République algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Amar Télidji de Laghouat

Faculté des sciences

Département d'informatique

Thèse en vue de l'obtention du diplôme de doctorat en sciences

Spécialité : **Informatique**

Thème:

Détection automatique de plagiat

Présentée par: **El Moatez Billah NAGOUDI**

Composition du jury :

M. Ouinten Youcef	Prof.	Université Amar Télidji de Laghouat	**Président**
Mme. Cherroun Hadda	Prof.	Université Amar Télidji de Laghouat	**Rapporteur**
M. Khorsi Ahmed	MCA	Université Al-Imam Md. Ibn Saoud	**Rapporteur**
M. Ziadi Djelloul	Prof.	Université Normandie Rouen	**Examinateur**
M. Guessoum Ahmed	Prof.	Université des sciences et de la technologie Houari-Boumédiène	**Examinateur**

Résumé

Dans cette thèse, nous nous intéressons au problème de détection du plagiat. Le phénomène du plagiat s'aggrave de plus en plus avec l'intensification de l'utilisation des technologies de l'information et de la communication. Bien que le problème semble une simple recherche de similarité entre textes, chose que la machine est supposée avoir bien géré, il s'avère qu'il est plus difficile de faire face à certaines formes dissimulées de plagiat telles que la traduction, la substitution des mots par des synonymes, le changement de la structure des textes ou la reformulation des passages. Cela rend les systèmes classiques de détection du plagiat de moins en moins fiables.

En effet, le rôle d'un système de détection du plagiat est de faire face à toutes ces offuscations. Très peu de recherches dans ce contexte ont été appliquées pour la langue arabe, et même les travaux abordés sont incomparables à ce qui se fait pour les autres langues. De ce fait, nous nous intéressons dans cette thèse à la détection du copier-coller ainsi qu'à plusieurs autres formes de plagiat dissimulé principalement pour les documents en langue arabe.

Premièrement, nous avons proposé deux approches de détection de similarités sémantiques monolingue (arabe-arabe) et translingue (arabe-anglais) basées sur les plongements de mots (*word embeddings*) et la pondération fréquentielle et morphosyntaxique des vecteurs de mots. Puis, nous avons également proposé deux systèmes de détection du plagiat dans les documents en langue arabe. Le premier système est basé sur la technique d'empreinte digitale et les *word embeddings* afin de détecter les différentes formes dissimulées de plagiat. Notre deuxième système fait appel aux techniques d'apprentissage automatique supervisé.

Nos expérimentations sont menées sur le corpus de test de la première campagne d'évaluation de détection du plagiat dédiée à la langue arabe *AraPlagDet*. Toutes nos approches ont donné des résultats très satisfaisants par rapport à ceux proposés dans la littérature avec un *PlagDet* de 87% et un *rappel* dépassant les 92%.

Mots clés: plagiat textuel, détection du plagiat, plagiat dissimulé, traitement automatique de la langue, langue arabe.

Abstract

In this thesis, we are interested in the problem of plagiarism detection, which is continually worsening, due to the increasing use of modern communication technology. Fighting against such practices may seem to be a simple check of similarity between documents, something that machines are supposed to perform well. Indeed, plagiarism detection is a challenging natural language processing task.

The most recent achievements are systems able to detect the simple verbatim reproduction (copy-paste). On the other hand, plagiarism techniques are evolving to more complex forms such as synonyms substitution, text manipulation, paraphrasing, and even translation. This makes the plagiarism detection much more challenging than a simple comparison of pieces of text. While the issue is being heavily investigated in various languages, very few research works were conducted on Arabic. In this thesis, we address plagiarism detection in Arabic documents. Our research work extends to several forms of plagiarism, including copy-paste, synonym substitution, rewording, text manipulation and paraphrasing.

First, we have proposed two word-embedding-based approaches for measuring the semantic similarity between Arabic monologue sentences and Arabic-English cross-language sentences. The main idea is to exploit the word embeddings and vectors weighting to support the identification of words that are highly descriptive in each sentence. Then, we have proposed two systems devoted to assist users in detecting plagiarism in Arabic documents. The first one is based on fingerprinting, word embedding, words alignment, and words weighting. The second approach is based on machine learning for the purpose of detecting plagiarism in several disguised forms.

Our experiments are conducted using the dataset of the first Arabic Plagiarism Detection *AraPlagDet* shared task. The results show that all the proposed approaches achieve promising results compared to state-of-the-art Arabic plagiarism detection systems with a *Plagdet* of 87% and a *recall* of 92%.

Keywords : Textual Plagiarism, Plagiarism Detection, Disguised Plagiarism, Natural Language Processing, Arabic Language.

ملخص

في هذه الأطروحة ركزنا إهتمامنا على مشكلة الكشف الآلي عن السرقة الأدبية أو ما يسمى بالإنتحال، والذي أصبح يشهد إنتشارا واسعا في ظل النمو المتسارع لتكنولوجيا الإعلام والإتصال الحديثة. يعتبرالكشف الآلي عن الإنتحال من بين التحديات التي تواجه نظم المعالجة الآلية للغات الطبيعية، بالرغم من أن المشكلة تبدو على أنها عبارة عن بحث بسيط للتشابه بين النصوص، لكن في حقيقة الأمر لا يكتفي المنتحل بالطرق البسيطة كالنسخ و اللصق، بل يتعداها إلى ممارسة عدة طرق ذكية لتضليل آليات الكشف منها : الترجمة، تغيير ترتيب الكلمات، إستخدام المرادفات، أو إعادة صياغة الفقرات مما يصعب بشكل كبير مهمة الانظمة التقليدية للكشف الآلي عن الإنتحال. من جهة أخرى، يجب أن تلعب نظم كشف الإنتحال دورا أساسيا للتصدي إلى كل هذه الحيل.

في الواقع، لا نجد في هذا المجال إلا عدد القليل جدا من الأعمال الخاصة باللغة العربية، حيث لا يمكن مقارنتها بما تم إنجازه في لغات أخرى. الهدف الرئيسي من هذه الأطروحة هو معالجة مشكلة الكشف الآلي عن الإنتحال في الوثائق المكتوبة باللغة العربية، حيث ركزنا في بحثنا على إيجاد حلول ناجعة للكشف عن عدة أشكال من أساليب الإنتحال المضللة.

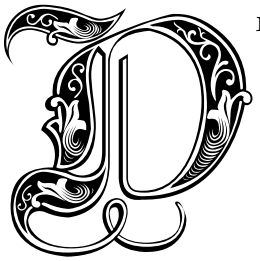
أولا، إقترحنا طريقتين لكشف التشابه الدلالي بين الجمل : الطريقة الأولى خاصة بالجملة المكتوبة بنفس اللغة (عربي ـ عربي)، أما في الثانية قمنا بكشف التشابه الدلالي بين الجمل في لغتين مختلفتين (عربي ـ أنجليزي). في كلتا الطريقتين إعتمدنا أساسا على تقنية تضمين الكلمات إضافة الى إسناد وزن لكل كلمة بالإعتماد على درجة تردد الكلمة و كذا أقسام الكلام. إضافة إلى ذلك، إقترحنا أيضا إثنين من أنظمة كشف الإنتحال في الوثائق المتكوبة باللغة العربية. النظام الأول يعتمد أساسا على تقنية البصمة الإ لكترونية، إسناد الأوزان الى الكلمات وكذا تضمين الكلمات. النظام الثاني يستخدم تقنيات التعلم الآلي من أجل إكتشاف مختلف طرق الإنتحال المضللة.

قمنا بإجراء تجاربنا على ذخيرة التجارب الخاصة باللغة العربية و المقترحة أثناء إجراء المسابقة الأولى (AabPlagDet) الخاصة بكشف الإنتحال في الوثائق المكتوبة باللغة العربية.

النتائج المتحصل عليها تثبت أن جميع الطرق المقترحة أعطت نتائج مرضية للغاية. بالإضافة إلى ذلك، تفوق أداء أنظمة الكشف الخاصة بنا على كل أنظمة كشف الإنتحال المقترحة سابقا، حيث بلغت نسبة الكشف 87% و نسبة الإرجاع 92%.

الكلمات المفتاحية : الإنتحال العلمي، السرقة الأدبية، الكشف الآلي للإنتحال، المعالجة الآلية للغات الطبيعية، اللغة العربية.

,



édicace À mes parents,

ma femme,

mes enfants,

mes frères et soeurs,

toute ma famille, ...

Remerciements

Je tiens tout d'abord à remercier *Dieu Le Tout Puissant et Miséricordieux,* qui m'a donné la force et la patience pour surmonter toutes les épreuves vécues au cours de cette thèse.

J'adresse de chaleureux remerciements à mes deux directeurs de thèse *Mme. Hadda Cherroun*, Professeur à l'université Amar Télidji de Laghouat et *M. Ahmed Khorsi*, Maître de conférences à l'université Al-Imam Muhammad Ibn Saoud (Ar. Saoudite), pour avoir eu confiance en moi et pour m'avoir donné l'opportunité d'effectuer cette thèse. Je les remercie pour leurs disponibilités, leurs critiques constructives, leurs patiences et leurs précieux conseils.

J'adresse évidemment un immense merci à *M. Didier Schwab*, Maître de conférences à l'université Grenoble Alpes (France), pour m'avoir accueilli dans son laboratoire pendant huit mois de mon stage résidentiel. Merci à Didier pour avoir également contribué à rendre tout ce travail possible, pour sa grande disponibilité et pour la qualité et la pertinence de ses questionnements et remarques.

Je voudrais remercier également *M. Ziadi Djelloul*, Professeur à l'université de Rouen Normandie (France), *M. Ahmed Guessoum*, Professeur à l'université des sciences et de la technologie Houari-Boumédiène (Alger) et *M. Ziani Benameur*, Maître de conférences à l'université Amar Télidji (Laghouat) pour l'honneur qu'ils m'ont fait en acceptant d'évaluer ce modeste travail. Je tiens à leur assurer de ma profonde reconnaissance pour l'intérêt qu'ils portent à ce travail. J'associe à ces remerciements *M. Youcef Ouinten*, Professeur à l'université Amar Télidji (Laghouat), pour avoir accepté de présider le jury et pour l'intérêt qu'il a porté à mon travail.

Je remercie tout naturellement toute ma famille, en particulier mes parents et ma femme, pour leur soutien durant cette thèse, pour leur patience et pour leur encouragements. Enfin, je tiens à remercier tous ceux et celles qui ont contribué de près ou de loin dans l'élaboration de ce travail.

Publications

Revues internationales

1. ***EMB Nagoudi***, A. Khorsi, H. Cherroun, and D. Schwab. "*2L-APD: A Two-Level Plagiarism Detection System for Arabic Documents*". Cybernetics and Information Technologies, 18(1), 124-138.

Conférences internationales

1. ***EMB Nagoudi***, H. Cherroun, and A. Alshehri. "*Disguised plagiarism detection in arabic text documents*". The 2^{nd} IEEE International Conference on Natural Language and Speech Processing, 2018, Alger, Algérie.

2. ***EMB Nagoudi***, J. Ferrero, D. Schwab, and H. Cherroun. "*Word embedding-based approaches for measuring semantic similarity of arabic-english sentences*". In the 6^{th} International Conference on Arabic Language Processing, pages 19–33. Springer, 2017, Fez, Morocco.

3. ***EMB Nagoudi***, J. Ferrero, and D. Schwab "*Enhancing the Semantic Similarity for Arabic Sentences with Vectors Weighting*", Proceedings of the 11^{th} International Workshop on Semantic Evaluations (SemEval-2017), Association for Computational Linguistics (ACL), August 3 - 4, 2017 Vancouver, Canada.

4. ***EMB Nagoudi***, J. Ferrero, and D. Schwab "*Amélioration de la similarité sémantique vectorielle par méthodes non-supervisées*", 24^{e} Conférence sur le Traitement. Automatique des Langues Naturelles (TALN), June 26-30, 2017, Orléans, France.

5. ***EMB Nagoudi*** and D. Schwab "*Semantic Similarity of Arabic Sentences with Word Embeddings*" in the Third Arabic Natural Language Processing Workshop (WANLP 2017) 3-7 April 2017, Association for Computational Linguistics (ACL), Valencia, Spain.

6. ***EMB Nagoudi***, A. Khorsi, H. Cherroun "*Efficient Inverted Index with n-Gram Sampling for String Matching in Arabic Documents*". 13^{th} ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2016), Nov 29^{th} to Dec 2^{nd}, 2016 Agadir, Morocco.

Table des matières

Liste des figures

Liste des Tableaux

Introduction générale

De nos jours et avec l'évolution rapide des nouvelles technologies de l'information et de la communication, une nouvelle forme de fraude a fait son apparition dans différents domaines et à plusieurs niveaux, notamment dans le milieu académique, scientifique et industriel. Il s'agit du Copier-Coller ou ce que l'on appelle le *plagiat*. Par exemple, dans le domaine académique, McCabe [McC05] a mené une étude sur plus de 80 000 étudiants dans les campus universitaires américains et canadiens entre 2002 et 2005. Cette étude révèle que plus de 25% des étudiants de premier cycle et 38% des étudiants de cycle supérieur ont déjà recopié ou paraphrasé des phrases provenant du Web sans citer la source. Une autre étude réalisée par Guibert et Michaut [GM11] montre aussi que plus de 34% de la proportion d'étudiants en Europe auraient déjà plagié tout ou partie d'un document pour le présenter comme un travail personnel.

