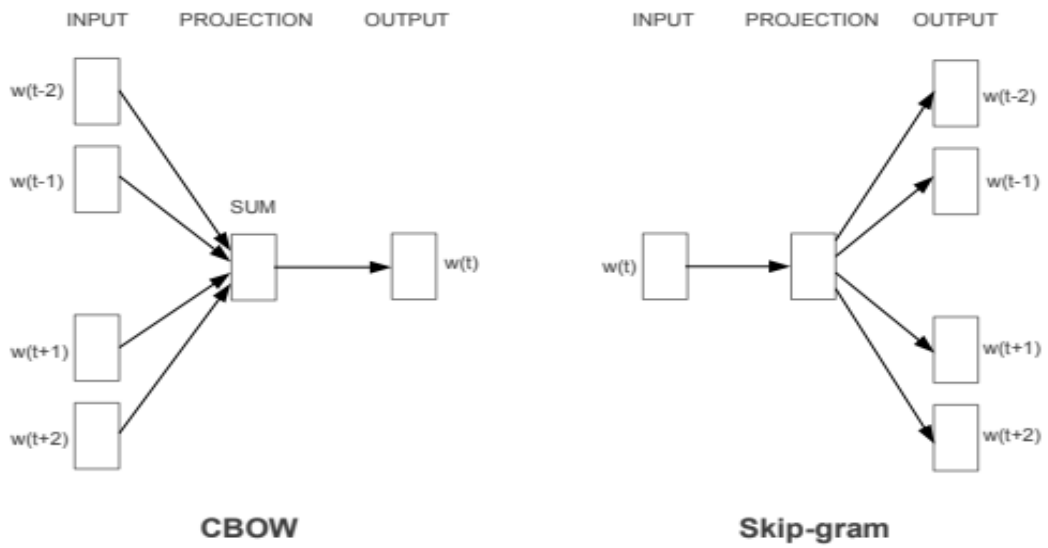

FIGURE 1.3 – Modèles CBOW et Skip-gram [Mikolov et al., 2013]

1.4 Technologie des Transformateurs

Le modèle d'encodeur-décodeur basé sur un transformateur a été introduit par [Vaswani et al., 2017], dans le célèbre papier "Attention is all you need". Aujourd'hui, l'architecture d'encodeur-décodeur est l'architecture standard dans le traitement du langage naturel (NLP).

Le transformateur est le premier modèle de transformation qui repose entièrement sur l'auto-attention pour calculer des représentations de son entrée et de sa sortie sans utiliser de RNN alignés sur la séquence ou de convolution (Figure 1.4).

1.4.1 Encodeur

Le codeur est constitué d'un empilement de N = 6 couches identiques, où chaque couche est composée de deux sous-couches :

1. La première sous-couche implémente un mécanisme d'auto-attention multi-têtes (multi-head self-attention mechanism).
2. La deuxième sous-couche est un réseau de rétroaction entièrement connecté (fully connected feed-forward network).

Chacune de ces deux sous-couches est entourée d'une connexion résiduelle. Suivie d'une normalisation de couche. Autrement dit, la sortie de chaque sous-couche est : $LayerNorm(x + Sublayer(x))$ où $Sublayer(x)$

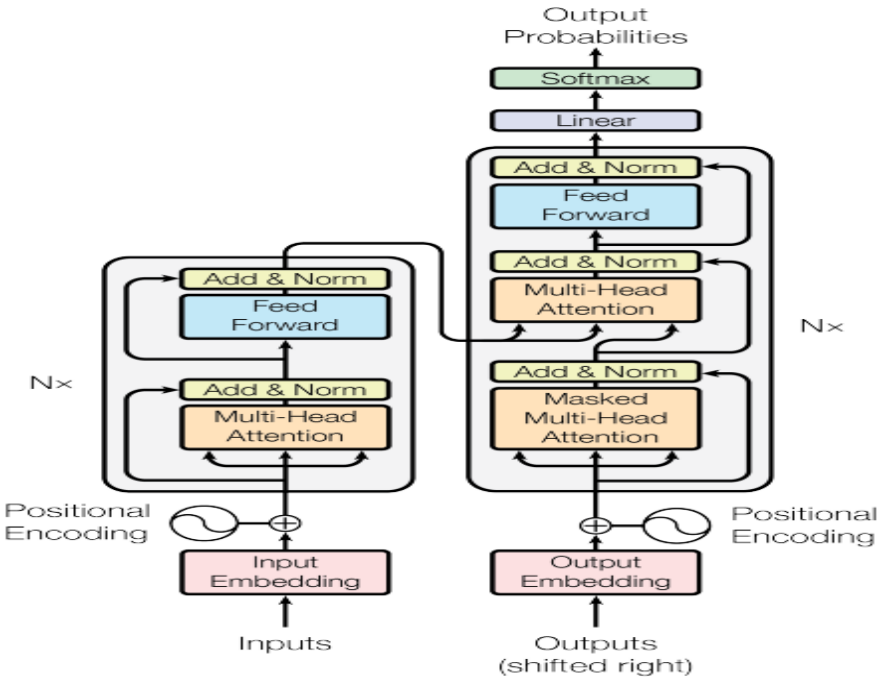

FIGURE 1.4 – Architecture du modèle de transformateur [Vaswani et al., 2017]

est la fonction implémentée par la sous-couche elle-même ([Vaswani et al., 2017]).

1.4.2 Décodeur

Le décodeur est également composé d'un empilement de N = 6 couches identiques. En plus des deux sous-couches dans chaque couche d'encodeur, le décodeur insère une troisième sous-couche, qui effectue une attention multi-tête sur la sortie de la pile d'encodeur.

Semblable à l'encodeur, des connexions résiduelles entourent de chacune des sous-couches, suivies d'une normalisation de couche. La sous-couche d'auto-attention est également modifiée dans la pile du décodeur pour empêcher les positions de s'occuper des positions suivantes ([Vaswani et al., 2017]).

1.4.3 Mécanisme d'attention

La fonction d'attention décrite comme mappant une requête et un ensemble de paires clé-valeur à une sortie, où la requête, les clés, les valeurs et la sortie sont tous des vecteurs. La sortie est calculée comme une somme pondérée des valeurs, où le poids attribué à chaque valeur est calculé par une fonction de compatibilité de la requête avec la clé correspondante ([Vaswani et al., 2017]).

1.4.4 Attention mise à l'échelle du produit

L'attention mise à l'échelle du produit scalaire (Scaled dot-product attention) est un mécanisme d'attention où l'entrée se compose de requêtes et de clés de dimension d_k, et de valeurs de dimension d_v, les produits scalaires de la requête avec toutes les clés sont calculés en divisant chacun par $\sqrt{d_k}$. Formellement, nous avons une requête Q, une clé K et une valeur V, nous calculons l'attention comme illustré dans l'Équation 1.8.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{1.8}$$

1.4.5 Attention multi-tête

L'attention multi-tête (Multi-head Attention) est un module pour les mécanismes d'attention qui parcourt un mécanisme d'attention plusieurs fois en parallèle. Les sorties d'attention indépendantes sont ensuite concaténées et transformées linéairement dans la dimension attendue.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^o \tag{1.9}$$

Où

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{1.10}$$

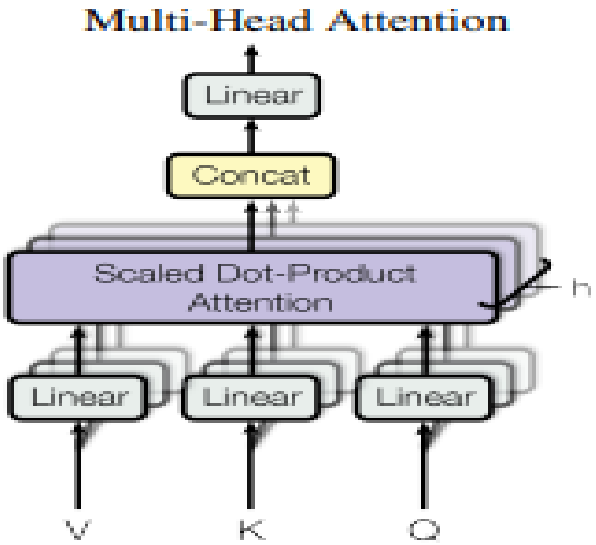

FIGURE 1.5 – Attention multi-tête consiste en plusieurs couches d'attention fonctionnant en parallèle ([Vaswani et al., 2017]).

1.5 Quelques définitions de bases

1.5.1 Réseau neuronal siamois

Un réseau de neurones siamois est une classe d'architectures de réseaux de neurones qui contiennent deux ou plusieurs sous-réseaux identiques, c'est-à-dire ils ont la même configuration avec les mêmes paramètres et poids.

Les architectures siamoises sont bonnes dans la tâche de la similarité car, lorsque les entrées sont du même type, il est logique d'utiliser un modèle similaire pour traiter des entrées similaires[Ranasinghe et al., 2019].

1.5.2 Doc2Vec

Le modèle Doc2vec est une méthode non supervisée. L'algorithme Doc2Vec est une méthode de construction de vecteurs par une transformation spatiale des paragraphes ou des documents. Ceci est une version modifiée de l'algorithme Word2vec ([Bilgin and Senturk, 2017]). Le but de Doc2vec est de créer une représentation numérique d'un document. Un texte est considéré comme un sac de mots où il n'y a plus d'ordre, et à chaque mot on associe un poids qui permet de mesurer son importance dans le texte.