Par ailleurs, plusieurs actes de plagiat ont été constatés dans le domaine scientifique. Par exemple, en novembre 2016, le magazine *L'Express*[1] a accusé le professeur *Étienne Klein* de plagiat dans son dernier livre "*Le pays qu'habitait Albert Einstein*" [Kle16]. Le professeur de physique de l'École centrale[2], directeur de recherche au Commissariat d'Énergie Atomique (CEA)[3] et président de l'Institut des Hautes Etudes pour la Science et la Technologie (IHEST)[4], s'est livré à des plagiats d'écrivains célèbres comme : *Gaston Bachelard, Émile Zola, Stefan Zweig* pour rédiger son nouveau livre consacré à *Einstein*. Dans le même contexte, une autre affaire de plagiat est celle de la ministre allemande de l'éducation *Annette Schavan* qui a remis sa démission après un scandale de plagiat dans son doctorat en philosophie [Gip14]. Au vu des exemples cités, la détection de plagiat devient alors un enjeu central, voire crucial, pour assurer la protection de la propriété intellectuelle.

[1] https://www.lexpress.fr/culture/livre/plagiat-les-copier-coller-du-physicien-etienne-klein_1855198.html
[2] https://www.centralesupelec.fr/ (consulté le 22/03/2018 à 12h)
[3] https://www.ihest.fr/ (consulté le 22/03/2018 à 12h)
[4] https://www.cea.fr/ (consulté le 22/03/2018 à 12h)

Dans le domaine informatique, la *détection de plagiat* est un hyperonyme qui désigne un système d'analyse textuelle informatisé conçu pour mesurer la similarité ou la quantité d'informations partagées entre deux documents [SKS07]. Ce système doit être capable de traiter des textes avec des styles différents, des langues différentes voire même des formats différents. En effet, la détection automatique de plagiat est devenue l'un des problèmes les plus importants dans le domaine de recherche d'informations [BC10].

Plusieurs systèmes de détection du plagiat ont été proposés dans la littérature [MKZ06, ASA12, BHZ12, OSA12, MFHDC14]. Les premiers systèmes sont basés sur la comparaison des caractères [Gip14]. Ces systèmes traditionnels utilisent des fonctions de similarités pour comparer littéralement le contenu d'un document plagié avec un ensemble de documents originaux. Cependant, ils ne peuvent pas identifier de manière fiable quelques formes dissimulées de plagiat telles que la substitution des mots par des synonymes, le changement de la structure des textes ou la reformulation des phrases. Cela rend ces systèmes classiques de détection du plagiat de moins en moins fiables. Il est donc nécessaire d'utiliser d'autres techniques anti-plagiats plus efficaces.

Objectifs

L'objectif principal de cette thèse vise à fournir un système de détection automatique du plagiat dans les documents en langue arabe[5]. Ce système doit être capable de détecter toutes formes de plagiat, à savoir le plagiat littéral comme le copier-coller, la restructuration des phrases, *etc.* Nous nous intéressons également à la détection de plusieurs formes de plagiat dissimulé y compris la substitution par des synonymes, la reformulation paraphrastique, *etc.* Afin d'atteindre ces objectifs, plusieurs questions doivent être traitées :

- Comment mesurer la similarité ou la quantité d'information partagée entre deux documents ?
- Comment détecter de façon efficace les formes dissimulées de plagiat telles que la substitution des mots par des synonymes, le changement de la structure des textes et la reformulation ?
- Comment lutter contre le plagiat dissimulé en faisant appel aux disciplines récentes du domaine de traitement automatique de la langue ?

[5] Dans le reste de ce manuscrit et dans un souci de simplicité nous utiliserons l'expression "*document arabe* " à la place de "*document en langue arabe*".

Bien qu'aucun paramètre de nos approches ne soit lié à une langue spécifique et que théoriquement la généralisation de nos approches à d'autres langues puissent paraître évidente, notre étude s'est focalisée sur la langue arabe pour les raisons suivantes:

1. très peu de recherches dans ce contexte ont été appliquées à la langue arabe, et même les travaux abordés sont incomparables à ce qui se fait sur les autres langues;
2. la langue arabe est caractérisée par une morphologie riche et complexe, et l'expérience a montré que l'adaptation des approches conçues pour les autres langues n'est pas évidente;
3. les travaux effectués pour la détection de plagiat dissimulé dans les documents arabes telles que la substitution par des synonymes et la reformulation sont encore très limités.

Principales contributions

Motivés par les raisons citées ci-dessus, nous nous sommes intéressés dans cette thèse à la détection de plusieurs formes de plagiat (copier/coller, la restructuration des phrases, la substitution par des synonymes, reformulation, *etc.*) dans les documents principalement ceux en langue arabe. Nos contributions portent sur la définition d'un cadre théorique et méthodologique pour la détection de plagiat. En effet, nous avons proposé une nouvelle technique de détection de similarités sémantiques monolingue (arabe-arabe) et translingue (arabe-anglais) basée sur les *word embeddings* et la pondération fréquentielle et morphosyntaxique des vecteurs de mots. Ainsi, nous avons conçu deux systèmes de détection de plagiat en langue arabe. Nous résumons nos principales contributions :

- **Sélection des n-grammes par fréquence**. Dans ce travail, nous avons exploité la distribution hétérogène des n-grammes dans un texte afin de réduire la taille de l'index inversé [BKC16]. En effet, nous avons illustré que les n-grammes les plus significatifs dans un document sont les n-grammes les moins fréquents. Par la suite, nous avons utilisé cette stratégie de sélection afin de construire un système de détection de plagiat [NKC18].

- **Mesurer la similarité sémantique entre les phrases en langue arabe.** Nous avons également proposé un système fondé sur les *word embeddings* [NS17]. Ce système est destiné à mesurer la similarité textuelle sémantique entre des phrases en langue arabe. L'idée principale est d'exploiter la représentation des mots par des vecteurs dans un espace multidimensionnel, afin de faciliter leur analyse sémantique et syntaxique. Des pondérations dépendant de la fréquence inverse en documents et de l'étiquetage morphosyntaxique sont appliquées sur les phrases examinées, afin d'améliorer l'identification des mots qui sont plus importants dans chaque phrase. Par ailleurs, nous avons participé avec cette technique à la sous-tâche monolingue arabe-arabe de détection de similarité textuelle sémantique (STS) de SemEval 2017 [CDA+17a]. Cette technique a prouvé son efficacité en arrivant *deuxième* parmi 49 systèmes proposés pour la compétition. Après cette compétition, nous avons proposé une pondération mixte (fréquentielle et morphosyntaxique) [NFS17b], cette dernière pondération serait arrivée en tête de la liste des systèmes participants à SemEval 2017.

- **Mesurer la similarité sémantique translingue entre les phrases arabes-anglais**. En se fondant sur notre méthode monolingue, nous avons proposé également une méthode pour mesurer la similarité translingue entre les phrases en arabe-anglais [NFSC17]. Cette méthode exploite, la traduction automatique, les *word embeddings*, l'alignement des mots et la pondération des vecteurs des mots.

- **Détection de plagiat à deux niveaux pour l'arabe**. Nous avons aussi proposé le système *2L-APD* (de l'anglais A Two-Level Plagiarism Detection System for Arabic) [NKC18]. Ce système est basé principalement sur deux modules (niveaux) : un module de détection utilise les empreintes digitales et un module de détection exploite la représentation *word embeddings*. Le module basé sur l'empreinte digitale est conçu pour détecter le plagiat littéral (*i.e.* niveau lexical), tel que copier/coller, le changement d'ordres des mots et l'ajout de mots vides. Cependant, dans les cas pratiques de plagiat, plusieurs autres formes dissimulées de plagiat sont utilisées, y compris la reformulation, la substitution par des synonymes ou la restructuration des phrases. Ces techniques d'obfuscations génèrent souvent un changement important dans la structure du texte original, ce qui peut affecter considérablement l'empreinte digitale du document. Ce

fait rend le module d'empreintes digitales très faible contre les modifications textuelles. Pour résoudre ce problème, nous avons proposé un deuxième module (niveau sémantique) basé sur les *word embeddings*, l'alignement des mots et la pondération fréquentielle et morphosyntaxique des mots, afin d'estimer la similarité sémantique entre documents.

- **Détection de plagiat arabe avec apprentissage automatique**. Notre deuxième système de détection du plagiat est basé sur l'apprentissage automatique supervisée (ML-APD) [NCA18]. Dans la phase d'apprentissage, nous caractérisons les couples de phrases (source, suspecte). En effet, nous identifions trois types de caractéristiques au défi de la détection de plagiat dans la langue arabe à savoir : lexical, syntaxique et sémantique. Les modèles sont générés à l'aide de l'un des trois types de classificateurs : *Machine à vecteurs de support* (*SVM*), les *arbres de décision* (*DT*) ou les *forêts d'arbres décisionnels* (*RF*). De plus, nous avons évalué l'efficacité de nos approches en utilisant le corpus proposé dans la campagne d'évaluation organisée pour la détection du plagiat en langue arabe *AraPlagDet* 2015.

Plan de la thèse

Cette thèse est répartie sur 4 chapitres. Dans le *chapitre* 1, nous introduisons dans un premier temps, la notion de plagiat textuel, les différents types de plagiat, ainsi que quelques terminologies liées au domaine de plagiat. En deuxième lieu, nous montrons que le plagiat est un problème d'actualité dans le milieu académique et de la recherche.

Afin de répondre aux objectifs visés, nous étudions le mode général de fonctionnement des systèmes de détection de plagiat dans le *chapitre* 2. Par la suite, nous détaillons les différentes approches utilisées pour concevoir un système de détection du plagiat à savoir les approches monolingues et translingues. Enfin, nous présenterons les protocoles et les métriques d'évaluation des systèmes de détection du plagiat.

Dans le *chapitre* 3, nous introduisons les caractéristiques de la langue arabe. Nous discutons aussi les défis associés au traitement automatique de l'arabe. Nous établirons dans la suite de ce chapitre un rapide état de l'art des systèmes de détection du plagiat dédiés à la langue arabe. Ce chapitre présentera ensuite la première campagne d'évaluation organisée pour la détection du plagiat en langue arabe *AraPlagDet* [BBR+15], ainsi que les corpus proposés.

Dans le *chapitre* 4, nous présentons nos approches de détection du plagiat en langue arabe. Nous décrivons dans un premier temps notre système de détection du plagiat à deux niveaux pour l'arabe (2L-APD) [NKC18]. Nous illustrons également notre deuxième système de détection du plagiat avec apprentissage automatique supervisé (ML-APD) [NCA18]. Ensuite, nous discutons le protocole et les corpus d'évaluation de nos systèmes. Finalement, Nous avons comparé nos systèmes de détection du plagiat avec celle des systèmes participants à la campagne d'évaluation *AraPlagDet 2015*.

" أخي لن تنال العلم إلّا بستّة ** سأنبيك عن تفصيلها ببيان
ذكاء وحرص واجتهاد وبلغة ** وصحبة أستاذ وطول زمان"

ديوان الإمام الشافعي بتعليق م. عفيف الزعبي (ص:٨١)

1

Généralités sur le plagiat

Dans ce chapitre, nous allons essayer de survoler le domaine de détection automatique de plagiat textuel dans le but de répondre à ces questions : Qu'est-ce que le plagiat ? Quels sont les différents types de plagiat ? En deuxième lieu, nous introduisons quelques terminologies et concepts de base dans le domaine du plagiat. Par la suite, nous présenterons quelques exemples récents de cas réels de plagiat en mettant en évidence le fait que le plagiat devient un problème d'actualité dans le milieu académique.

1.1 Définitions et terminologies

Dans cette section, nous définissons dans un premier temps la notion de plagiat en s'attardant plus particulièrement sur le plagiat textuel. En deuxième lieu, nous décrivons brièvement certaines terminologies essentielles dans le domaine de plagiat textuel. Nous présentons par la suite les différentes formes de plagiat textuel. Enfin, nous étudions la propagation de ce phénomène dans le domaine académique.

1.1.1 Plagiat

Le dictionnaire en ligne d'Oxford[1] définit le plagiat comme : « *la pratique de prendre les idées ou le travail d'un autre et de les faire passer comme étant les siennes* » (en anglais, *"The practice of taking someone else's work or ideas and passing them off as one's own"*). Toujours d'après le dictionnaire d'Oxford, l'origine du mot *plagiat* est issu du mot grec « *plagiarius* », ce mot désigne dans le monde grec le voleur d'esclave (*plagia* : voleur et *rius* : homme).

Selon Fishman [Fis09], le plagiat est défini comme suit : « *le plagiat est l'utilisation des idées, des concepts, des mots ou des structures et les intégrer à son propre travail sans en mentionner la provenance* » (en anglais, *"The use of ideas, concepts, words, or structures without appropriately acknowledging the source to benefit in a setting where originality is expected"*) .

Par ailleurs, le Trésor de la Langue Française Informatisée (TLFI) définit le plagiat comme : « *le plagiat est l'action d'emprunter à un ouvrage original, et par métonymie à son auteur, des éléments, des fragments dont on s'attribue abusivement la paternité en les reproduisant, avec plus ou moins de fidélité, dans une œuvre que l'on présente comme personnelle* » [2]. Il est important de noter que le mot ouvrage dans cette définition désigne tout travail original, c'est-à-dire qu'il est possible de conclure que l'utilisation d'un schéma, d'un programme informatique, d'une image ou d'une vidéo sans en citer le propriétaire (la source) est considéré comme un acte de plagiat.