1.5.3 Réseau neuronal convolutif

Dans le domaine du l'apprentissage en profondeur, CNN est l'algorithme le plus connu et le plus fréquemment utilisé. Un réseau neuronal convolutif (CNN) est un réseau neuronal d'apprentissage en profondeur conçu pour traiter des tableaux structurés de données telles que des images. Il est un modèle de programmation puissant permettant notamment la reconnaissance d'images. CNN est largement utilisé dans divers domaines tels que la vision par ordinateur, le traitement du langage et la reconnaissance faciale, il est devenu l'état de l'art pour de nombreuses applications visuelles telles que la classification d'images, et il a également rencontré du succès dans le traitement du langage naturel pour la classification de texte. La structure de CNN est inspirée des neurones présents dans le cerveau humain et animal.

Les réseaux de neurones convolutifs sont très efficaces pour détecter des motifs dans l'image d'entrée, tels que des lignes, des dégradés, des cercles ou même des yeux et des visages. C'est cette propriété qui rend les réseaux de neurones convolutifs si puissants pour la vision par ordinateur. Contrairement aux algorithmes de vision par ordinateur antérieurs, les réseaux de neurones convolutifs peuvent fonctionner directement sur une image brute et ne nécessitent aucun prétraitement.

Les réseaux de neurones convolutifs contiennent de nombreuses couches convolutives (Figure 1.6) empilées les unes sur les autres, chacune capable de reconnaître des formes plus sophistiquées. Avec trois ou quatre couches convolutives, il est possible de reconnaître les chiffres manuscrits et avec 25 couches, il est possible de distinguer les visages humains. L'utilisation de couches convolutives dans un réseau de neurones convolutifs reflète la structure du cortex visuel humain, où une série de couches traitent une image entrante et identifient des caractéristiques de plus en plus complexes.

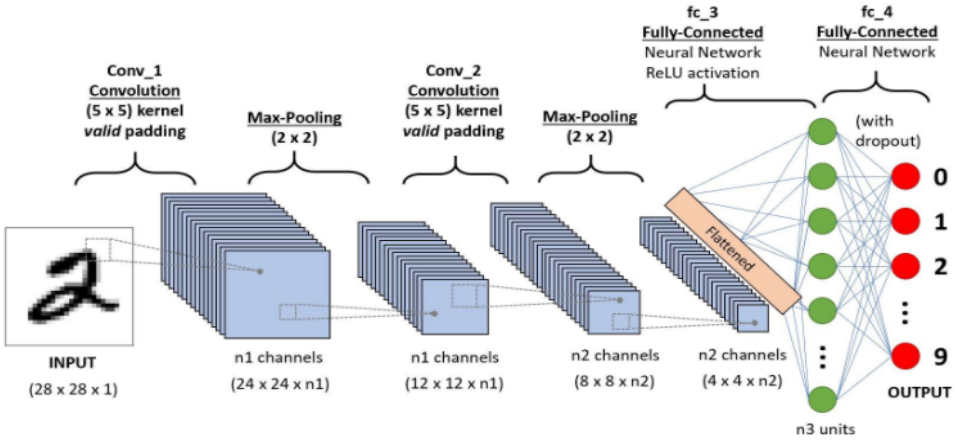

FIGURE 1.6 – Les différentes couches du CNN.

1.5.4 Glove

GloVe est un modèle d'incorporation de mots qui a été proposé par [Pennington et al., 2014]. Le modèle est un algorithme d'apprentissage non supervisé pour obtenir des représentations vectorielles du mots. GloVe signifient Global Vectors, il capture les statistiques du corpus global directement à partir du modèle, au lieu de dépendre des fenêtres de contexte locales comme le modèle word2vec. De plus, Glove utilise efficacement les statistiques en entraînant le modèle sur le nombre global de cooccurrences mot à mot. Il se compose de deux étapes : la première étape consiste à utiliser le corpus d'apprentissage pour obtenir la matrice de cooccurrence X. Dans la matrice X, le symbole X_{ij} représente la fréquence d'occurrence des mots i et j ensemble. La deuxième étape consiste à construire les vecteurs en factorisant X [Suleiman and Awajan, 2018].

1.5.5 TF-IDF

Le TF-IDF (de l'anglais terme frequency-inverse document fréquence) est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un

corpus. Il peut être défini comme le calcul de la pertinence d'un mot dans une série ou un corpus pour un texte. Le poids augmente proportionnellement au nombre de fois qu'un mot apparaît dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

1.5.6 Part of speech tagging

Part of speech tagging (POST) est une application logicielle qui lit un texte dans une langue particulière et attribue à chaque mot sa catégorie de mot ; c'est-à-dire qu'il le marque comme nom, verbe, adjectif, etc. Part of speech tagging est un processus essentiel pour comprendre comment les phrases sont formées à partir de petits constituants. Il est principalement utilisé dans l'analyse syntaxique et sémantique des phrases ([Al-Thwaib et al., 2020]).

1.5.7 Empreinte digitale (Fingerprinting)

L'empreinte digitale en informatique, est un algorithme qui mappe un élément de données arbitrairement volumineux (tel qu'un fichier informatique) à une chaîne de bits beaucoup plus courte, son empreinte digitale, est une liste d'entiers résultant du hachage des sous-chaînes du document identifie de manière unique les données ([El Moatez Billah Nagoudi et al., 2018]). Cette empreinte peut être utilisée à des fins de déduplication des données. Les fonctions d'empreintes digitales peuvent être considérées comme des fonctions de hachage hautes performances utilisées pour identifier de manière unique des blocs de données substantiels où les fonctions de hachage cryptographique peuvent être inutiles.

1.5.8 n-gram

Les n-grammes d'une chaîne sont les parties ou sous-chaînes de longueur n construite à partir d'une séquence donnée. Les formes populaires de n-grammes incluent bi-grammes (2 mots), tri-grammes (3 mots) et quatre-grammes (4 mots) ([Al-Thwaib et al., 2020]).

1.6 Langue arabe

La langue arabe est l'une des langues les plus anciennes, c'est la langue du Coran pour les musulmans. La langue arabe appartient au groupe des langues afro-asiatiques. Elle a beaucoup de spécificité qui la rend très différente des autres langues indo-européennes ([Menai, 2012]). L'arabe est la cinquième langue la plus utilisée au monde et c'est la langue maternelle de plus de 200 millions de personnes et de plus de 450 millions

de locuteurs[Zrigui et al., 2016].

La langue arabe a de nombreux problèmes, un gros dictionnaire et tant de synonymes [Zaher et al., 2018]. Le mot a différentes significations avec une position différente dans la phrase et un temps différent selon le diacritique [Mahmoud et al., 2018]. L'arabe est une langue difficile qui reste à la hauteur du niveau de recherche et d'expérimentation en raison de la grande variabilité des caractéristiques morphologiques et typographiques de l'écriture arabe [Meddeb et al., 2016].

Les principales caractéristiques de la langue arabe sont les suivantes :

- La langue arabe appartient au groupe linguistique sémitique
- Elle s'écrit de droite à gauche.
- La langue arabe a vingt-huit lettres de l'alphabet. Trois d'entre eux sont des voyelles longues et les autres sont des lettres consonantiques ([Menai, 2012]).
- Une particularité de la langue arabe est que les lettres changent de forme en fonction de leur emplacement dans le mot.
- Les différentes formes des lettres : isolée, initiale, médiane ou finale.
- Un avantage de l'alphabet arabe par rapport aux autres alphabets est qu'il n'inclut pas de majuscules.
- Une certaine difficulté est que les voyelles courtes sont remplacées en arabe par des signes diacritiques, symboles souvent omis dans les textes, rendant difficile la détermination du sens du mot.

1.7 Conclusion

Le premier chapitre est consacré pour la l'introduction du phénomène du plagiat et les méthodes de détection automatique, à savoir, extrinsèques et intrinsèques. Ensuite nous avons décrit les différents logiciel qui existe. Le chapitre comprend aussi les concepts liés à l'incorporation de mots, technologies des transformateurs et enfin quelques aspects de la langue arabe.

Chapitre 2

État de l'art

2.1 Introduction

Dans ce chapitre, nous discutons quelques études dans le domaine de détection de plagiat, en se focalisant sur le texte Arabe. La détection du plagiat pour les textes Arabes, se présente en deux catégories, la détection du plagiat intrinsèque et la détection du plagiat extrinsèque.

2.2 Travaux connexes

2.2.1 Détection du plagiat intrinsèque

Dans [Nath, 2021], l'auteur considère les propriétés intrinsèques d'un document. Autrement dit, il se concentre sur la détection du changement de style du document. Il offrent une approche globale du réseau de neurones au siamois pour apprendre les similarités stylistiques entre deux textes, même lorsque les textes sont relativement courts.

Dans la Figure 2.1 les deux paragraphes à comparer sont converties en séquences d'entiers, les entrées x_1 et x_2 sont des vecteurs de suites d'entiers représentant, respectivement, le paragraphe 1 et le paragraphe 2. La taille maximale de chaque séquence est de 300 ; les entrées sont passées à travers une couche d'incorporation GloVe (Embedding layer) à 50 dimensions qui intègre les entrées dans leurs vecteurs de mots de dimension supérieure correspondants, résultant en une forme de sortie de [(None, 300, 50)]. La couche suivante est la couche bidirectionnelle LSTM (ou GRU) contenant $n = 50$ unités avec la forme de sortie correspondante comme [(None, 100)]. Ensuite, une fonction de distance cosinus est choisie pour calculer la similarité entre

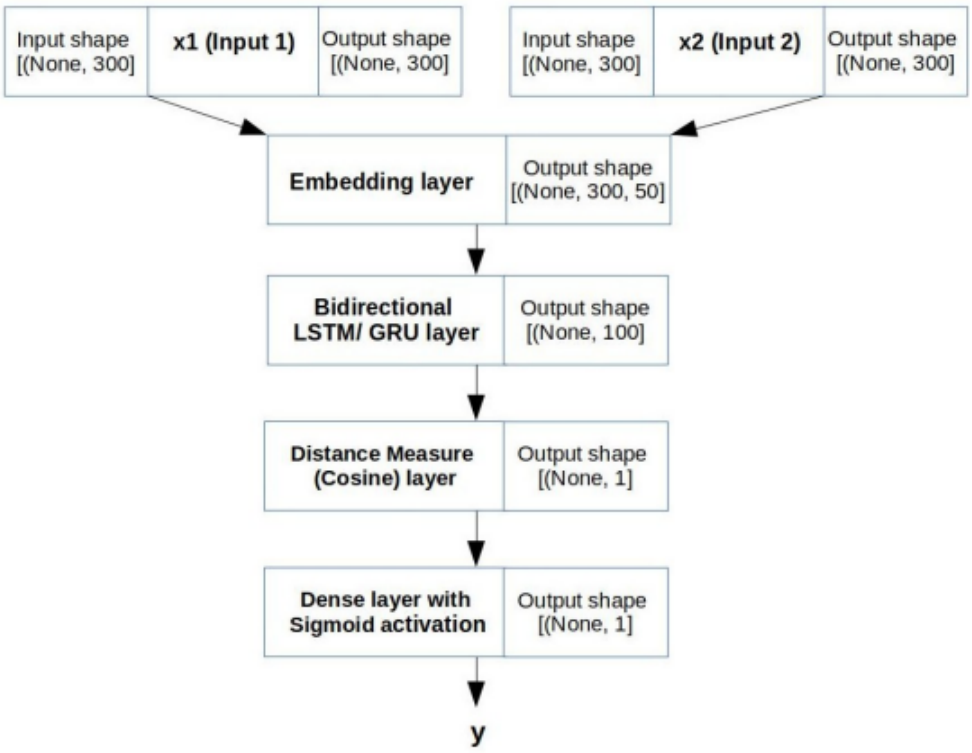

FIGURE 2.1 – Architecture du réseau siamois proposé par [Nath, 2021]

les sorties des couches bidirectionnelles LSTM (ou GRU).

La dernière couche est une couche dense avec activation sigmoïde qui produit l'étiquette y. La fonction de perte d'entropie croisée binaire (2.1) est appliquée. La valeur $y = 1$ indique un style d'écriture différent, c'est-à-dire différents auteurs, tandis que $y = 0$ témoine aucun changement de style.

$$H(p, q) = -\sum_{x} p(x) \log q(x) \tag{2.1}$$

L'évaluation de l'approche proposée est effectuée sur l'ensemble de données d'apprentissage PAN 2021 SCD qui se base sur les questions et réponses StackExchange qui ont été combinées dans des documents. L'ensemble de training contient 11200 documents, l'ensemble de validation et l'ensemble test contient chacun 2400 documents. Le nombre de changements de style varie de 0 à 20. Le nombre d'auteurs par document varie de 1 à 4.

L'évaluation est conduite sur trois modèles, à savoir SNN-LSTM, SNN-GRU et un modèle de base.

Le modèle SNN-LSTM a obtenu les meilleurs résultats en considérant les sous-tâches de validation et de test. De plus, SNN-LSTM et SNN-GRU fonctionnent beaucoup mieux que le modèle de base. Par ailleurs, le papier montre que l'architecture proposé (end-to-end siamese neural network) donne des résultats acceptables

même quand le texte est relativement court.

2.2.2 Détection du plagiat extrinsèque

Dans plusieurs recherches concernant la détection du plagiat du texte Arabe, nous trouvons le travail de [Mahmoud et al., 2018]. Ce travail se base essentiellement sur une approche de similarité sémantique entre le texte source et le texte suspect. L'approche proposée est composée de quatre phases : prétraitement, extraction des caractéristiques, détection du paraphrase et enfin, la phase d'évaluation comme illustré dans la Figure 2.2.

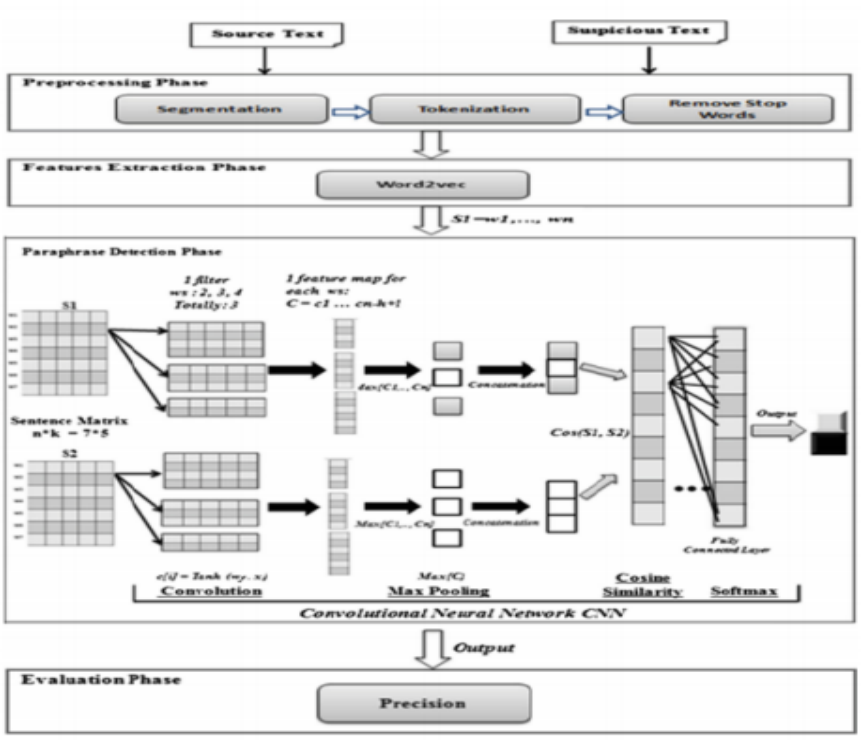

FIGURE 2.2 – Méthode proposée pour la détection de paraphrases arabes[Mahmoud et al., 2018]

La phase de prétraitement se préoccupe de la préparation des documents textes commençant par la segmentation (diviser les documents source et suspect en phrases), puis la tokenization (découpage des phrases en mots) et enfin la suppression des mots vides.

La phase d'extraction des caractéristiques se résume en la représentation *word2vec* du texte prétraité, par le bias du modèle Skip-Gram qui montre de meilleures performances en analyse sémantique. À l'issue de cette phase, une phrase S de longueur n sera représentée par une matrice W de taille $n \times k$ ($W_{1.n} = w_1, w_2, ..., w_n$), où w_i est la représentation k-dimensionnelle du mot i dans la phrase S.

Ensuite, la représentation des vecteurs de mots de chaque phrase est utilisée comme entrée dans un réseau de neurones convolutif (CNN) afin de déterminer le taux de paraphrase entre les documents sources et suspects. Le traitement est effectué à travers les couches suivantes : couche de convolution, couche de max pooling, couche de mesure de similarité et la couche entièrement connectée.

L'entrée de la couche de similarité se manifeste par deux vecteurs $S1$ et $S2$ représentant respectivement un document source et un document suspect.

La relation sémantique entre $S1$ et $S2$ se calculé par l'Èquation 2.2.

$$Cosinus(S1, S2) = \frac{< S1.S2 >}{||S1||.||S2||} \tag{2.2}$$

Dans la couche entièrement connectée, tous les résultats des opérations effectuées à chaque couche seront transmis à la couche entièrement connectée en appliquant une fonction softmax qui convertit les score de sorties en probabilités.

L'évaluation est effectuée sur un corpus Arabe Open Source OSAC [1] qui contient environ 22429 documents textes. Les résultats obtenus sont prometteurs en termes de taux de détection de paraphrase (88% de précision et 89% de rappel).

Le système 2L-APD [El Moatez Billah Nagoudi et al., 2018] (Two-Level Arabic Plagiarism Detection) est une méthode de détection du plagiat extrinsèque du texte Arabe à deux niveaux d'évaluation. Il se base essentiellement sur deux modules : module de détection à base d'empreintes digitales et module de détection à base d'incorporation de mots (Word embedding) comme illustré dans la Figure 2.3 :

— Le module de détection à base d'empreintes digitales compare simplement les empreintes digitales des documents pour détecter la reproduction textuelle.

— Le module de détection à base d'incorporation de mots (Word embedding) utilise les propriétés sémantiques et syntaxiques des mots pour détecter des reproductions beaucoup plus compliquées.