Avant de parler sur des différentes formes de plagiat, il est important de définir quelques notions qui ont une relation directe avec ce thème. Dans cette section, nous définissons quelques terminologies qui ont une correspondance avec le domaine de plagiat textuel, telles que: les références bibliographiques, les citations, la propriété intellectuelle, les droits de l'auteur, *etc.*

1.1.2 Référence

Selon Baudry [Bau68], une référence est une « *action de se référer ou de renvoyer à un article, à un passage, à une chose ayant quelque rapport* ». Référencer une source dans un document représente un indicateur d'existence de source d'une information, texte,

[1] https://en.oxforddictionaries.com/definition/plagiarism (consulté le 05/12/2017 à 9h)
[2] http://atilf.atilf.fr/(consulté le 05/12/2017 à 10h)

parole ou autre, ayant un rapport direct ou indirect avec ce document. La référence doit contenir suffisamment d'information pour identifier de façon unique les sources telles que : un article, un livre, ou tout autre document. Ainsi, une référence doit contenir : un numéro (comme [12] ou bien [Chaouch, 2005]), un nom du/des auteurs, un titre, le nom de l'éditeur, et la date de parution , et le numéro international normalisé du livre (ISBN) (dans le cas d'un livre). Dans le cas d'un article : le nom du/des auteurs, le titre de l'article, le titre de la revue, le volume, le numéro de l'édition, la date et la page.

1.1.3 Citation

Millet [Mil97], définit une citation comme étant: « *un fait de parole (ou d'écriture), par définition individuelle et unique, qui est repris comme tel -cité- par un autre locuteur ou une infinité de locuteurs* ». Par ailleurs, l'action *cite* est définie dans le dictionnaire Larousse comme : « *rapporter les mots ou les phrases de quelqu'un ; paroles, passages empruntés à un auteur ou à quelqu'un qui fait autorité* »[3]. Donc, une citation est la reproduction d'un extrait d'un texte ou un écrit antérieur (ou même sous forme d'expression orale) dans la rédaction d'un nouveau texte.

Dans le domaine de la recherche, les deux termes citation et référence sont souvent utilisés de manière ambiguë [Laro4]. Egghe et Rousseau [ER90] définissent de manière formelle cette distinction comme suit : une référence dans un document A est une note bibliographique qui décrit un document B dans A. Si le document A contient une référence vers le document B, alors B reçoit une citation de A.

1.1.4 Bibliographie

La bibliographie est une liste structurée des sources citées (références), notamment des livres, des ouvrages, des articles de journaux, ou autres documents utilisés pour la préparation d'un document scientifique. La bibliographie se trouve généralement à la fin d'un article de journal, d'un livre ou d'un article d'encyclopédie [RTR06].

1.1.5 Droit d'auteur

Selon l'Organisation des Nations Unies : « *le droit d'auteur est un domaine du droit qui accorde aux auteurs (écrivains, musiciens, artistes et autres créateurs) une protection de*

[3] https://www.larousse.fr/dictionnaires/francais/citation/16228(consulté le 01/12/2017 à 08h)

leurs œuvres »[4] [TR85]. Ces œuvres peuvent être littéraires, artistiques, graphiques, phonogrammes, vidéogrammes, logiciels et programmes informatiques, *etc.* Ainsi, le droit d'auteur représente pour les auteurs une protection contre l'utilisation non autorisée de leurs œuvres.

1.1.6 Brevet

Kenneth Laudon [LL+11], définit un brevet par « *un document qui accorde à son propriétaire un monopole exclusif sur les idées qui sous-tendent son invention, pendant un nombre d'années qui varie selon les législations nationales* ». Le brevet permet à son détenteur (l'inventeur) d'avoir un titre de propriété et un droit exclusif pour l'invention, l'exploitation d'un procédé industriel, une nouvelle solution technique, une nouvelle manière de procéder ou autres. Le brevet offre une exploitation commerciale exclusive d'une invention sur une zone géographique précise et dans une période de temps limitée, généralement vingt ans [G+99].

1.2 Plagiat textuel

Le plagiat textuel est un plagiat impliquant la réutilisation d'un travail écrit (œuvre écrite) sans mentionner la source [GM11]. Le copier-coller ou la reproduction mot à mot des textes sans mentionner sa provenance est la forme la plus triviale du plagiat textuel. Cependant, l'acte de plagiat est aussi considéré comme présent lors de l'utilisation d'un processus de manipulation de texte comme par exemple la reformulation, le résumé, la traduction ou autre [C+03]. Donc, on peut dire que le plagiat textuel est un phénomène qui peut être produit sous plusieurs formes, telles que :

- Présenter comme sien un travail original de quelqu'un d'autre;
- Intégrer dans un travail des passages de textes, des données, des résultats expérimentaux ou même des images provenant de sources externes sans en mentionner la source;
- Résumer une idée créative de quelqu'un d'autre en l'exprimant avec d'autres mots tout en omettant d'en mentionner la provenance.

[4] http://www.un.org/fr/aboutun/copyright/index.html(consulté le 01/12/2017 à 13h)

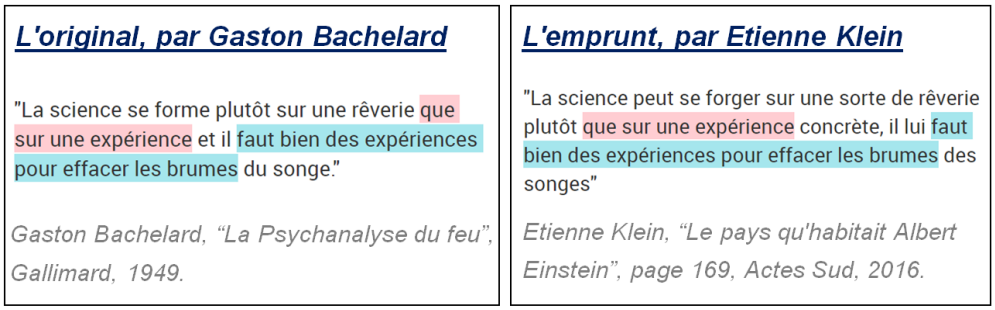


Figure 1.1: Comparaison entre les textes d'Étienne Klein « *Le pays qu'habitait Albert Einstein* » [Kle16] (page 169) et le livre « *La Psychanalyse du feu* » de Gallimard Gaston Bachelard [Gas49][5].

1.3 Formes de plagiat textuel

Nous présentons en détails les différentes formes de plagiat textuel dans la section suivante. En effet, en fonction de la méthode de plagiat, du support utilisé et des acteurs participants, on distingue plusieurs types de plagiat, à savoir la duplication, la paraphrase, la traduction, la collusion, le plagiat des idées, le ghostwriting, l'auto-plagiat, le plagiat avec sources inadmissibles et le plagiat inconscient. Dans ce qui suit, nous détaillons chacune de ces formes.

1.3.1 Duplication

La duplication ou le copier-coller, appelée aussi le cyber-plagiat est la forme de plagiat la plus utilisée [SSJ11]. Elle consiste à copier à l'identique tout ou une partie d'un texte mot à mot -sans aucun modification- dans un autre texte sans mentionner la source [BC10]. Par exemple, la figure 1.1 illustre un cas réel de duplication effectué récemment par le professeur *Étienne Klein* dans son dernier ouvrage « *Le pays qu'habitait Albert Einstein* » [Kle16]. Le docteur physicien a recopié des passages de texte sans guillemet et sans aucune mention[5] de l'auteur original *Gaston Bachelard* [Gas49].

1.3.2 Reformulation paraphrastique

La paraphrase ou la reformulation paraphrastique, consiste à reprendre un texte original sans altération de son contenu en utilisant le changement de son vocabulaire par l'ajout, la suppression ou la substitution de mots par des synonymes [WW10]. La

[5] https://www.lexpress.fr/culture/livre/etienne-klein-en-flagrant-delit-de-plagiat-la-preuve-par-quatre_1855211.html (consulté le 22/03/2018 à 11h)

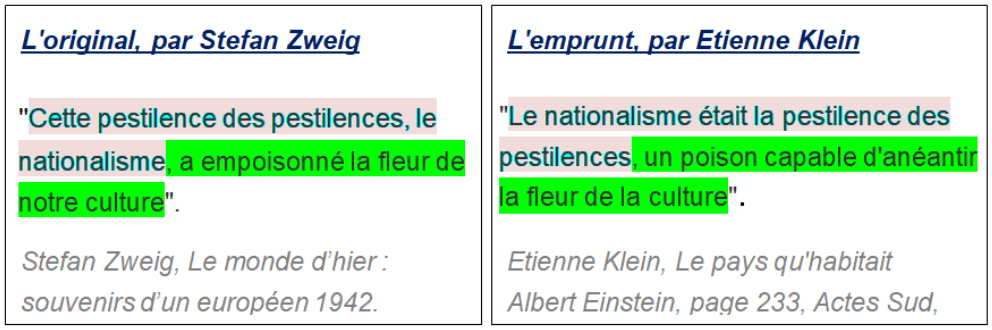


Figure 1.2: Comparaison entre les textes d'Étienne Klein « *Le pays qu'habitait Albert Einstein* » [Kle16](page 233) et le livre « *Le monde d'hier: souvenirs d'un européen* » [Ste42][6]

détection du paraphrase est une tâche particulièrement difficile et elle est considérée comme un problème ouvert [BC10]. Dans ce sens Burrows et al. [BPS13] soulignent que : « *la copie textuelle exacte est facile à détecter, alors que les cas de reformulation paraphrastique manuelles sont assez difficiles* ». La figure 1.2 présente un autre cas de plagiat avec paraphrase[6] apparue dans le même livre « *Le pays qu'habitait Albert Einstein* » d'*Étienne Klein* [Kle16]. Dans ce passage *Étienne Klein* a paraphrasé des passages du livre « *Le monde d'hier: souvenirs d'un Européen* » de l'auteur *Stefan Zweig* [Ste42].

1.3.3 Traduction

Le plagiat par traduction aussi appelé plagiat translingue consiste à faire une transformation manuelle ou automatisée d'un texte original d'une langue à une autre sans mentionner la source [WW10]. Par exemple traduire la définition de *Traitement Automatique du Langage Naturel (TALN)*[7] du français vers l'anglais comme l'illustre la figure 1.3. La détection du plagiat après traduction est considérée difficile surtout lorsque le plagiaire traduit de courts passages à partir de plusieurs sources de langues différentes [BC10]. Cependant, celui qui tenterait de faire passer un travail entier par traduction serait plus facilement détectable par les techniques de détection du plagiat translingue (*cf. section 2.4*).

[6] https://www.lemonde.fr/livres/article/2016/11/30/accuse-de-plagiat-le-physicien-etienne-klein-reconnait-avoir-pu-oublier-que-des-citations-n-etaient-pas-de-lui_5040869_3260.html (consulté le 06/12/2017)

[7] https://fr.wikipedia.org/wiki/Traitement_automatique_du_langage_naturel (consulté le 07/12/2017)

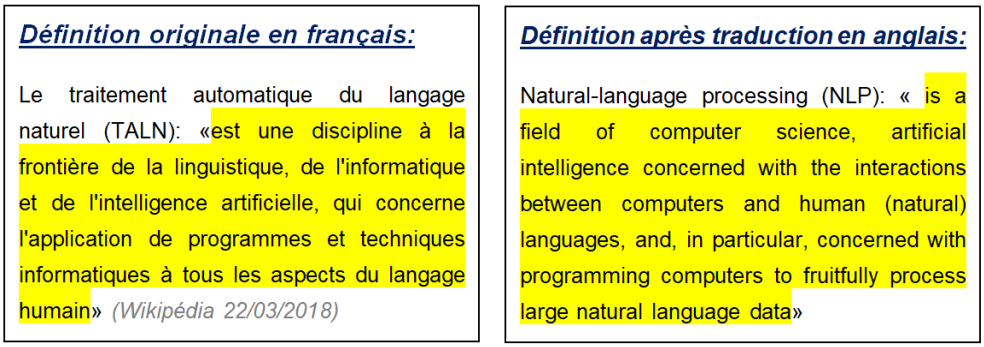


Figure 1.3: Exemple de cas de plagiat translingue[7]

1.3.4 Plagiat des idées

Selon Larivée [Lar95b], le plagiat implique non seulement les mots, mais aussi les idées. Le plagiat des idées est l'utilisation d'un concept ou une idée de quelqu'un d'autre sans mentionner la source. Et ceci, même si le texte contient les propres mots de l'auteur, mais la source de l'idée appartient à une autre personne.

1.3.5 Collusion

Le plagiat avec collusion se produit quand deux ou plusieurs étudiants coopèrent frauduleusement afin de réaliser un écrit académique (p. ex. un projet ou un devoir) qui devait en principe être individuel [Wil02]. Par ailleurs, Freegard [Fre06] ajoute que : « *la collusion est un comportement de tricherie qui peut être considéré comme une offense passible d'une exclusion de l'institution* ».