Dans leur article ([El Moatez Billah Nagoudi et al., 2018]) considèrent que le prétraitement et la segmentation comme une première étape. chaque document suspect (d_{sus}) et document source (d_{src}) est découpé en phrases. Pour ce faire, ils ont décidé d'utiliser les signes de ponctuations (.), (,), (;), (:), (!) et (?)

1. https ://www.academia.edu/2424592/OSAC$_{O}pen_{S}ource_{A}rabic_{C}orpora$

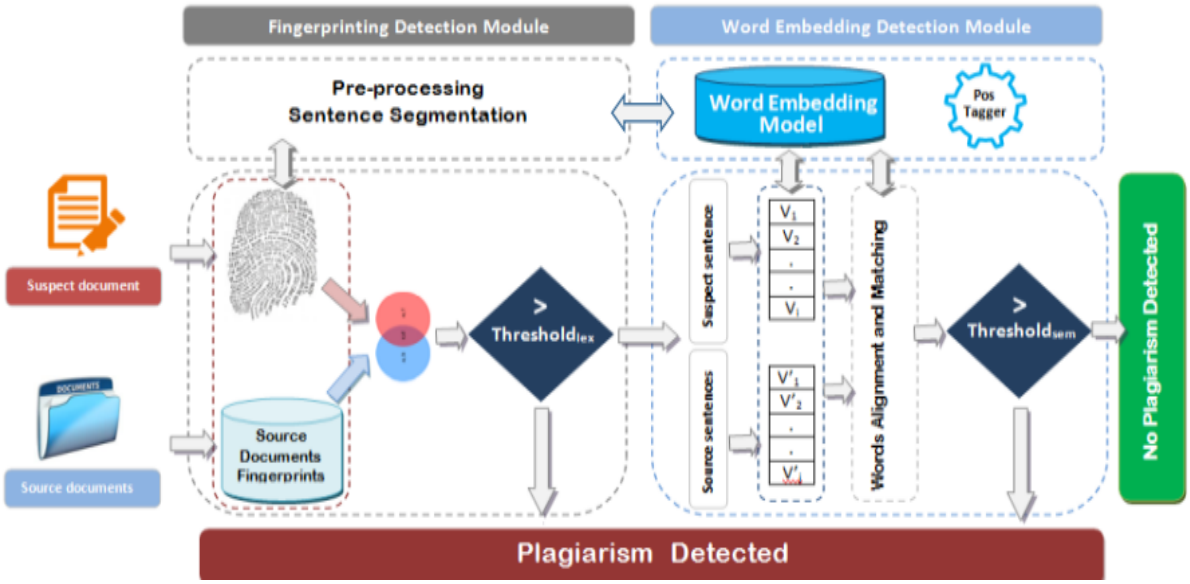

FIGURE 2.3 – 2L-APD system [El Moatez Billah Nagoudi et al., 2018]

comme points de segmentation, à condition que la longueur de la phrase compte entre 25 et 35 mots. Afin de normaliser les phrases pour les deux modules de détection, un ensemble d'étapes de prétraitement est appliqué (Tokénisation, Suppression des mots vides et Lemmatisation).

Pour chaque phrase, on construit son empreinte comme suit :

- Chunking : Chaque phrase est découpée en un ensemble de n-grams.
- Sélection
- Hachage : Application de la fonction de hachage de Brian Kernighan and Dennis Ritchie [Ritchie et al., 1988] sur les n-grams sélectionnés afin de générer l'empreinte de la phrase.

A l'issue de ces étapes on mesure la similarité entre deux documents en comparant leurs empreintes de phrases à l'aide de la similarité de Jaccard. Ensuite, la similarité calculée est comparée à un seuil fixe (T_{lex}) pour juger l'existence d'un texte partagé et suggérer un plagiat potentiel. Si la similarité est inférieure à T_{lex}, alors la phrase suspecte est envoyée au module de détection à base d'incorporation de mots pour détecter un plagiat intelligent potentiel.

Le module de détection à base d'incorporation s'occupe du niveau sémantique de détection de plagiat. En effet, on passe à une représentation vectorielle des mots Arabes en utilisant la technique CBOW. Par conséquent, la similarité entre deux phrases peut être calculée en utilisant la mesure cosinus.

Dans une perspective d'améliorer les résultats de la similarité, les auteurs font appels à la technique d'alignement de mots.

Les performances du système ont été évaluées sur un ensemble de données External Arabic Plagiarism Corpus (ExAra-2015) [2]. Les résultats obtenus témoignent que la méthode 2L-APD outrepasse tous les sytèmes participant au "Arabic Plagiarism Detection Shared Task 2015" avec un score de détection de plagiat (Plagdet) de 83%.

Le papier de [Mahmoud and Zrigui, 2019] propose une méthode de détection de plagiat du texte Arabe reposant sur une composition de l'algorithme *word2vec* et une architecture feed-forward basée sur un modèle de réseau neuronal convolutif (CNN) tel que illustré dans la Figure 2.4.

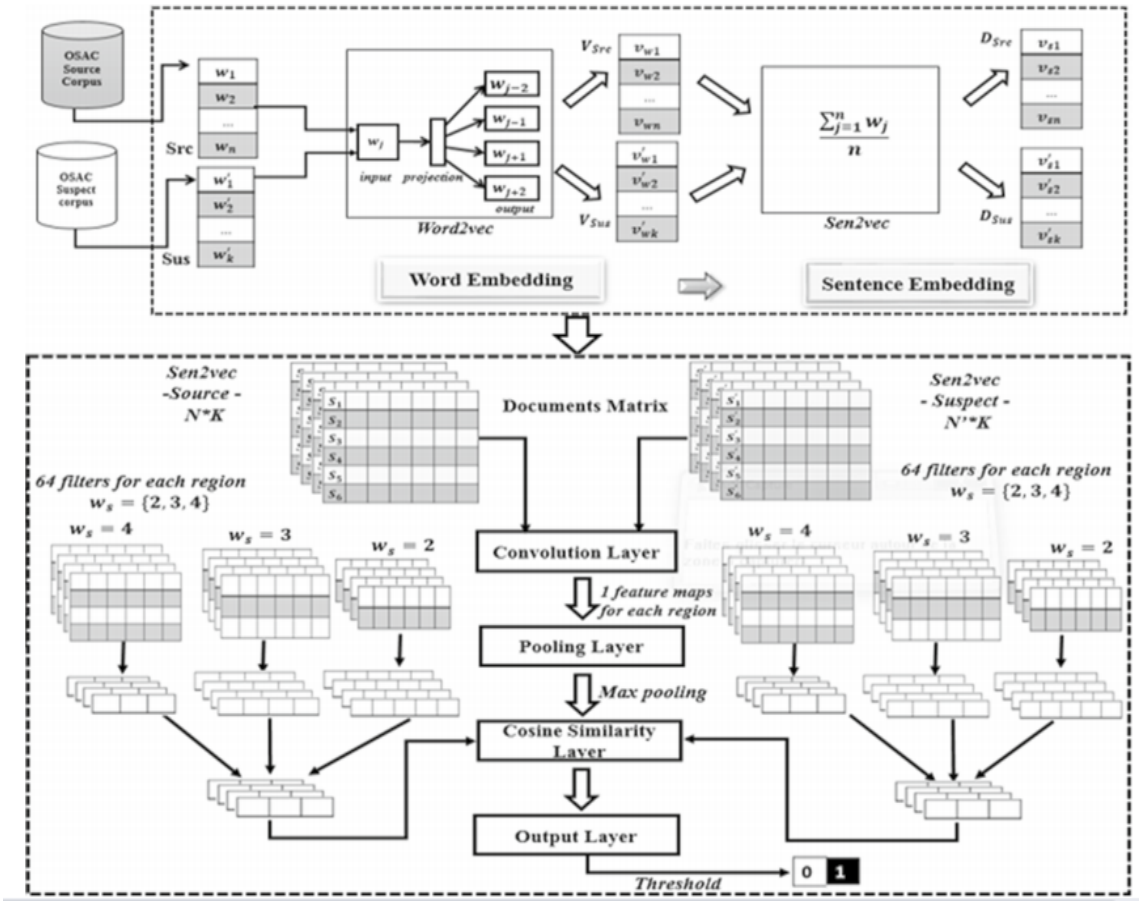

FIGURE 2.4 – Modèle de détection de plagiat du texte Arabe proposé par [Mahmoud and Zrigui, 2019]

Le système est alimenté par les documents sources et suspects, sur lesquels on applique l'algorithme *word2vec* en utilisant le modèle de skip-gram.

2. http ://misc-umc.org/AraPlagDet/

Les vecteurs résultants d'une phrase donnée S sont mappés dans une matrice $v_{w_1}, v_{w_2}, \ldots, v_{w_n}$ de taille $n \times k$, où n est le nombre de mots dans la phrase S et k est la taille des vecteurs de mots (embedding).

Dans une perspective d'améliorer la qualité d'apprentissage des phrases, les auteurs ont étendu la technique *word2vec* aux phrases pour donner naissance à une technique baptisée *Sen2vec* en calculant la moyenne de tous les vecteurs de mots à partir de la matrice associée à la phrase S (Équation 2.3) :

$$Sen2vec(S) = \frac{\sum_{i=1}^{n} v_{w_i}}{n} \tag{2.3}$$

Par conséquence, une carte de caractéristiques est obtenue de h phrases d'un document donné dans une matrice D de taille fixe $h \times k$, comme le montre l'Équation 2.4 :

$$D_{1:h} = Sen2vec(S_1), Sen2vec(S_2), \ldots, Sen2vec(S_h). \tag{2.4}$$

Ensuite, le modèle CNN est appliqué sur les incorporations de phrases de documents sources et suspects (D_{src} et D_{sus}) en tant qu'entrées. Le but de cette phase d'apprentissage consiste à extraire les caractéristiques cachées.

À l'issue de la couche pooling, chaque document est représenté par un vecteur. la couche de calcul de similarité sémantique applique une mesure cosinus pour chaque paire de documents.

Les résultats obtenus sont dans l'intervalle de $[0, 1]$. Dans le cas où ce degré est supérieur à un seuil α, les paires de documents sont considérées comme paraphrasées.

L'évaluation de l'approche proposée est effectuée sur l'ensemble de données Open Source Arabic Corpora (OSAC) en tant que corpus source dans lequel 30% de son contenu est paraphrasé de manière aléatoire.

Le système proposé basé sur la représentation vectorielle de phrases (Sen2vec) et les fonctionnalités basées sur du modèle CNN a obtenu des résultats prometteurs en termes de précision 85% et de rappel $86,8\%$.

Un autre travail de [Mahmoud and Zrigui, 2021] propose une méthode de détection des paraphrases Arabes basée une architecture de réseau de neurones siamois (Siamese Neural Network, SNN) pour extraire les caractéristiques discriminantes des documents textuels. SNN se compose de deux réseaux de neurones identiques partageant les mêmes poids. Ensuite, Les vecteurs de sortie résultants de SNN sont transmis à une fonction de calcul de distance entre eux. Cette architecture se déroule en quatre phases, à savoir, Embedding, CNN, Attention et Similarity comme le montre la Figure 2.5.

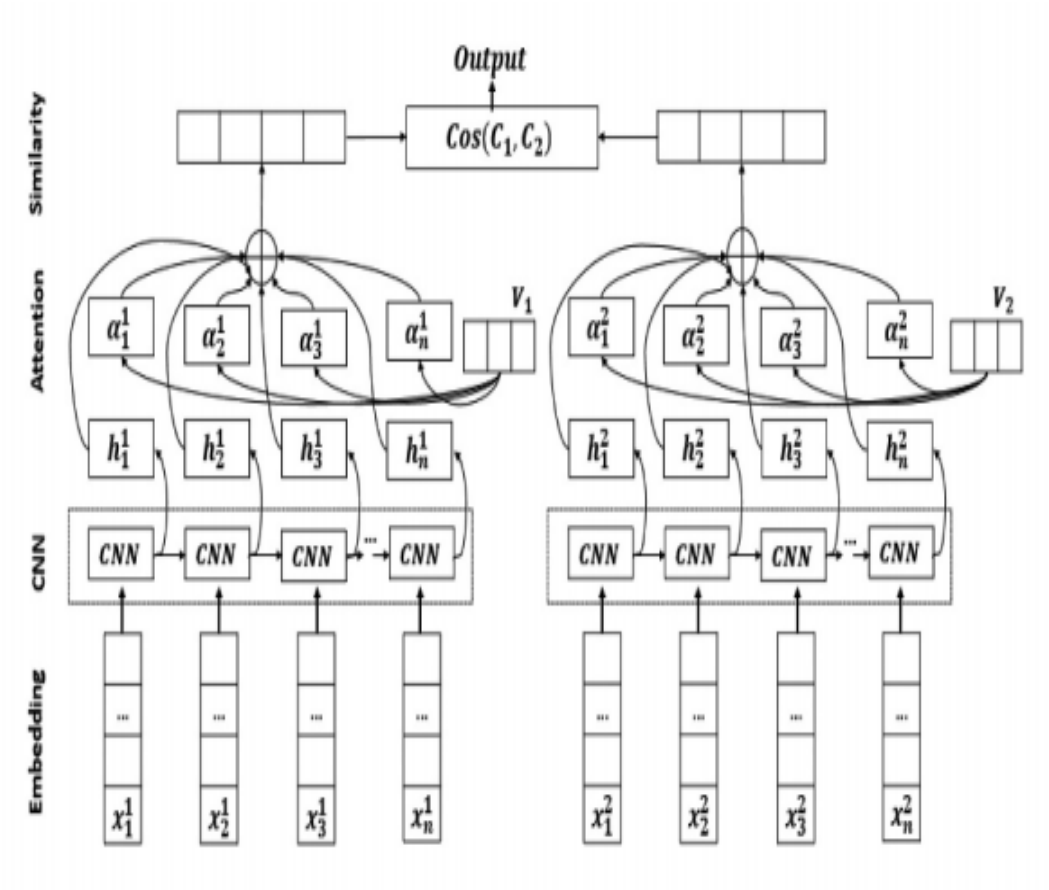

FIGURE 2.5 – Architecture de détection de paraphrases Arabes de [Mahmoud and Zrigui, 2021]

Avant d'entamer la phase d'incorporation (Embedding), des opérations de prétraitement sont effectuées (suppression des mots vides, normalisation de certaines formes d'écriture, tokénisation).

Dans la phase Embedding, la méthode GloVe est utilisée pour apprendre des informations sémantiques, en tenant compte du contexte des mots et capturer les relations contextuelles entre les mots. Rappelons que GloVe construit une matrice de co-occurrences mot-mot en estimant la probabilité d'apparition d'un mot dans le contexte d'un autre mot.

Les vecteurs sortants de la méthode GloVe sont considérés comme des entrées dans la couche CNN. Elle capture des caractéristiques contextuelles invariantes grâce à des couches convolutionelles, de pooling et entièrement connectées. Elle est utilisée pour extraire les caractéristiques les plus influents dans le texte.

La sortie de la couche pooling est k-cartes de caractéristiques pertinentes et utiles qui sont ensuite concaténées pour améliorer la capacité de généralisation du modèle.

À l'issue de la phase CNN, Le mécanisme d'attention est appliqué pour mettre en évidence les mots importants représentant la phrase sous la forme d'une représentation vectorielle.

Formellement, ce mécanisme d'attention est concrétisé par :
e_t est une représentation cachée obtenus à l'aide d'un état cachés h_t alimenté à travers un réseau entièrement connecté, Il est basé sur une fonction tangente hyperbolique. Il aide à identifier les caractéristiques pertinentes en leur générant des scores élevés (2.5).

$$e_t = \tanh\left(W_h h_t + b_h\right), e_i \in [-1, 1] \tag{2.5}$$

Ensuite, a_t sont des poids d'attention représentant la pertinence de chaque mot dans la phrase.ils ont calculés en appliquant la fonction Softmax sur les scores, comme le montre l'Équation 2.6.

$$a_t = \frac{\exp\left(e_t^T u_h\right)}{\sum^{t=1} T \exp\left(e_t^T u_h\right)} \tag{2.6}$$

Enfin, une représentation fixe r de la phrase entière est calculée comme la somme pondérée de tous les mots en utilisant les poids d'attention a_t, comme le montre l'Équation 2.7 :

$$r = \sum_{t=1}^{T} a_t h_t, r \in R^{2L} \tag{2.7}$$

Où W_h et b_h sont les poids des couches.

Enfin, la similarité cosinus est utilisée pour calculer la similarité sémantique entre les deux vecteurs cachés obtenus. La similarité cosinus prend des valeurs entre $[-1, +1]$ qui sont converties en probabilités.

Pour la détection de paraphrase Arabe, la sortie de la similarité cosinus est comparé avec un seuil $\beta = 30\%$, si la sortie est inférieur ou égal à β alors les phrases sont similaires.

L'expérimentation de l'approche proposée est menée sur le corpus source OSAC. Sa qualité est validé en utilisant le benchmark SemEval 2017. Il contient un corpus monolingue arabe paraphrasé comprenant 250 paires de phrases avec leurs scores de similarité sémantique.

Les expériences globales ont démontré que l'application de GloVe avec CNN attentionnel améliore les résultats avec une précision de $0,775$, un rappel de $0,745$ et un score F1 de $0,759$ en utilisant l'ensemble de données SemEval. Ces valeurs sont encore augmentées en utilisant le corpus OSAC avec une précision de $0,805$, un rappel de $0,785$ et un score F1 de $0,794$.

2.3 Conclusion

Dans ce chapitre, nous avons étudié quelques articles de l'état de l'art sur la détection de plagiat des textes Arabes. La totalité des articles étudiés conviennent que la phase de prétraitement est cruciale pour toutes les tâches du traitement automatique de la langue, en particulier, la détection du plagiat. Cette phase consiste à normaliser, segmenter le texte source et le texte suspect en phrases et extraire les tokens du texte. Pour mener notre étude, nous avons considéré les deux types de détection de plagiat, à savoir, intrinsèque et extrinsèque. Concernant la détection du plagiat de texte intrinsèque, et vu la rareté des travaux dans cette catégorie, nous avons étudié seulement le travail de [Nath, 2021]. Par contre pour le type de détection de plagiat extrinsèque, notre étude a visé quelques articles récents dans le domaine, nous pouvons citer les suivants : [Mahmoud et al., 2018], [Mahmoud and Zrigui, 2019] ,[Mahmoud and Zrigui, 2021],[El Moatez Billah Nagoudi et al., 2018] . Cette étude bibliographique nous a permis d'apercevoir l'état actuel de l'art dans ce domaine, les techniques utilisées et les outils.