1.3.6 Fantôme littéraire

Le *ghostwriting* appelé aussi l'écrit de fantôme est une autre forme de fraude académique, souvent utilisée par les étudiants dans le domaine universitaire. L'étudiant paie une personne tierce ou une entreprise sur le net pour lui faire ses devoirs écrits, réaliser son mémoire de master ou même sa thèse de doctorat [FNBF17]. En 2016, Thomas Nemet le directeur général de l'entreprise *Acad Write*[8] spécialisée dans le conseil rédactionnel, déclare dans une interview avec le quotidien d'information *20 minutes*[9] à propos de ce phénomène : « *En 2015, nous avons écrit des travaux pour plus de 200 étudiants en Su-*

[8] https://www.acad-write.com/ (consulté le 07/12/2017 à 09h)

[9] http://www.20min.ch/ro/news/suisse/story/Uni–les–ghostwriters–ont-toujours-autant-la-cote-12297372 (consulté le 19/03/2018 à 8h)

isse ». Il précise par ailleurs que tous les travaux menés par des ghostwriters de Acad Write ne sont jamais détectés : « *dans la majorité des universités, les professeurs ne sont pas capables d'identifier le style d'écriture d'un étudiant* ».

1.3.7 Auto-plagiat

L'auto-plagiat, aussi appelé réplication, est la réutilisation par un auteur de ses propres travaux antérieurs sans y faire référence. La réplication consiste à soumettre une deuxième fois un travail (ou même une partie) déjà réalisé [BS08]. L'auto-plagiat est matière à débat car selon la définition de plagiat (*cf. section1.1.1*) « *le plagiat est la pratique de prendre les idées ou le travail d'un autre et de les faire passer comme étant les siennes* »; cependant, dans le cas de l'auto-plagiat il s'agit d'un travail personnel réutilisé. Dans ce sens, Resnik [R+05] souligne que la réplication relève de la malhonnêteté, mais pas un vol intellectuel.

Par exemple, la figure 1.4 illustre un cas d'un auto-plagiat effectué par le chercheur *Étienne Klein*[10]. Dans ce cas l'article intitulé « *L'instant présent, unique mais banal* » [Kle10] est publié initialement dans la revue « *Pour La Science* » par *Étienne Klein* en octobre 2010. Plusieurs parties de cet article ont été ensuite plagiées dans l'article « *Le temps est-il affaire de conscience ?* » publié dans la revue *European Psychiatry* en 2014 par le même auteur [Kle14].

1.3.8 Assemblage

Connu aussi sous le nom de "*patchwork*". Ce type de plagiat consiste à combiner deux sources ou plus pour créer un nouveau passage sans mention de sources. Selon Howard [How99] l'assemblage est considéré comme un vol intellectuel.

1.3.9 Plagiat inconscient

Le plagiat inconscient ou la cryptomnésie (souvenir caché) est un acte mémoriel par lequel une personne a le souvenir erroné d'avoir produit une pensée (une idée, un poème, un concept...), alors que cette pensée a été en réalité produite par une autre personne [Smi11].

[10] https://blogs.mediapart.fr/seraya-maouche/blog/210117/plagiats-en-serie-etienne-klein-recidive-dans-la-recherche (consulté le 21/03/2018 à 10h)

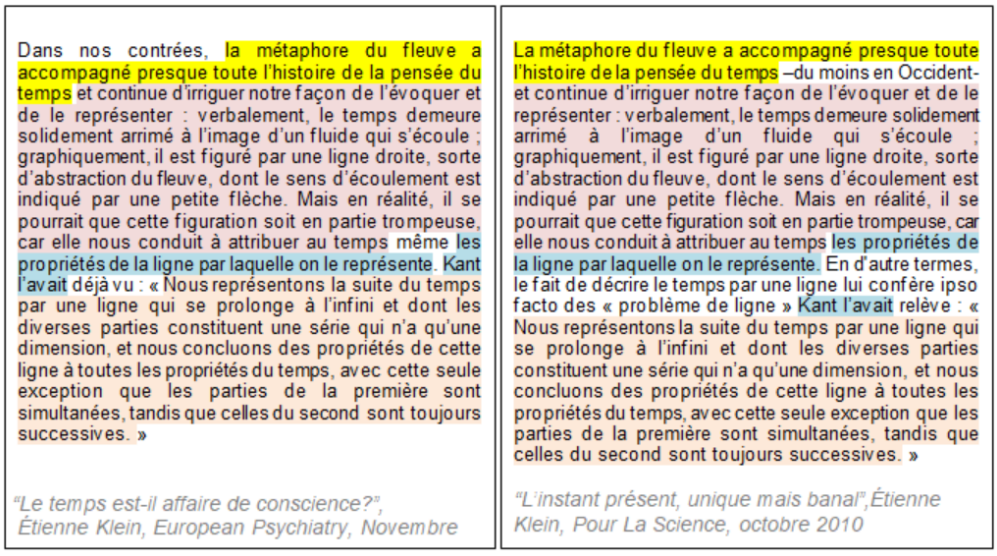


Figure 1.4: Un exemple d'un auto plagiat effectué par le docteur physicien *Étienne Klein*[10]

1.3.10 Source inadmissible

Parmi les fautes les plus fréquentes de rédaction est l'utilisation des références inadmissibles (même en citant la source). Par exemple, dans le domaine scientifique, l'usage d'un article tiré de Wikipédia est inacceptable[Nas06].

1.4 Le plagiat dans l'environnement académique

Le plagiat dans le domaine académique n'est pas un phénomène nouveau. Depuis un siècle, les chercheurs ont analysé ce phénomène en se concentrant sur des études empiriques. Ces études fournissent des preuves et des critiques sur la malhonnêteté académique en général [BE01], le comportement de tricherie collective [Whi98] et le plagiat en particulier [Par03].

Pour analyser et évaluer le phénomène de plagiat, la majorité des études traitent des enquêtes basées sur l'auto-évaluation (*self-report study*) [RAY97], [SN02], [KWL03], [McC05], [MBT06], [Gib06] et [GM11]. Par exemple, Guibert et Michaut [GM11] rapportent que 35% des étudiants en Europe ont déjà utilisé le copier-coller pour présenter tout ou une partie d'un autre travail comme un travail personnel. Une autre étude réalisée par Gibney [Gib06] sur les étudiants en France déclare que près de 46% des étudiants ont déjà utilisé une des formes de plagiat pendant leurs cursus

académiques. Ces études confirment les travaux déjà réalisés par McCabe [McC05] dans les campus américains et canadiens sur environ 80 000 étudiants universitaires, durant la période de 2002 à 2005. McCabe rapporte que plus de 38% des étudiants avaient copié ou paraphrasé au moins quelques phrases d'Internet sans indiquer la source. Bien que ces résultats paraissent déjà impressionnants, l'auteur suppose que les taux réels sont plus élevés, car les étudiants étaient très préoccupés par leur anonymat dans cette évaluation réalisée en ligne.

Il est important de noter que la plupart de ces études ne faisaient souvent pas la distinction entre les différentes formes de plagiat à savoir plagiat simple ou dissimulé (intelligent) et ne prennent pas en compte le degré de dissimulation du plagiat. Ce qui peut influencer les résultats de l'étude.

1.4.1 Le plagiat dans le domaine de la recherche

En ce qui concerne les études sur la malhonnêteté dans le domaine de la recherche, Swazey et al. [SALL93] présentent une enquête à grande échelle menée auprès de 2 000 doctorants et 4 000 de leurs encadrants. Selon cette étude, 28% des encadrants ont déjà trouvé du plagiat dans un travail réalisé par leurs doctorants. Par ailleurs, près de 7% des doctorants et 8% des encadrants ont déclaré avoir subi du plagiat durant leurs travaux de recherche. Par ailleurs, on constate que ces résultats pourraient être sous-évalués, car d'après Cheers et Dayton [SD87] les enquêtes d'auto-évaluation montrent une tendance à sous-estimer un comportement malhonnête.

Sorokina et al. [SGWG06] développent un système de détection du plagiat pour scanner environ 285 000 articles scientifiques dans l'archive de prépublications électroniques d'articles scientifiques *arXiv.org*[11]. Ils ont repéré plus de 500 documents qui contiennent des cas probables de plagiat et environ 11% de la proportion de documents étaient susceptibles de contenir un auto-plagiat excessif.

Dans le même contexte, on trouve également dans le domaine biomédical une étude très intéressante réalisée durant le projet Déjà Vu[12] [EHF+07, ESL+08, LEG+09, ESG+10, SEL+10]. Ce projet mesure les similitudes textuelles entre les résumés d'articles de biosciences dans MEDLINE[13] et leurs textes intégraux s'ils sont

[11] arXiv.org est une archive de prépublications électroniques d'articles scientifiques.
[12] http://dejavur.org/index.html (consulté le 12/12/2017 à 10h)
[13] https://www.nlm.nih.gov/pubs/factsheets/medline.html (consulté le 12/12/2017 à 11h

disponibles dans PubMed[14] Central (PMC). MEDLINE est un index qui contient des citations et des résumés d'articles de recherche biomédicale. Il contient plus de 24 millions d'articles[15]. PubMed Central (PMC) est une base de données bibliographique à accès libre d'ouvrages scientifiques dans les sciences de la vie et biomédicale. Par ailleurs, cette étude a identifié environ 80 mille articles avec des résumés très similaires. De plus, après une vérification manuelle de 4 515 textes intégraux d'articles, les auteurs ont identifié plus de 250 cas probables de plagiat et 89 cas d'auto-plagiat [EHF+07].

En novembre 2016, le magazine *L'Express*[16] a accusé le professeur *Étienne Klein* de plagiat dans son dernier livre « *Le pays qu'habitait Albert Einstein* » [Kle16]. Le professeur de physique de l'École centrale[17], directeur de recherche au Commissariat d'Énergie Atomique (CEA)[18], président de l'Institut des hautes études pour la science et la technologie (IHEST)[19], s'est livré à des plagiats d'écrivains célèbres comme : *Gaston Bachelard, Émile Zola, Stefan Zweig* pour rédiger son nouveau livre consacré à *Einstein*. Dans cette enquête, le journaliste de L'Express *Jérôme Dupuis*, qui a comparé les textes d'*Étienne Klein* aux travaux originaux, a montré quatre exemples de plagiat[20] (*voir les figures 1.1, 1.2 et 1.4 de la section 1.2*). Ainsi, il a prouvé que : « *Étienne Klein a aussi pratiqué les emprunts dans ses chroniques* »[21].

Après ces actes de plagiat, la ministre française de l'éducation nationale et de la Recherche *Najat Vallaud-Belkacem*, a décidé de mettre en place un comité scientifique pour fournir une expertise sur l'intégrité scientifique des travaux d'*Étienne Klein*. En se basant sur l'avis de ce comité, le docteur physicien a été démis de ses fonctions de président de l'*Institut des hautes études pour la science et la technologie (IHEST)* par un décret présidentiel[22] daté du 26 avril 2017.

[14] https://www.ncbi.nlm.nih.gov/pubmed (consulté le 12/12/2017 à 11h)
[15] Jusqu'à ce jour (consulté le 12/12/2017 à 11h)
[16] https://www.lexpress.fr/culture/livre/plagiat-les-copier-coller-du-physicien-etienne-klein_1855198.html
[17] https://www.centralesupelec.fr/ (consulté le 22/03/2018 à 12h)
[18] https://www.ihest.fr/ (consulté le 22/03/2018 à 12h)
[19] https://www.cea.fr/ (consulté le 22/03/2018 à 12h)
[20] http://www.lexpress.fr/culture/livre/etienne-klein-en-flagrant-delit-de-plagiat-la-preuve-par-quatre
[21] https://www.lexpress.fr/culture/livre/plagiat-les-copier-coller-du-physicien-etienne-klein_1855198.html
[22] https://www.lexpress.fr/actualite/societe/enquete/etienne-klein-nie-ses-plagiats-voici-sept-nouveaux-exemples_1857882.html (consulté le 24/03/2018 à 8h)

1.4.2 Le plagiat au sein des universités Algériennes

Ces dernières années, les universités algériennes ont été témoins de plusieurs cas de plagiat et de vols scientifiques, que ce soit par des étudiants ou même des enseignants chercheurs. Concernant les étudiants, par exemple en janvier 2016, le conseil scientifique de la faculté de droit à l'université de sciences politiques de l'Université Mohamed Ben Ahmed (Oran 2)[23] a sanctionné deux thèses de doctorat en plus d'un mémoire de magister[24]. Selon les rapports de la commission scientifique, ces travaux sont copiés à partir d'autres travaux originaux. Ainsi, le conseil de discipline de l'université a décidé de retirer ces travaux avec l'exclusion définitive des propriétaires.

Par ailleurs, les enseignants chercheurs sont aussi impliqués dans plusieurs actes de plagiat. Par exemples, lors de l'apparition du premier numéro de la revue scientifique internationale « *El Bourhane* » (en Arabe البرهان ISSN 2352-9857), spécialisée dans les sciences sociales et humaines et éditée en arabe, deux cas de plagiat intégral ont été identifiés[25].

Dans le premier cas, il s'agit d'un article reproduit mot à mot par le doyen de la faculté des sciences sociales et humaines, Université Abbès Laghrour de Khenchela (le professeur B. N.). Cet article intitulé : « *Diriger la recherche scientifique à l'université pour répondre aux exigences du développement économique et social* » (en Arabe : توجيه البحث العلمي في الجامعة لتلبية متطلبات التنمية الاقتصادية والاجتماعية) est intégralement copié-collé de l'article original de Prof. *Mahmoud Mohamed Abdullah Kasnawi* [MMA01], département d'éducation islamique et de comparaison - Faculté d'éducation - Université UMM Al-Qura, A.Saoudite [26].

Le deuxième cas s'agit d'un article publié dans le même numéro par l'enseignant chercheur R. S. de la même université. Cet article intitulé : « *Planification spatiale des services de santé dans le district de Jérusalem-Est par l'utilisation du système d'information géographique SIG* » (en Arabe : التخطيط المكاني للخدمات الصحية في منطقة ضواحي القدس الشرقية باستعمال تقنيات نظام المعلومات الجغرافية), est déjà réalisé en 2004, dans le cadre de l'obtention du diplôme du Magistère de planification urbaine et régionale par l'étudiant

[23] http://www.univ-oran2.dz/index.php/fr/ (consulté le 15/12/2017 à 14h)
[24] https://www.echoroukonline.com/ara/articles/271198.html (consulté le 15/12/2017 à 14h)
[25] https://www.djazairess.com/echorouk/236883 (consulté le 15/12/2017 à 14h)
[26] https://uqu.edu.sa/ (consulté le 17/12/2017 à 9h)

palestinien *Samer Hatem Rushdie* de l'université nationale An-Najah à Naplouse, Palestine [Esto9]. L'enseignante R. S. a pris le soin de modifier seulement « *Jérusalem-Est* » par « *Khenchela* » avant de soumettre l'article avec son nom.

Conformément aux dispositions de décret exécutif n° 08 — 130 du 3 mai 2008, l'article 24 de la loi algérienne relative à l'enseignement supérieur, le plagiat est considéré comme : « *faute professionnelle de quatrième degré, le fait pour les enseignants chercheurs, d'être auteurs ou complices de tout acte établi de plagiat, de falsification de résultats ou de fraude dans les travaux scientifiques revendiqués dans les thèses de doctorat ou dans le cadre de toutes autres publications scientifiques ou pédagogiques* »[27]. En se basant sur cette loi et après ces actes de plagiat, la présidence de l'université a décidé : le retrait définitif de la revue *El Bourhane*, de rétrograder le doyen de la faculté et l'enseignante en question, ainsi que l'exclusion pendant quatre ans de toute participation à des activités pédagogiques comme les supervisions des thèses et des mémoires de fin d'étude[28].

Dans un autre cas, l'université Ben Youssef Ben Khedda (Alger1)[29] a présenté des excuses officielles à l'ancien ministre marocain des Affaires étrangères, *Saad Eddine Othmani*, après avoir déposé une plainte auprès de la commission scientifique de l'université contre *Dr. B. B. A.* enseignant chercheur à l'université de Hassiba ben Bouali de Chlef[30], après un vol scientifique par ce dernier. Selon la commission scientifique de l'université, le chercheur *B. B. A.* a publié un article dans le journal de l'université d'Alger 1 qui contient plusieurs parties plagiées à partir du livre de Prof. *Saad Eddine Othmani* : « *Les actions du Prophète, que la paix soit sur lui, par l'imama* » [Otho1] (En Arabe : تصرفات الرسول صلى الله عليه وسلم بالإمامة) sans mentionner la référence originale.

1.5 Conclusion

Dans ce chapitre, nous avons introduit la notion de plagiat textuel, ainsi que certaines terminologies qui ont un rapport direct avec la tâche de détection du plagiat. Nous avons également présenté les différentes formes de plagiat textuel étudiées dans la littérature à savoir : duplication, assemblage, reformulation paraphrastique, traduc-

[27] https://www.joradp.dz/FTP/jo-arabe/2008/A2008023.pdf (consulté le 15/12/2017 à 15h)
[28] http://www.lematindz.net/news/17601-luniversite-de-khenchela-secouee-par-un-scandale-de-plagiat.html
[29] http://www.univ-alger.dz/ (consulté le 15/12/2017 à 15h)
[30] www.univ-chlef.dz/ (consulté le 15/12/2017 à 15h)

tion, collusion, ghostwriting et l'auto-plagiat. Par ailleurs, nous avons discuté dans ce chapitre des exemples récents et des cas réels de plagiat en mettant en évidence l'utilisation excessive du plagiat dans le milieu académique et de la recherche.

Dans le chapitre suivant, nous examinerons le problème de détection du plagiat textuel. Nous présenterons également les modèles de détection du plagiat textuel, à savoir le modèle de détection extrinsèque et intrinsèque. Nous illustrons aussi les différentes méthodes de détection dans chacun de ces modèles.

”روي عن الأخطل أنه قال: نحن معاشر الشعراء أسرق من الصاغة“

الموشح ، للمرزباني، صـ٢٢٥

2

Détection du plagiat

Dans ce chapitre, nous introduisons tout d'abord la notion de similarité textuelle. Nous exposons ensuite les deux modèles génériques de détection du plagiat à savoir les modèles extrinsèque et intrinsèque. Nous présentons également un bref état de l'art sur les techniques de détection du plagiat monolingue. Enfin, nous abordons le problème lié à la détection du plagiat translingue, ainsi que les différentes approches proposées pour traiter ce type de plagiat.

2.1 Similarité textuelle

L'analyse de similarité entre textes est une tâche qui trouve des applications dans divers domaines de recherches telles que la recherche d'information, la classification des textes, la traduction automatique, l'extraction de l'information ou la détection du plagiat. En effet, la similarité entre deux textes peut être évaluée en terme de distance, en utilisant un certain nombre de métriques à savoir : cosinus, distance euclidienne, distance Manhattan ou n'importe quelle autre fonction de similarité. Toutefois, il existe trois dimensions de similarité textuelle: lexicale, sémantique, structurelle.

2.1.1 Similarité lexicale

La similarité lexicale mesure le degré de similarité entre deux unités textuelles (phrases, paragraphes ou texte) par évaluation de taux de chevauchement du vocabulaire avec ordre. Par exemple, deux phrases sont lexicalement similaires si elles partagent les mêmes mots dans le même ordre [MRS+08]. Il existe plusieurs techniques de mesure de similarité lexicale, les plus connues sont : Damerau Levenshtein [Lev66], Needleman Wunsch [NW70], la plus longue sous-séquence commune (LCS) [CS75], Jaro Winkler [Win99], *etc.*

2.1.2 Similarité sémantique

La similarité sémantique consiste à mesurer le degré de relation sémantique entre deux unités textuelles, en se basant sur le sens des mots [ABC+15]. Par exemple, ils peuvent être synonymes, représenter la même chose ou ils sont utilisés dans le même contexte. Une manière classique de mesurer la similarité sémantique est d'utiliser des ressources linguistiques, comme WordNet [Mil95], HowNet [DD03], BabelNet [NP12a] ou Dbnary [Sér15]. Toutefois, le plongement de mots ("*word embeddings*" *cf. section 2.3.4*) représente une alternative plus efficace à ces bases lexicales [MKB+10]. Le principe de base de cette technique tient compte du contexte dans lequel les mots apparaissent, *i.e.* deux mots utilisés dans un même contexte ont potentiellement une similarité sémantique importante.

2.1.3 Similarité structurelle

La similarité structurelle est utilisée pour décrire les similitudes entre la composition de deux ou plusieurs documents. Les similarités structurelles dans le texte peuvent prendre différentes formes. Un exemple de similarité structurelle de document est le nombre de sections par chapitre, le nombre moyen de paragraphes par section, l'occurrence de citations partagées dans un ordre similaire dans deux documents, *etc* [Gip14].

2.2 Modèles de détection

Dans le domaine informatique, la *détection du plagiat* est un hyperonyme qui désigne un modèle d'analyse textuelle informatisée pour l'identification de la réutilisation des

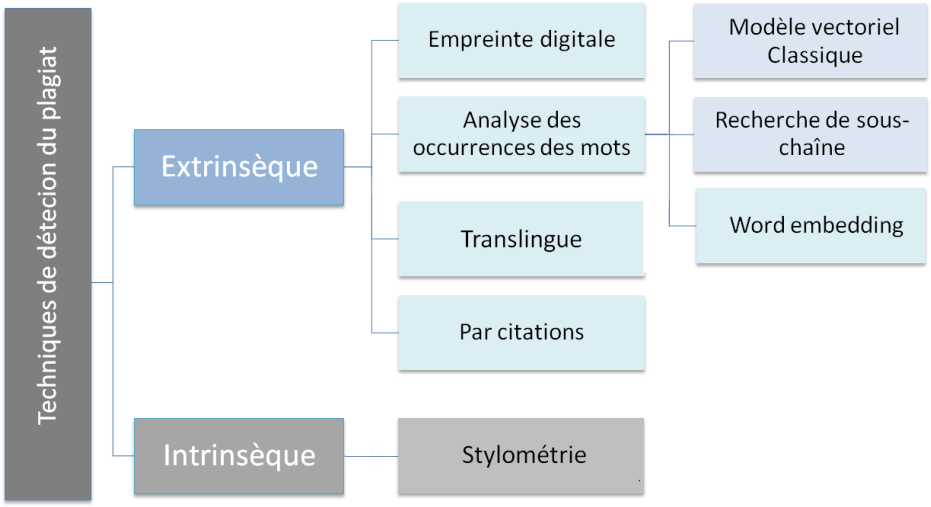


Figure 2.1: Différents techniques déployées dans la détection de plagiat textuel.

textes [SKS07]. Plusieurs *Systèmes de Détection Plagiat (SDP)* ont été proposés dans la littérature; ces systèmes appartiennent à l'une des deux familles : *extrinsèque* ou *intrinsèque* [BBR+15]. La figure 2.1 illustre les techniques de détection du plagiat les plus connues déployées dans le domaine de détection du plagiat textuel.

2.2.1 Détection extrinsèque

Les méthodes de détection du plagiat extrinsèques consistent à comparer un document suspect avec une collection de documents de référence authentiques (sources) pour trouver des passages similaires [SKS07]. Cette comparaison nécessite un ensemble de critères pour mesurer les similitudes, différences, écarts et distances afin de juger si le document contient du plagiat ou non. Ainsi, la révélation d'un nombre suffisant d'éléments communs est considérée comme une preuve d'existence d'un plagiat potentiel entre deux documents. Le principe de fonctionnement des méthodes extrinsèques s'effectue en trois étapes [Gip14]:

1. **Réduction de l'espace de comparaison.** Le but de cette étape est de sélectionner à partir de la collection de référence un sous-ensemble de documents (documents sources candidats) qui peuvent contenir des passages similaires par rapport au document suspect. Ainsi, les méthodes de détection du plagiat extrinsèques utilisent des modèles heuristiques pour réduire le nombre de documents

références concernés par la comparaison [Gip14]. Les modèles les plus utilisés dans cette étape sont les algorithmes de recherches de sous-chaîne [MRS+08], l'empreinte digitale [SGM95] et le modèle vectoriel [SB88].

2. **Comparaison.** Une comparaison approfondie est effectuée entre le document suspect et les documents candidats sélectionnés dans l'étape de réduction. Une comparaison efficace entre deux documents permet de repérer leurs points de similitude [FSL15]. Dans le cas d'une grande collection de référence, cette étape implique la comparaison d'un nombre très important de lignes de texte. Pour remédier à ce problème, les systèmes de détection proposent généralement un compromis entre l'effort de calcul et l'acceptation d'un certain degré de perte d'information [Gip14].

3. **Élimination des faux positifs.** L'étape de comparaison recherche les similitudes textuelles entre le document suspect et les documents candidats. Le résultat fourni par cette comparaison représente donc un taux de similarité, et non forcément un taux de plagiat. Par exemple, certains passages de texte contiennent un taux de similarité très élevés, mais ne sont pas du plagiat comme les passages entre guillemets avec mention de la source ou les expressions du langage courant. Afin d'éliminer au maximum le faux positif, les systèmes de détection font appel aux techniques relevant essentiellement des systèmes à bases de connaissances (*Expert System*) [Jac86], des langages dédiés (*Domain Specific Language*) [VDK+98] et des procédures de post-traitement [SKS07].

2.2.2 Détection intrinsèque

Les modèles intrinsèques utilisent une analyse statistique fondée sur les caractéristiques linguistiques d'un texte concerne, par exemple, les fréquences des mots, lettres ou n-grammes [ZES06]. L'étude statistique vise à caractériser les variations stylistiques de l'auteur. Ce processus est fréquemment utilisé pour solutionner le problème d'attribution d'auteur [VHBT+05]. Ceci est connu sous le nom de la *stylométrie*. Contrairement au modèle extrinsèque, les approches stylométriques ne font pas des comparaisons avec d'autres documents [OV13]. Elles utilisent les changements dans le style d'écriture comme indicateur pour détecter les documents plagiés. Ainsi, si un

passage de texte dans un document ne possède pas les mêmes caractéristiques stylométriques que le reste du document, on peut en conclure que ce passage n'appartient pas à la même personne [OV13, VH04].