Chapitre 3

Étude expérimentale

3.1 Introduction

Dans ce chapitre, nous proposons un système de détection de plagiat du texte arabe. Pour ce faire, nous avons adopté le modèle le plus utilisé dans ce type d'applications, à savoir, le modèle basé sur une architecture siamoise. Aussi, dans une perspective de prendre en charge l'aspect sémantique du texte (word embedding, contexte, ordre), nous avons utilisé la technologie Transformer comme une pierre d'angle dans notre architecture. Avant d'entrer dans les détails, nous commençons par décrire l'environnement matériel et logiciel ainsi que le dataset utilisé pour conduire nos expérimentations.

3.2 Environnement

Pour mettre en oeuvre notre système, nous avons utilisé la plateforme Google Colab. Elle nous a fourni une machine avec les caractéristiques suivantes :

— Processeur : GPU

— RAM : 12.68 GB

— Disque : 107.72GB

En ce qui concerne l'environnement logiciel, nous utilisons le langage de programmation Python 3 avec les bibliothèques suivantes :

— **PyTorch** : est une bibliothèque d'IA, développée par Meta, écrite en Python pour se lancer dans le deep learning et le développement de réseaux de neurones artificiels.

— **Farasa** : est une boîte à outils de traitement automatique de la langue arabe, elle est développée à l'institut de recherche informatique du Qatar. Farasa sert les tâches suivantes : segmentation, lemmatization, reconnaissance d'entité nommée (Named Entity Recognition, NER), balisage parties du discourt (Part Of Speech, POS) et diacritisation.

3.3 Dataset

Nous utilisons le corpus d'évaluation pour la détection de plagiat ExAra (EXternal ARAbic) [Bensalem et al., 2015]. Ce corpus a été utilisé dans la tâche partagée AraPlagDet 2015 [1].

Le corpus ExAra comprend 2345 documents ; près de la moitié d'entre eux (documents suspects) contiennent des passages empruntés à l'autre moitié (documents sources) pour simuler des documents contenant des fragments plagiés.

Le corpus comporte deux parties : Training et test. Chaque partie du corpus est constituée principalement de trois ensembles de données : deux ensembles de fichiers texte et un ensemble de fichiers XML. Les fichiers textes sont les documents suspects et les documents sources. Les documents XML contiennent l'annotation du plagiat, c'est-à-dire qu'ils fournissent pour chaque passage plagié son décalage de départ et sa longueur à la fois dans les documents suspects et source (le décalage et la longueur étaient tous les deux exprimés en caractères). Un fichier de document suspect (.txt) et son fichier d'annotation de plagiat (.xml) partagent le même nom.

À titre d'exemple, la Figure 3.1 montre une partie d'un document texte suspect nommé "suspicious-document0001.txt".

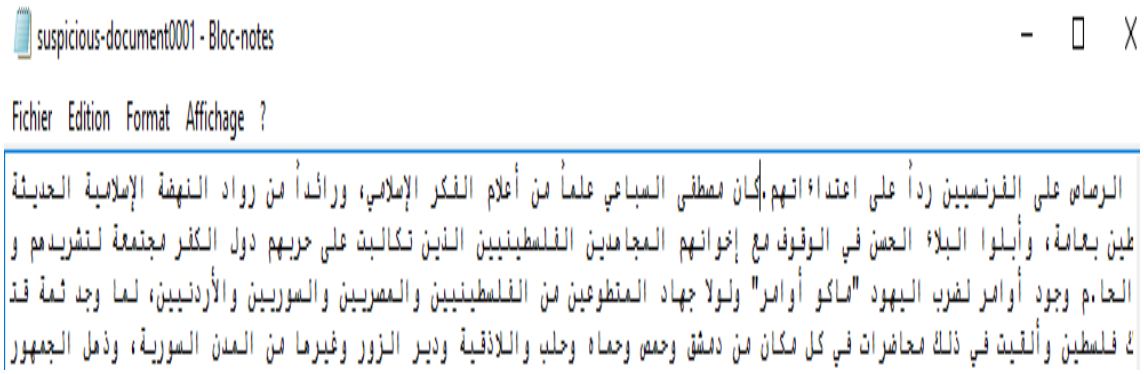

FIGURE 3.1 – Partie d'un document texte suspect

1. http ://misc-umc.org/AraPlagDet/

La Figure 3.2 contient une annotation XML sous le nom "suspicious-document0001.xml" décrivant le document texte "suspicious-document0001.txt". En effet, il révèle l'absence du plagiat dans ce dernier.

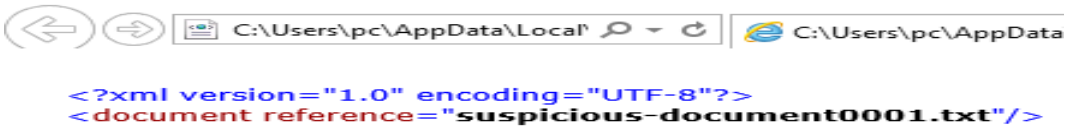

FIGURE 3.2 – Exemple de fichier XML révélant l'absence du plagiat dans le document "suspicious-document0001.txt"

Par contre, la Figure 3.3 montre un fichier XML qui indique que le fichier suspect "suspicious-document0170.txt" est plagié du document source "source-document00054.txt" avec le type du plagiat *artificial sans obscurcissement*. Il est à préciser que la partie plagiée est de longueur 198, sa position dans le document source est 1150 et dans le document suspect est 1120.

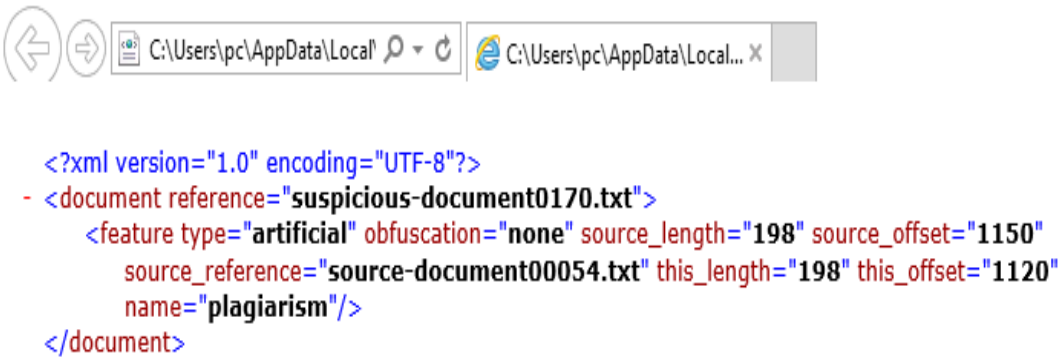

FIGURE 3.3 – Exemple de fichier XML indiquant l'existence du plagiat dans le document "suspicious-document0170.txt"

Par ailleurs, il est à noter que dans les documents XML peut figurer d'autres types de plagiat, à savoir, un plagiat artificiel avec un mélange de mots, un plagiat artificiel avec un mélange de phrases, un plagiat simulé avec substitution manuelle des synonymes et un plagiat simulé avec paraphrase manuelle. De plus, un document suspect peut plagier de plusieurs documents sources.

3.4 Modèle proposé

En s'inspirant de l'étude bibliographique élaborée dans le Chapitre 2, nous proposons un modèle faisant partie à une grande famille ayant comme occurrence réseaux *siamois* (siamese).

Nous donnons d'abord une brève description de cette famille de réseaux, puis nous décrivons le modèle que nous implémentons dans ce travail.

AraBERT [2] [Antoun et al., 2020] est un modèle de la langue arabe pré-entraîné basé sur l'architecture BERT de Google. Ce modèle est largement considéré comme la base de la plupart des résultats de l'état de l'art dans différentes tâches de NLP. AraBERT utilise la même configuration $BERT_{BASE}$ [Antoun et al., 2020]. Il est entraîné sur 23 Go de textes. Cet ensemble de données est constitué de plusieurs articles de presse et des actualités de différents médias dans différentes régions arabes.

La Figure 3.4 montre notre architecture siamoise proposée pour la détection du plagiat du texte arabe. Elle se déroule en trois phases : pré-traitement, encodage et en fin la comparaison des documents sources et suspects.