2.3 Détection du plagiat monolingue

2.3.1 Détection du plagiat à base d'empreinte digitale

La technique d'*Empreinte Digitale* (ED) (en anglais *fingerprint*) également connue sous le nom de signature numérique, est la technique la plus utilisée par les systèmes de détection du plagiat extrinsèque [ZES06]. Cette méthode est basée essentiellement sur les travaux de Brinet et al. [BDGM95] et Shivakumar et al. [SGM95]. Elle représente un document par un vecteur d'entiers ou ce qu'on appelle une *empreinte digitale*. Ainsi, l'empreinte digitale d'un document représente une brève description de son contenu. Cette empreinte est utilisée par la suite pour la comparer avec celle d'autres documents de la collection de référence. Les fragments correspondants entre deux empreintes sont identifiés comme étant des passages identiques dans les deux documents [BDGM95].

2.3.1.1 Construction d'une empreinte digitale

Pour construire une empreinte digitale d'un document, la plupart des *SDP*s reposent sur un processus en trois étapes [HZ03]:

1. **Segmentation.** Dans un premier lieu, le document est segmenté en sous-séquences textuelles (segments), qui pourraient être une suite de caractères, mots, phrases, etc. [H+96].
2. **Sélection**. Un sous-ensemble de segments est sélectionné selon une stratégie bien définie, ces segments présentant des *points singuliers* (ou *minuties*) dans chaque document et constituant un motif unique appelé une *empreinte* [HZ03].
3. **Hachage**. Afin d'utiliser cette empreinte de manière efficace, une fonction de hachage est appliquée sur chaque minutie pour construire une *empreinte digitale* sous forme d'un vecteur d'entiers [HZ03, SWA03].

Par la suite, le *SDP* compare l'empreinte du document suspect avec un index pré-calculé d'empreintes digitales de tous les documents de la collection de référence. La figure

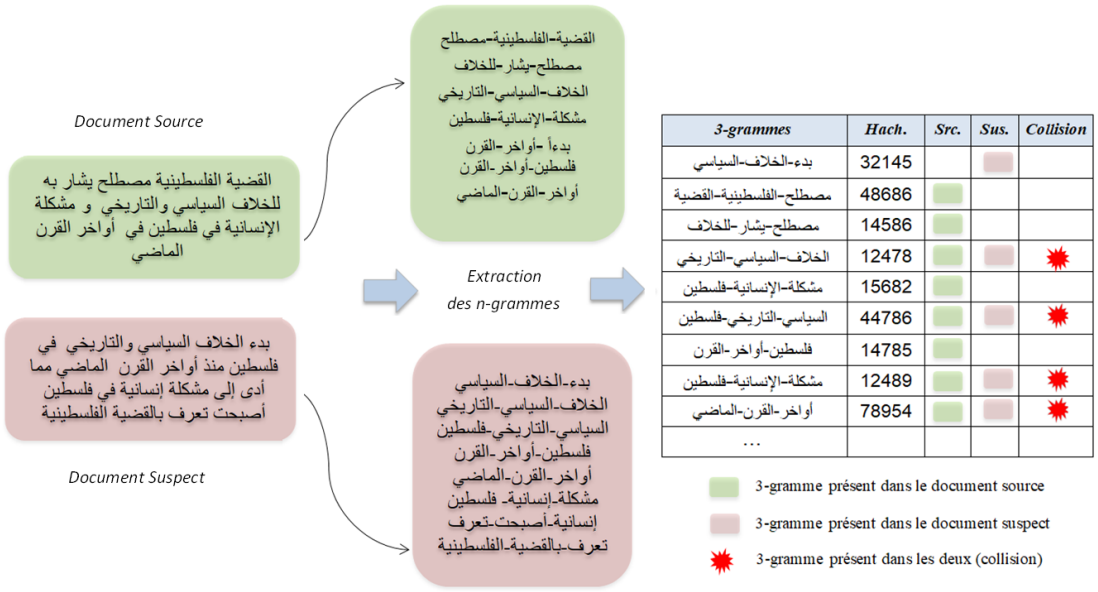


Figure 2.2: Détection du plagiat avec la méthode d'empreinte digitale

2.2 illustre les différentes étapes de construction d'une empreinte digitale dans le cas de détection du plagiat. Dans cette figure, nous avons choisi les mots comme unité de segmentation, les segments sont constitués par regroupement de trois unités (sans chevauchement), *i.e.* les documents d_1 et d_2 seront représentés par un ensemble de 3-*grammes* de mots.

En pratique, les *SPD*s basés sur la technique d'empreinte digitale partagent le même principe, la seule différence étant le choix des fonctions de segmentation, de sélection et de similarité. Nous présentons ensuite les principales méthodes de segmentation et de sélection appliquées sur le domaine de détection du plagiat.

2.3.1.2 Fonctions de segmentation

La construction d'une empreinte digitale nécessite une phase de découpage du document en sous-séquences textuelles (segments). Toutefois, il faut bien noter que deux propriétés doivent être prises en compte lors du découpage du texte [BDGM95, SGM95, H+96]. La première représente l'unité indivisible du texte appelée par Heintze [H+96] *unité de segmentation* (*unit of chunking*) et *unité primitive* (*primitive unit*) par Shivakumar et Garcia-Molina [SGM95]. Plusieurs tailles d'unités de segmen-

tation ont été proposées. Par exemple, Barrón-Cedeño et Rosso [BCR09] considèrent les n-grammes de caractères comme unité de segmentation, tandis que Schleimer et Garcia-Molina [SGM95] et Lyon et al. [LBM04] utilisent les n-grammes de mots. Le tableau 2.1 résume les unités de segmentation proposées dans la littérature dans le cadre de détection du plagiat avec l'empreinte digitale [Gip14].

La seconde propriété est la taille de la sous-séquence textuelle (*taille du segment*). Elle correspond à un regroupement séquentiel d'instances d'unités de segmentation. Brin et al. [BDGM95] et Shivakumar et Garcia-Molina [SGM95] nomment cette unité *chunk*. Broder et al. [BGMZ97] y font référence en utilisant le terme *w-tuile* que l'on pourrait traduire par une séquence d'unités de segmentation (*tuile*) de taille w. Hoad et Zobel [HZ03] utilise le mot *minutia* pour nommer la version hachée de ce segment. Nous choisissons d'appeler cette unité *taille du segment*. Par exemple, si nous choisissons le *mot* comme *unité de segmentation* et 5 comme *taille de segment*, on obtient des *segments* avec une taille égale à 5 mots.

Unité de segmentation	**Utilisé par**
k caractères ($k \in 3, 4$)	[BCR09], [NKC18]
Un seul mot	[SGM95]
k mots ($k \in 3 - 5$)	[HZ03],[KB10], [LMD01], [LC11], [JE12], [Men12], [Alz15]
k mots ($k \in 7 - 10$)	[BGMZ97, SGWG06]
k mot vides ($k \in 8 - 11$)	[Sta11]

Table 2.1: Unités de segmentation proposées pour la détection du plagiat avec ED [Gip14].

2.3.1.3 Fonctions de sélection des unités

La sélection des unités de segmentation est une étape fondamentale pour construire un système de détection de plagiat efficace. La fonction de sélection permet de définir la façon dont les unités de segmentation (u) ainsi construites, vont être assemblées afin de constituer l'empreinte du document. La sélection de toutes les unités génère une *Empreinte Digitale Complète (EDC)* (full fingerprint). L'empreinte complète représente une mise en oeuvre simple et efficace car elle permet une couverture complète du document. De plus, elle a fait ses preuves dans le domaine de détection

du plagiat et la réutilisation de texte, sous condition de bien choisir la taille de segment, voir les travaux de [LBM04], [C+03], [BGMZ97], [YC05], [SC08], [BCR09], et [Men12]. Cependant, la sélection de toutes les unités de segmentation implique un coût élevé du point de vue effort de calcul et espace de mémoire nécessaire. Pour pallier ce problème, un certain nombre de stratégies de sélection ont été proposées, dont les plus connus sont: *k-premières unités* [HZ03], *k-premières unités avec décalage* [HZ03], k^{eme} *unité* [HZ03], *o mod k* [BDGM95], et *Winnowing* [SWA03] et *la sélection par fréquence* [BKC16]. Dans ce qui suit, nous présentons ces fonctions de sélection.

- *La fonction k-premières unités* :

  La fonction k-premières unités sélectionne les k premières unités (sans chevauchement). Par exemple, si nous utilisons les 4-*grammes* comme unité de segmentation, les k-premières 4-*grammes* qui commencent aux positions $0, 4, 8, 12, .., 4k$ sont sélectionnés [HZ03, SZE06]. Donc, elle fournit une empreinte digitale de taille fixe k quelle que soit la taille du document. Cette fonction est présentée par l'algorithme 1.

- *La fonction k-premières unités avec décalage* :

  Cette fonction partage le même principe de k-premières unités (voir l'algorithme 2); la seule différence est qu'elle considère le chevauchement [HZ03]. Par exemple, si on continue avec le même exemple précédent, les positions des 4-*grammes* sélectionnés sont : $0, 1, 2, 3, \ldots, k$.

- *La fonction k-ème unité*:

  Karp et Rabin [KR87] font le choix de sélectionner chaque k-ème unité comme décrit par l'algorithme 3. Ainsi, la taille de l'empreinte proposée par cette fonction est proportionnelle à la taille du texte $(taille(texte)\ div\ k)$.

Algorithm 1: k-premières unités

Input : EDC: Empreinte Digitale Complète avec $EDC=\{u_0, u_1, \ldots, u_m\}$.
n: taille du n-gramme, k : le paramètre le la fonction
Output: Empreinte Digitale $ED=\{u_0, u_n, u_{2n}, \ldots, u_{kn}\}$
for $(i = 0; i < k; i=i+n)$ **do**
  Ajouter u_i à ED.
Retourne(ED)

Algorithm 2: k-premières unités avec décalage

Input : EDC: Empreinte Digitale Complète avec $EDC=\{u_0, u_1, \ldots, u_m\}$.
n: taille du n-gramme, k : le paramètre le la fonction
Output: Empreinte Digitale $ED=\{u_0, u_1, u_2, \ldots, u_k\}$
for $(i = 0; i < k; ++i)$ **do**
  Ajouter u_i à ED.
Retourne(ED)

Algorithm 3: k-ème unité

Input : EDC: Empreinte Digitale Complète avec $EDC=\{hach_1, hach_2, \ldots, hach_n\}$.
k : le paramètre le la fonction
Output: Empreinte Digitale $ED=\{hach'_1, hach'_2, \ldots, hach'_m\}$
foreach $hach_i \in EDC$ **do**
  if $i \bmod k = 0$ **then**
    Ajouter $hach_i$ à ED.
  Retourne(ED)

Il est à noter que les fonctions *k-premières, k-premières avec décalage* et k^{eme} *unité* sont rapides et simples à implémenter. Cependant, elles sont très sensibles en cas de suppression, d'insertion ou de reformulation [E+05]. Par exemple, l'insertion d'un seul caractère (voire k caractères) déplace les positions de toutes les lettres dans le document par une position (k positions respectivement), ce qui implique une modification au niveau de l'empreinte digitale du document [Men12]. Pour contourner ce problème, d'autres méthodes de sélection ont été mises au point, à savoir: la fonction *o mod k*, le *winnowing* et *la sélection par fréquence*.

- *La fonction o mod k*:

  Manber [M+94] propose de garder toutes les unités de segmentation qui vérifient la condition ($hach$(unité) $mod\ k = 0$) pour un certain k fixe. L'avantage de cette fonction par rapport à celles de k-premières (avec ou sans décalage) et k^{eme} *unité* est le fait que les unités sont choisies indépendamment de leur position dans le document. Donc, si deux documents partagent la même unité u avec $hach(u)\ mod\ k = 0$, alors u sera sélectionné dans les deux documents. Cette fonction est décrite dans l'algorithme 4.

Algorithm 4: o mod k

Input : *EDC*: Empreinte Digitale Complète avec $EDC=\{hach_1, hach_2, \ldots, hach_n\}$.
k : le paramètre le la fonction
Output: Empreinte Digitale $ED=\{hach'_1, hach'_2, \ldots, hach'_m\}$
foreach $hach_i \in EDC$ **do**
  if $hach_i \ mod \ k = 0$ **then**
    Ajouter $hach_i$ *to ED.*
  Retourne(ED)

- *Le Winnowing*:

  Développé par Schleimer et al. [SWA03], l'algorithme winnowing représente la fonction la plus populaire pour sélectionner les unités d'une empreinte digitale. Basé sur les n-grammes de mots ou de caractères comme unité de segmentation, il propose un compromis entre le nombre des unités à sélectionner et la plus courte correspondance que nous sommes sûrs de détecter. À cette fin, il utilise deux seuils choisis par l'utilisateur, un seuil de garantie G et un seuil de bruit B. Pour générer l'empreinte, l'algorithme winnowing sélectionne dans chaque fenêtre de taille G la valeur minimum la plus droite $hach(B\text{-}gramme_i)$. Afin de détecter la similarité entre deux documents, l'algorithme winnowing assure deux propriétés très importantes : (1) S'il y a une correspondance entre deux sous-chaînes au moins aussi longues que le seuil de garantie G, alors cette correspondance est détectée. (2) Il ne détecte aucune correspondance plus courte que le seuil de bruit B. L'algorithme5 illustre cette procédure de sélection.

- *La sélection par fréquence*:

  La sélection par fréquence consiste à juger la pertinence d'une unité u selon sa distribution dans une collection ou un document. Plusieurs stratégies de sélection à base de fréquences ont été proposées; la plupart de ces stratégies suppriment les unités les moins ou les plus fréquentes. Par exemple, Kocz et Chowdhury [KCA04] et [BZ04] sélectionnent uniquement les unités qui sont présentes dans plusieurs documents dans la collection. Tandis que, Chowdhury et al. [CFGM02] utilisent la fréquence inverse de document pour supprimer les unités rares et les unités fréquentes. Heintze [H+96] propose de sélectionner les unités d'un segment à condition que les cinq premières lettres de l'unité sont peu fréquentes dans le doc-

ument. Récemment, Nagoudi al. [BKC16] exploitent la distribution hétérogène des n-grammes dans un texte afin de sélectionner les unités d'une empreinte digitale. Le pseudo-code de notre approche de sélection par fréquence est illustré par l'algorithme 6.

Algorithm 5: Winnowing [SWA03]

```
Input  : EDC={hach_1, hach_2, ..., hach_n,}   /*Empreinte Digitale Complète */
         w: taille de la fenêtre, B: seuil de bruit, G: seuil de garantie
Output: Empreinte Digitale ED={(hach_1, pos_1), (hach_2, pos_2), ..., (hach_m', pos_m')}.
int  h[w]                                   /*  une fenêtre de taille w */
for ( i = 0 ; i < w ;+ + i) do
    h[i] = int_max
int r, min =0                               /* Indice de la fenêtre et de hash minimum */
while (true) do
    r = (r + 1) % w                         /* Déplacer la fenêtre par un */
    h[r] = hash_suivant()                   /* Ajouter un nouveau hachage*/
    if min == r then
        for ( i = (r − 1)%w; i ! = r; i = (i − 1 + w)%w) do
            if h[i] < h[min]) then
                min = i
            Record(h[min], global_pos(min, r, w))
    else
        if h[r] <= h[min] then
            min = r
            Record(h[min], global_pos(min, r, w));
```

Algorithm 6: Sélection par fréquence [BKC16]

```
Input  : EDC: Empreinte Digitale Complète EDC={hach_1, hach_2, ..., hach_n}.
         frq: fréquence seuil
Output: Empreinte Digitale EDC={hach'_1, hach'_2, ..., hach'_m}
foreach hach_j ∈EDC do
    if Freq(hachj) < frq then
        Ajouter hach_j à ED.
    Retourne(ED)
```

2.3.1.4 Propriétés d'une empreinte digitale

Dans le cas de détection du plagiat, il y a quatre facteurs qui doivent être soigneusement choisis par un *SDP* afin d'assurer l'efficacité d'une empreinte digitale [HZ03]:

1. **Granularité d'une empreinte digitale.** Cette propriété a un impact significatif

sur la précision de détection du plagiat. La granularité grossière (*i.e.* diviser le document en segments de grande taille) permet d'obtenir une empreinte rapide en terme de vitesse de calcul, mais très sensible aux changements. La granularité fine (segments de petite taille) est moins sensible à de tels changements, cependant elle nécessite un effort de calcul important et provoque un taux plus élevé de faux positifs.

2. **Stratégie de sélection.** Bien que la sélection de toutes les unités de segmentation est susceptible d'être le meilleur choix pour une mise en correspondance stricte entre deux empreintes, la sélection d'un sous ensemble d'unités (les plus significatives) selon une stratégie bien définie est plus efficace et moins sensible aux changements insignifiants.

3. **Résolution de l'empreinte.** Le nombre d'unités sélectionnées pour représenter un document définit la résolution d'empreinte digitale. L'effort de traitement et l'espace de stockage augmentent proportionnellement à la résolution des empreintes digitales.

4. **Fonction de hachage.** La fonction de hachage mappe les unités sélectionnées à des nombres positifs. Il est particulièrement important de choisir la fonction de hachage de manière à minimiser les collisions dues au mappage de différentes unités vers le même hachage.

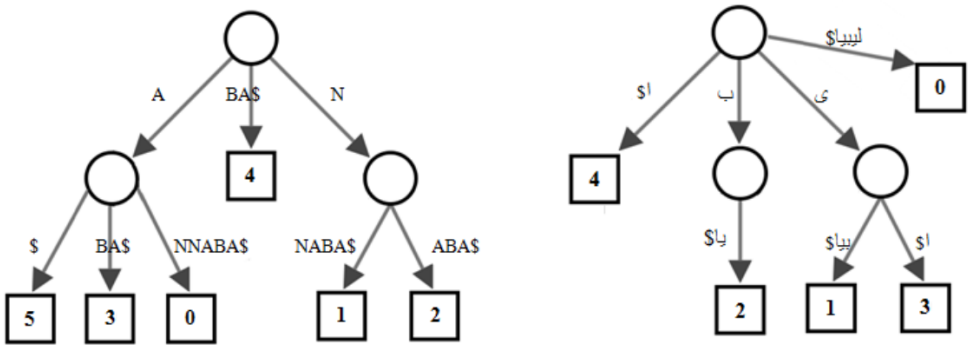
2.3.2 Détection du plagiat par recherche de sous-chaîne

Ce type d'algorithme a pour objectif de trouver une sous-chaîne de caractères (*motif*) à l'intérieur d'une autre chaîne (*texte*). Plusieurs modèles ont été proposés pour réaliser cette tâche; les plus connus sont les modèles des suffixes [MRS+08]. Un modèle de suffixe est une structure de données contenant tous les suffixes d'une chaîne de caractère et elle permet des comparaisons efficaces. Les structures les plus utilisées sont les arbres des suffixes [Wei73] et les tableaux des suffixes [MM93]. Nous rappelons tout d'abord le principe général de fonctionnement de chacune de ces structures ainsi que leurs mises en œuvre pour la détection du plagiat extrinsèque.

2.3.2.1 Les arbres des suffixes

Introduit par Weiner et Peter en 1973 [Wei73], l'arbre des suffixes (suffix tree) est une structure de données arborescente contenant tous les suffixes d'une chaîne de caractères. Il est couramment utilisé par les systèmes de recherche d'information [MRS+08]. L'arbre de suffixe représente le texte (T) sous la forme d'un graphe d'unités textuelles (suites de lettres) comme suit:

1. Les feuilles de l'arbre contiennent le numéro de la position du début du suffixe dans le texte.
2. Les branches peuvent être étiquetées de différentes manières:
   - Par une chaîne de caractères de longueur supérieure ou égale à 1,
   - Par un couple (*Pos*, *Long*) qui correspond à la sous chaîne commençant à la position *Pos*, de longueur *Long*, dans le texte.
3. Toutes les étiquettes des branches sortant d'un même nœud commencent par une lettre différente.
4. Dans l'arbre des suffixes, chaque chemin de la racine vers un nœud (feuille) est un suffixe de la chaîne. La figure 2.3 illustre deux exemples d'arbres des suffixes des chaînes "ANNABA$" et "$ليبيا"



(a) "ANNABA$" (b) '$ليبيا '

Figure 2.3: Exemples d'arbres des suffixes.

La recherche d'un motif $M = m_1..m_k$ dans l'arbre des suffixes du texte $T = t_1..t_n$ est très simple. En partant de la racine de l'arbre, il faut d'abord suivre la branche dont

l'étiquette commence comme m_1. En suivant cette branche on arrive alors à un nouveau nœud et on recommence le même processus avec la sous chaîne $(m2..m_k)$ [Wei73], ce processus est répété jusqu'à:

1. On ne peut plus descendre dans l'arbre par la lettre lue m_i, ce qui signifie qu'il n'y a pas d'occurrence de M dans le texte T,
2. Lire M en entier, dans ce cas soit on arrive sur: (a) Un nœud interne, dans ce cas le nombre de feuilles présentes dans le sous-arbre correspond au nombre d'occurrences de M dans T. (b) Sur une feuille, ce qui signifie qu'il y a une seule occurrence de M dans T, et elle se trouve à la position indiquée par le numéro de la feuille.

Il existe plusieurs algorithmes pour la construction de l'arbre des suffixes; les plus importants sont la méthode de Weiner [Wei73], McCreight [McC76] et de Ukkonen [Ukk95]. La méthode Weiner a été la première à montrer que les arbres des suffixes, peuvent être construits en temps linéaire. Quant à la méthode Ukkonen, elle est tout aussi rapide, mais elle utilise beaucoup moins d'espace mémoire [Gus97].

Par ailleurs, l'arbre des suffixes peut être construit en un temps et espace linéaires : pour un texte de longueur n, la construction est effectuée en temps $O(n)$ [Gono0]. La recherche de toutes les occurrences d'un motif M de k caractères nécessite un temps d'ordre $O(k + c)$, où c est le nombre d'occurrences de M dans l'arbre [Gono0].

2.3.2.2 Les tableaux des suffixes

Le tableau des suffixes (en anglais Suffix Array), est une structure de données qui a été introduite par Manber et Myers [MM93] et indépendamment par Gonnet et al. sous le nom de *Tableau de PAT* [GBYS92]. Le tableau des suffixes contient tous les suffixes triés par ordre alphabétique ainsi que leurs positions dans le texte.
La construction de la table des suffixes d'un texte T de longueur n, passe par les étapes suivantes :

1. La récupération de tous les n suffixes du texte T.
2. Ordonner les suffixes d'une manière croissante selon l'ordre lexicographique.

Suffixe	$	a$	aba$	annaba$	ba$	naba$	nnaba$
position	7	6	4	1	5	3	3

Figure 2.4: Le tableau des suffixes de la chaîne *annaba*$.

3. Chaque suffixe a une position de début dans T; par exemple, le suffixe à la position 0 est T lui-même. La position 2 indique le début de la sous-chaîne $T[2]..T[n-1]$, etc.
4. Une fois les suffixes sont ordonnés, leurs positions de début forment le tableau des suffixes; *i.e.* le tableau des suffixes de T contient seulement les positions des débuts (triés) de ces suffixes. Par exemple, la figure 2.4 illustre le tableau des suffixes de la chaîne *annaba*$.

Pour rechercher un motif M dans le tableau des suffixes, il suffit de trouver ce motif comme un préfixe de l'un des suffixes de la chaîne. Comme le tableau est trié par ordre alphabétique, donc on aura les suffixes commençant par le motif recherché dans la même partie dans le tableau des suffixes. Le tableau des suffixes peut être construit en $O(n \log n)$ dans le pire des cas et $O(n \log\log n)$ en temps moyen [CNP+12].

La détection du plagiat par modèles de suffixes

L'utilisation d'un modèle de suffixes dans la détection du plagiat passe par trois étapes:

1. **Construction d'un modèle de suffixes.** Dans cette étape un modèle de suffixe est construit pour le document suspect et pour chaque document dans la collection de référence, ce qui implique un effort de calcul et d'espace très élevés.

2. **Stratégie de sélection de motif.** Contrairement aux systèmes de recherche d'information, dans le cas d'un système de détection du plagiat le motif recherché est inconnu [Bak95]. Pour cette raison une procédure de sélection des parties suspectes dans un document est nécessaire.

3. **Recherche.** Une recherche est lancée dans tous les modèles des suffixes de la collection de référence pour trouver des passages similaires à ceux des parties suspectes sélectionnées.

Dans le contexte de détection du plagiat, Baker [Bak95] a été parmi les premiers à utiliser les arbres de suffixes pour mesurer la correspondance entre les codes sources. Elle a augmenté les sommets des arbres avec des informations de position qui ont permis de détecter toutes les correspondances avec une longueur maximale. Elle a aussi suggérée l'adaptation de cet algorithme au contexte de détection du plagiat de texte, mais elle n'a pas poursuivi cette application. Monostori et al. [MZS00] construisent un arbre des suffixes pour chaque document à comparer. Par la suite, le taux de correspondance est calculé à l'aide de l'algorithme statistique de Chang et Lawler [CL94]. Khmelev et Teahan [KT03] ont construit un système basé sur les tableaux des suffixes et une fonction nommée la "*R-mesure*" pour calculer la similarité entre deux documents. Cette fonction somme la longueur de tous les suffixes partagés entre les deux documents.

La force des approches de détection du plagiat avec des modèles des suffixes est leur précision dans la détection des chevauchements textuels, *i.e.* si deux documents partagent des sous-chaînes, les modèles des suffixes assurent la détection de ce chevauchement. Cependant, l'inconvénient majeur de ces modèles est qu'ils ne permettent de détecter que des passages exactes ou très proches. Ainsi que l'effort de stockage et de calcul requis est très élevé [CNP+12].

2.3.3 Détection du plagiat à base d'un modèle vectoriel classique

Dans cette famille d'approches, la détections du plagiat est basée sur l'analyse des occurrences de mots dans les documents. En effet, la représentation des documents par des modèles vectoriels classiques (sacs de mots) est utilisée pour mesurer la similarité globale entre documents. Le modèle vectoriel est fréquemment mis en œuvre dans le domaine de RI, notamment pour l'indexation et la classification des documents [SB88]. Nous présentons tout d'abord le principe général du modèle vectoriel. Par la suite, nous discutons sa mise en œuvre pour la détection du plagiat.