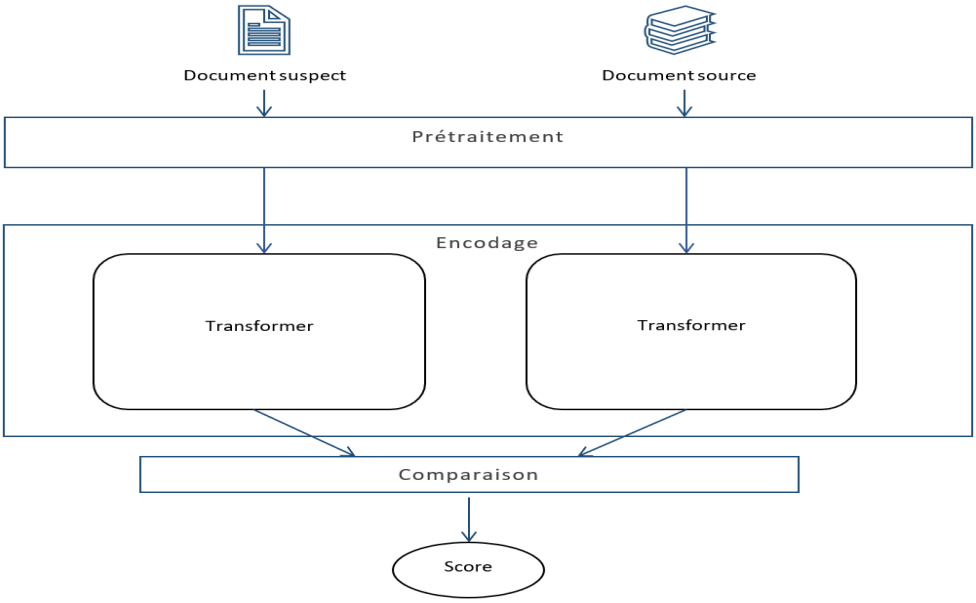

FIGURE 3.4 – Architecture siamoise proposée pour la détection du plagiat du texte arabe.

3.4.1 Pré-traitement

Dans la phase de prétraitement, nous utilisons *ArabertPreprocessor* [Antoun et al., 2020], c'est une fonction de prétraitement sur les collections de données en arabe. ArabertPreprocessor assure les tâches suivantes :

2. https ://github.com/aub-mind/arabert

— Supprimer le balisage html.

— Remplacer les mentions d'emails d'urls par des jetons spéciaux.

— Supprimer les signes diacritiques (FATHATAN, DAMMATAN, KASRATAN, FATHA, DAMMA, KASRA, SUKUN, SHADDA).

— Supprimer l'allongement.

— Insère un espace blanc avant et après tous les chiffres ou l'alphabet arabe et anglais ou les deux crochets, puis insère un espace blanc entre les mots et les chiffres ou les chiffres et les mots.

— Supprimer la répétition non numérique par remplacer la répétition de plus de deux caractères non numériques par deux de ce caractère.

— Remplacer la barre oblique "/" par un tiret "-", car "/" est absent du vocabulaire AraBERT

— Remplacer les nombres hindis par l'arabe correspondant.

— Appliquer la segmentation Farasa : Farasa est une boîte à outils NLP.

— Garder les emojis pendant le pré-traitement.

À titre d'illustration, la Figure 3.5 montre le résultat de prétraitement sur un texte arabe.

avant : سيَستمتعُون بأَحلى مَناظر السُّعب المُرجانية فى البَحر الاحمر مهما كَان سبب زيارَتهم للمكان

aprés : س+ يستمتع +ون ب+ أحلى مناظر ال+ شعب ال+ مرجاني +ة في ال+ بحر ال+ احمر مهما كان سبب زيار +ت +هم ل+ ال+ مكان

FIGURE 3.5 – Un texte arabe avant et après le prétraitement

Une fois que les documents sources et suspects sont prétraités, ils doivent être passés à la phase d'encodage.

3.4.2 Encodage

Dans cette phase d'encodage de données, nous utilisons le transformateur AraBERT dédié à la langue arabe avec la configuration suivante : 12 couches d'encodeur, 12 têtes d'attention, 768 unités cachées, ce qui donne $110M$ paramètres et 512 comme longueur maximale des séquences.

Les deux textes prétraités constituent l'entrée pour cette phase d'encodage. Un texte source et un texte suspect de longueurs respectivement n et m, tel que n et m soient inférieurs à la longueur maximale $max = 512$. L'encodeur AraBERT appliques les tâches d'incorporation de mots (word embedding), incorporation de position (position embedding) et le mécanisme d'auto-attention (self-attention) sur ses entrées pour produire en sortie des textes source et suspects avec des informations sémantiques.

À titre d'exemple, la sortie de notre encodeur AraBERT se manifeste sous forme de la matrice de la Figure 3.6 représentant l'incorporation du texte prétraité de la Figure 3.5.

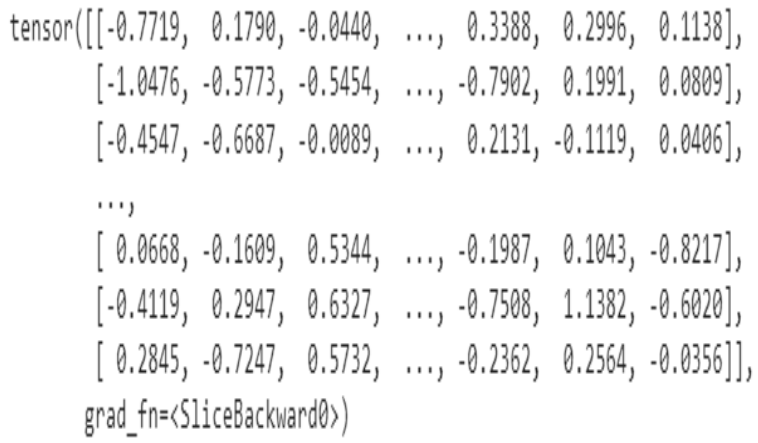

FIGURE 3.6 – Encodage AraBERT du texte prétraité de la Figure 3.5

3.4.3 Comparaison

Dans la phase de la comparaison, nous disposons de deux matrices de tailles $n \times k$ et $m \times k$ représentant respectivement les deux textes source et suspect. Ces deux représentations dérivent du modèle neuronal du réseau siamois. Nous calculons la similarité entre les textes source et suspect en comparant les deux matrices associées. Pour cela, nous utilisons le produit scalaire comme mesure de similarité. Formellement, ceci, peut être décrit comme suit :

Soit deux matrice, la matrice $W = w_1, w_2, \ldots, w_n$ de taille $n \times k$, et la matrice $V = v_1, v_2, \ldots, v_m$ de taille $m \times k$ représentant respectivement l'incorporation de textes source et suspect. Où w_i est l'incorporation de taille k du ième mot dans le texte source, et v_i est l'incorporation de taille k du ième mot dans le texte suspect.

Nous appliquons une opération de normalisation sur les deux matrices en multipliant élément par élément la matrice W par $W' = \frac{1}{||w_1||}, \frac{1}{||w_2||}, \ldots, \frac{1}{||w_n||}$, et la matrice V par $V' = \frac{1}{||v_1||}, \frac{1}{||v_2||}, \ldots, \frac{1}{||v_m||}$. Où $||w_i|| = \sqrt{< w_i, w_i >}$ et $||v_i|| = \sqrt{< v_i, v_i >}$ sont, respectivement les normes de w_i et v_i.

Ensuite, nous appliquons le produit scalaire entre W et V^T. nous obtenons à la fin une matrice de taille $n \times m$. Puis, on réduit cette matrice au vecteur Z de taille n où $z_i = \max_{1 \leq j \leq m}(< w_i, v_j >)$. Enfin, on réduit ce vecteur Z à un score, tel que $score = \frac{1}{n} \sum_{i=1}^{n} z_i$.

Le résultat de ce *score* peut être assimilé au taux de plagiat entre le texte source et le texte suspect. Nous considérons, expérimentalement, un seuil $\alpha = 0,75$ pour différencier entre plagiat et non plagiat. Si $score \geq \alpha$ alors le texte suspect est plagié.

À titre d'exemple, pour les deux séquences non plagiés de la Figure 3.7, notre modèle donne une mesure de similarité de 0, 66.

text1 = "كان مصطفى السباعي علماً من أعلام الفكر الإسلامي، ورائداً من رواد النهضة الإسلامية الحديثة. ولد في حمص عام 1915م، وتلقى فيها علومه الأولى، ثم قصد إلى الأزهر الشريف ليستكمل دراسته للعلوم الإسلامية"

text2 = "اتفق النسابون على أن عدنان من ذرية إسماعيل بن إبراهيم عليهما السلام ولكن اختلفوا في عدد الأجداد بين عدنان وإسماعيل فمنهم من قال عشرون جدا ومنهم من قال أربعون جدا ومنهم من قال أن المدة طويلة بحيث يستحيل عد الأجداد"

FIGURE 3.7 – Séquence de texte non plagiées

Par contre, le modèle a calculé une similarité de 0, 88 témoignant un plagiat dans les séquences de la Figure 3.8.

text1 = "وهي رسالة جادة وهادفة تدفع كيد خصوم الإسلام وأعدائه وتفضح مؤامرتهم وألاعيبهم! فلا تفضح أمرنا أيها السراج - جلال الدين الرومي) أحاول الآن أن أوقف الذاكرة على مشهد أو ، قصيدة, فتتداخل الصور والأحداث, الدم والمطر, النساء والشظايا, النصوص والأصداء, الشعر والإسطبل, الصحافة والمنفى."

text2 = "أرفع رأسي إلى السقف متسائلا لأرى قطرتين تنهمران على خدي المتجعد كأنهما دمعتان من السماء.. (إن هذه الأرض, وتلك السماء, مزقتا قلبي بضيقهما. فلا تفضح أمرنا أيها السراج - جلال الدين الرومي) أحاول الآن أن أوقف الذاكرة على مشهد أو قصيدة, فتتداخل الصور والأحداث, الدم والمطر, النساء والشظايا, النصوص والأصداء, الشعر والإسطبل, الصحافة والمنفى."

FIGURE 3.8 – Séquence de texte plagiées

3.4.4 Évaluation

Cette partie d'évaluation consiste à valider notre approche proposée de détection de plagiat du texte arabe. Cette validation consiste à évaluer notre système en utilisant le dataset ExAra et en calculant les mesures de précision, de rappel et le F1 score.

Nous rappelons que nous disposons de trois types de documents : les documents sources, les documents suspects dont chacun dispose d'un fichier d'annotation de plagiat en format XML tel que décrit dans la Section 3.3.

Nous commençons par le module d'évaluation du plagiat entre un document suspect et un document source. Ceci peut être réalisé ainsi :

— Segmenter le document source et le document suspect en des phrases. Techniquement, la taille de phrase ne dépasse pas 500 mots.

— Comparer chaque phrase de document suspect avec toutes les phrases du document source. Si une phrase est plagiée alors le document suspect est considéré comme plagié et la valeur retournée est 1. Sinon la valeur 0 est retournée pour témoigner un état de non plagiat.

Ensuite, nous appliquons le module du plagiat entre un document source et un document suspect pour tous les documents sources de notre dataset. À l'issue de cette phase, nous construisons une vecteur associé à un document suspect qui emmagasine son état de plagiat contre tous les documents sources.

Après cela, il suffit d'appliquer le traitement ci-dessus pour tous les documents suspects. Le résultat est une matrice de prédiction $(n \times m)$ de l'état de plagiat pour tous les documents suspects.

Ensuite, nous préparons une matrice de l'état réel de plagiat à partir des fichiers d'annotation en format XML associés aux fichier suspects.

Depuis la matrice de prédiction et la matrice de l'état réel du plagiat nous construisons la matrice de confusion pour enfin calculer les mesures de précision, de rappel et le F1 score

Actuellement, nous somme dans l'attente de terminaison de notre programme de calcul de taux de plagiat pour achever cette phase de validation de notre travail.

3.5 Conclusion

Dans ce chapitre, nous avons proposée une méthode appliquée au sujet de la détection du plagiat dans les textes arabes, et pour la mettre en évidence, nous avons pris des mesures, notamment l'étape de prétraitement. Ensuite, nous avons appliqué réseau siamois, qui contient un modèle AraBERT.

Conclusion

Le plagiat des textes Arabes est un phénomène ennuyant dans notre ère de démocratisation de l'information. Le développement des système de détection du plagiat, notamment pour le texte Arabe, est devenu une urgence.

Après l'étude des travaux connexes, nous avons proposé un système de détection de plagiat du texte Arabe prenant en compte l'aspect sémantique (incorporation des mots, ordre, contexte) des documents sources et suspects. Notre système se base essentiellement sur l'utilisation de la nouvelle technologie des transformateurs. Dans notre cas, nous avons utilisé AraBERT.

Les résultats obtenus, malgré prématurés, sont encourageants pour finaliser notre système, notamment dans la phase évaluation qui nous serve comme feedback pour raffiner et ajuster les différents composants de notre système.

Nous conjecturons que cette opération de raffinement peut améliorer considérablement les résultats de notre système

Bibliographie

[Abbas and Hamdad, 2020] Abbas, J. and Hamdad, A. (2020). *Apprentissage automatique du dialecte algérien.* PhD thesis, Université Mouloud Mammeri.

[Abdelrahman et al., 2017] Abdelrahman, Y. A., Khalid, A., and Osman, I. M. (2017). A method for arabic documents plagiarism detection. *International Journal of Computer Science and Information Security*, 15(2) :79.

[Al-Thwaib et al., 2020] Al-Thwaib, E., Hammo, B. H., and Yagi, S. (2020). An academic arabic corpus for plagiarism detection : Design, construction and experimentation. *International Journal of Educational Technology in Higher Education*, 17(1) :1–26.

[Alzahrani et al., 2012] Alzahrani, S. M., Salim, N., and Abraham, A. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2) :133–149.

[Antoun et al., 2020] Antoun, W., Baly, F., and Hajj, H. (2020). Arabert : Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

[Balikas, 2017] Balikas, G. (2017). *Mining and learning from multilingual text collections using topic models and word embeddings.* PhD thesis, Grenoble 1 UGA-Université Grenoble Alpes.

[Belattar, 2021] Belattar, K. (2021). *Suivi de l'évolution des thèmes de publications scientifiques dans les communautés d'auteur· e· s et leurs co-citations.* PhD thesis, Université de Sherbrooke.

[Bengio et al., 2000] Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.

[Bensalem et al., 2015] Bensalem, I., Boukhalfa, I., Rosso, P., Abouenour, L., Darwish, K., and Chikhi, S. (2015). Overview of the araplagdet pan@ fire2015 shared task on arabic plagiarism detection. In *FIRE workshops*, pages 111–122.

[Bilgin and Senturk, 2017] Bilgin, M. and Senturk, I. (2017). Sentiment analysis on twitter data with semi-supervised doc2vec. pages 661–666.

[Chowdhury and Bhattacharyya, 2018] Chowdhury, H. A. and Bhattacharyya, D. K. (2018). Plagiarism : Taxonomy, tools and detection techniques. *arXiv preprint arXiv :1801.06323*.

[Eissen and Stein, 2006] Eissen, S. and Stein, B. (2006). Intrinsic plagiarism detection. volume 3936, pages 565–569.

[El Moatez Billah Nagoudi et al., 2018] El Moatez Billah Nagoudi, A. K., Cherroun, H., and Schwab, D. (2018). 2l-apd : A two-level plagiarism detection system for arabic documents. *Cybern. Inf. Technol*, 18(1) :124–138.

[Ferrero, 2017] Ferrero, J. (2017). *Similarités Textuelles Sémantiques Translingues : vers la Détection Automatique du Plagiat par Traduction*. PhD thesis.

[Fishman, 2009] Fishman, T. (2009). "we know it when we see it" is not good enough : toward a standard definition of plagiarism that transcends theft, fraud, and copyright.

[Gupta et al., 2016] Gupta, D. et al. (2016). Study on extrinsic text plagiarism detection techniques and tools. *Journal of Engineering Science & Technology Review*, 9(5).

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3) :146–162.

[Jansen, 2017] Jansen, S. (2017). Word and phrase translation with word2vec. *arXiv preprint arXiv :1705.03127*.

[Kusner et al., 2015] Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

[Li and Yang, 2018] Li, Y. and Yang, T. (2018). Word embedding for understanding natural language : a survey. In *Guide to big data applications*, pages 83–104. Springer.

[Mahmoud et al., 2018] Mahmoud, A., Zrigui, A., and Zrigui, M. (2018). A text semantic similarity approach for arabic paraphrase detection. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, pages 338–349, Cham. Springer International Publishing.

[Mahmoud and Zrigui, 2017] Mahmoud, A. and Zrigui, M. (2017). Semantic similarity analysis for paraphrase identification in arabic texts. In *Proceedings of the 31st Pacific Asia conference on language, information and computation*, pages 274–281.

[Mahmoud and Zrigui, 2019] Mahmoud, A. and Zrigui, M. (2019). Sentence embedding and convolutional neural network for semantic textual similarity detection in arabic language. *Arabian Journal for Science and Engineering*, 44(11) :9263–9274.

[Mahmoud and Zrigui, 2021] Mahmoud, A. and Zrigui, M. (2021). Hybrid attention-based approach for arabic paraphrase detection. *Applied Artificial Intelligence*, pages 1–16.

[Maurer et al., 2006] Maurer, H. A., Kappe, F., and Zaka, B. (2006). Plagiarism-a survey. *J. Univers. Comput. Sci.*, 12(8) :1050–1084.

[Meddeb et al., 2016] Meddeb, O., Maraoui, M., and Aljawarneh, S. (2016). Hybrid modeling of an offline arabic handwriting recognition system ahrs. pages 1–8.

[Menai, 2012] Menai, M. E. B. (2012). Detection of plagiarism in arabic documents. *International Journal of Information Technology and Computer Science*, 10(10) :80–89.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

[Nath, 2021] Nath, S. (2021). Style change detection using siamese neural networks. In *CLEF*.

[Niwattanakul et al., 2013] Niwattanakul, S., Singthongchai, J., Naenudorn, E., and Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Ranasinghe et al., 2019] Ranasinghe, T., Orăsan, C., and Mitkov, R. (2019). Semantic textual similarity with siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011.

[Ritchie et al., 1988] Ritchie, D. M., Kernighan, B. W., and Lesk, M. E. (1988). *The C programming language*. Prentice Hall Englewood Cliffs.

[Suleiman and Awajan, 2018] Suleiman, D. and Awajan, A. (2018). Comparative study of word embeddings models and their usage in arabic language applications. In *2018 International Arab Conference on Information Technology (ACIT)*, pages 1–7. IEEE.

[Vadivel et al., 2003] Vadivel, A., Majumdar, A., and Sural, S. (2003). Performance comparison of distance metrics in content-based image retrieval applications. In *International Conference on Information Technology (CIT), Bhubaneswar, India*, pages 159–164.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Zaher et al., 2018] Zaher, M., Shehab, A., Elhoseny, M., and Osman, L. (2018). A new model for detecting similarity in arabic documents. pages 488–499.

[Zrigui et al., 2016] Zrigui, S., Ayadi, R., Zouaghi, A., and Zrigui, S. (2016). Isao : An intelligent system of opinions analysis. *Res. Comput. Sci.*, 110 :21–30.