Un Modèle Vectoriel (MV) (en anglais Vector Space Model (VSM)) permet de représenter un document par un vecteur v dans un espace à n *dimensions*, où chaque entrée i dans v correspond à un terme du vocabulaire, et n correspond au nombre de termes dans le vocabulaire du modèle. La valeur de v_i représente le poids affecté au

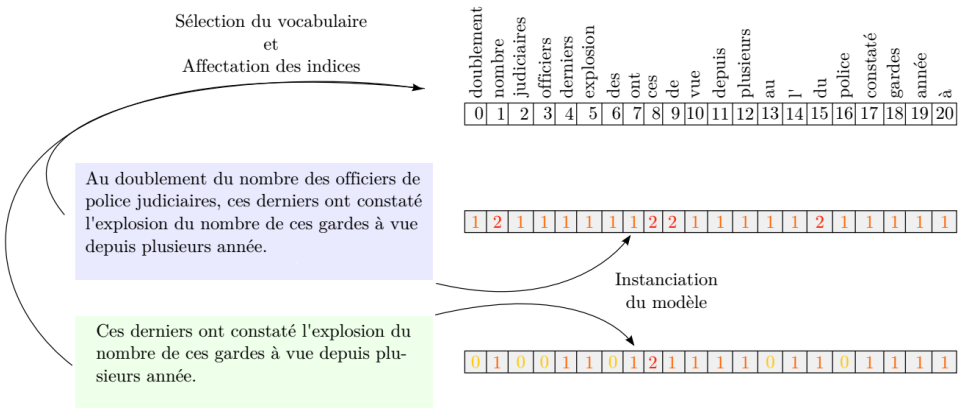


Figure 2.5: Un exemple d'un modèle vectoriel pour deux documents issu de [Pou11].

terme *i* du vocabulaire; ce poids permet d'évaluer l'importance du terme dans un document. Plusieurs méthodes de pondération sont utilisées dans la littérature telles que : le nombre d'occurrences de ce terme dans le document (Term frequency *TF*), ou l'importance du terme dans l'ensemble de la collection (Inverse Document Frequency *IDF*), l'hybridation de ces deux pondérations (*TF-IDF*) est souvent utilisée dans les modèles vectoriels [C+03, UD03]. Un exemple d'un modèle Vectoriel est présenté par la figure 2.5. Dans cet exemple, le poids de chaque terme est le nombre d'occurrences de ce terme dans un document.

Les systèmes de détection du plagiat utilisent souvent un seul modèle d'espace vectoriel pour représenter la totalité du document (MV global) [Dre07, HZ03, SLL97]. Cependant, certains systèmes utilisent plusieurs modèles qui codent des paragraphes ou des phrases (MV local) pour effectuer une évaluation de similarité plus locale [ZMKG09, KGH06, HKF+10]. Cette approche augmente la précision de la détection, mais elle est plus coûteuse en termes d'effort de calcul.

La plupart des systèmes de détection du plagiat basés sur le MV considèrent les mots comme termes [Dre07, HZ03], cependant n'importe quelle entité linguistique peut être utilisée comme terme, par exemple, les n-grammes de mots [BBC+09, DRRA10] ou les phrases [KGH06, ZMKG09]. Avant la construction du modèle, les termes subissent le plus souvent une phase de pré-traitement (normalisation) qui peut inclure racinisation, lemmatisation et la suppression de mots vides [HZ03].

Mesurer la similarité entre deux documents est assurée par des fonctions de similarités entres leurs MVs. Un score de similarité (généralement compris entre [0,1]) est associé à chaque couple de vecteurs. La plupart des travaux utilisent la mesure de similarité cosinus [Dre07, HKF+10, ZMKG09]. Cependant, des fonctions de similarité plus complexes intègrent des informations sémantiques pour identifier le plagiat déguisé, par exemple Kang et al. [KGH06] proposent une fonction de similarité qui évalue les correspondances entre mots au niveau des phrases, y compris les synonymes et les chevauchements entre les vecteurs de mots. Ainsi, Tsatsaronis et al. [TVGK10], Pera et Ng [PN11] utilisent WordNet [Mil95] et l'encyclopédie Wikipedia[1] pour donner un poids supplémentaire aux termes sémantiquement liés. Certains travaux ont expérimenté des fonctions probabilistes telles que Kullback-Leibler (KL) divergence [YC05] et likelihood [MBC+05]

Dans le contexte de détection du plagiat les MVs ont prouvé leurs performances pour identifier le chevauchement textuel. Par exemple Hoad et Zobel [HZ03] proposent un MV à fréquences relatives (*Relative Frequency Model (RFM)*) pour la détection du plagiat. Ce modèle a été initialement proposé par Shivakumar et Garcia-Molina [SGM95]. Pour construire les vecteurs des documents, le modèle RFM sélectionne seulement les termes qui apparaissent dans les deux documents avec un nombre d'occurrence supérieurs à un seuil donné. Par la suite, la détection de similarité est effectuée avec l'utilisation d'une la fonction cosinus [SB88] entre vecteurs.

Le principe de termes partagés avec un nombre comparable de fois est aussi utilisé par Stein [SZE06] pour créer un modèle nommé *Signature Floue (SF)*. Toutefois, Stein propose une normalisation par rapport à la collection de documents, et non entre les couples de documents. Ce modèle représente chaque document par un ensemble de classes de préfixes (entre 10 et 30 classes) au lieu d'un ensemble de termes.

2.3.4 Détection du plagiat à base des word embeddings

Le plongement de mots ou plongement lexical (en anglais *Word Embeddings (WE)*) permet de représenter les mots par des vecteurs dans un espace multidimensionnel, en prenant compte le contexte des mots dans le texte. Cette représentation est principalement issue des modèles de langue neuronaux [BDVJ03], et elle se fonde sur la

[1] https://en.wikipedia.org(consulté le 07/01/2018 à 13h)

construction d'un modèle sémantique qui projette les termes d'une langue dans un espace dans lequel certains liens sémantiques entre ces termes peuvent être observés et mesurés. L'avantage majeur de ce modèle sémantique est qu'il permet de conserver les propriétés sémantiques et syntaxiques de la langue [MYZ13]. Il faut noter que plusieurs appellations sont utilisées pour désigner la notion de *word embeddings*, par exemple, le plongement de mots [VCR15], le plongement lexical [Bra15], la représentation vectorielle continue des mots [JMDL16] ou la représentation distributionnelle distribuée continue de mots [Fer17]. Afin d'éviter toute ambiguïté terminologique, nous choisirons d'utiliser dans ce qui suit l'appellation populaire anglaise *word embeddings*.

2.3.4.1 Modèles existants

Dans la littérature, plusieurs techniques ont été proposées pour construire des modèles *word embeddings*; les plus connus sont:

Collobert et Weston [CW08] présentent une architecture unifiée pour le Traitement Automatique du Langage Naturel (TALN) basée sur un réseau neuronal profond et entraîné simultanément sur quatre tâches classiques: annotation en parties du discours, segmentation en chunks, reconnaissance d'entités nommées et l'étiquetage en rôles sémantiques. Le modèle de [CW08] est stocké dans une matrice $M \in R^{d*|D|}$, où D est un dictionnaire de tous les mots uniques dans le corpus d'apprentissage, où chaque mot est représenté par un vecteur de d dimensions. Indépendamment, une idée similaire a été proposée et utilisée par Turian et al. [TRB10].

Mnih et Hinton [MH09] proposent une autre forme pour représenter les mots dans un espace vectoriel continu, nommé HLBL (de l'anglais Hierarchical Log-Bilinear Model). Comme pratiquement tous les modèles de langue neuronaux, HLBL est un modèle probabiliste qui représente chaque mot avec un vecteur de caractéristiques à valeurs réelles. Pour les n-grammes de mots, HLBL concatène les (n-1) premiers mots $(w_1..w_{n-1})$ et entraîne le modèle linéaire de neurones pour prédire le mot w_n.

Mikolov et al. [MYZ13] utilisent le réseau de neurones récurrent de Mikolov et al. [MKB+10] et une version simplifiée de Bengio et al. [BDVJ03] pour apprendre un modèle *word embeddings*, nommé *"word2vec"*. Dans le modèle *"word2vec"* les

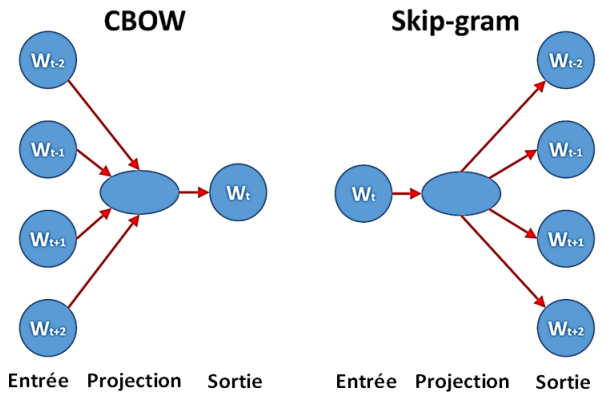


Figure 2.6: L'architecture CBOW [MCCD13] et Skip-gram [MSC+13]

poids du réseau récurrent entraîné sont utilisés comme une représentation vectorielles des mots. Deux architectures neuronales sont proposées pour apprendre le modèle *"word2vec"*: une dénommée sac-de-mots continu *CBOW* (de l'anglais Continuous Bag-Of-Words) [MCCD13] et l'autre *Skip-gram* [MSC+13]. L'objectif du *CBOW* est de prédire un mot à partir de son contexte d'apparition, en utilisant une fenêtre de mots. Soit une suite de mots $S = w_1, w_2, ..., w_i$, le modèle CBOW permet de prédire tous les mots w_k de leurs environnants de mots $(w_{k-l}, , w_{k-1}, w_{k+1}, ..., w_{k+l})$. En revanche, la deuxième architecture Skip-gram permet de prédire, pour un mot donné, le contexte dont il est issu (mots environnants). Ces deux architectures sont illustrées par la figure 2.6.

Pennington et al. [PSM14] présentent un nouveau modèle, dénommée *Global Vectors (GloVe)*. Dans ces travaux, ils ont montré qu'il n'était pas nécessaire d'utiliser des réseaux de neurones pour trouver presque les même résultats. GolVe repose ainsi sur l'utilisation de co-occurrences entre les mots pour construire une matrice de co-occurrences M. Par la suite, la matrice M est utilisée pour calculer la probabilité que le mot w_i apparaît dans le contexte d'un autre mot w_j, cette probabilité $P(i/j)$ représente la relation entre les mots.

Récemment, plusieurs travaux ont utilisé le modèle *word embeddings* dans différentes tâches de TALN, par exemple, la similarité sémantique entre textes [NFS17a, FABS17a, ABC+15, TZLW17, WHJ+17a], la classification [TWY+14, ZZL15], [YMO17] et la détection du plagiat [FABS17b]. Les systèmes de détection de plagiat basés sur les

word embeddings utilisent le modèle généré afin d'identifier et de mesurer les relations sémantiques entre textes. La plus part de ces systèmes utilisent le modèle *word embeddings* afin d'identifier les relations sémantiques entre les mots comme par exemple la synonymie ou le changement de nombre et de genre.

2.3.4.2 PROPRIÉTÉS DU MODÈLE WORD EMBEDDINGS

Dans le modèle *word embeddings* les mots sont représentés par des vecteurs dans un espace multidimensionnel (en prenant en compte le contexte des mots). En effet, Mikolov et al. [MYZ13] ont observé un aspect très important concernant les régularités sémantiques et syntaxiques dans ce modèle : ils ont montré que les angles entre les projections des mots sont corrélés aux relations sémantiques qui les relient. Par exemple, « *singulier/pluriel* », « *féminin/masculin*» ou « *pays/capitale* ». Grâce à cette observation, nous pouvons exploiter ces relations avec des opérations arithmétiques simples sur ces vecteurs.

La relation sémantique entre deux mots w_i et w_j (e.g. la synonymie ou le changement de nombre et de genre) est obtenue en comparant leurs représentations vectorielles v_i et v_j en utilisant la similarité cosinus, la distance euclidienne, la distance manhattan ou n'importe quelle autre fonction de similarité.

Illustration. Soient les trois mots: الجامعة (l'université), المساء (le soir) et الكلية (la faculté). La similarité sémantique entre eux est mesurée en calculant la similarité cosinus entre leurs vecteurs comme suit:

$$sim(\text{الجامعة}, \text{المساء}) = cos(v(\text{الجامعة}), v(\text{المساء})) = 0,13$$
$$sim(\text{الجامعة}, \text{الكلية}) = cos(v(\text{الجامعة}), v(\text{الكلية})) = 0,72$$

L'interprétation est que les mots الكلية (*la faculté*) et الجامعة (*l'université*) sont sémantiquement plus proches que les mots المساء (*le soir*) et الجامعة *(l'université*).

Une autre propriété très intéressante du modèle *word embedding* est l'additivité de ces vecteurs [MYZ13]. Par exemple, soient les six mots ملك (roi), رجل (homme), ملكة (reine), إمرأة (femme) الملوك (rois) et الملكات (reines). On observe que le vecteur le plus proche de $v(\text{ملك}) + v(\text{رجل}) - v(\text{إمرأة})$ est le vecteur $v(\text{ملكة})$. On obtient ainsi, la relation $v(\text{ملك}) - v(\text{ملكة}) \approx v(\text{رجل}) - v(\text{إمرأة})$. Ceci indique que le modèle *word embeddings* capture le changement de genre grammatical [MWSC15]. De plus, on constate que la distance entre le vecteur $v(\text{الملوك}) - v(\text{ملك}) \approx v(\text{الملكات}) - v(\text{ملكة})$, cequi veut

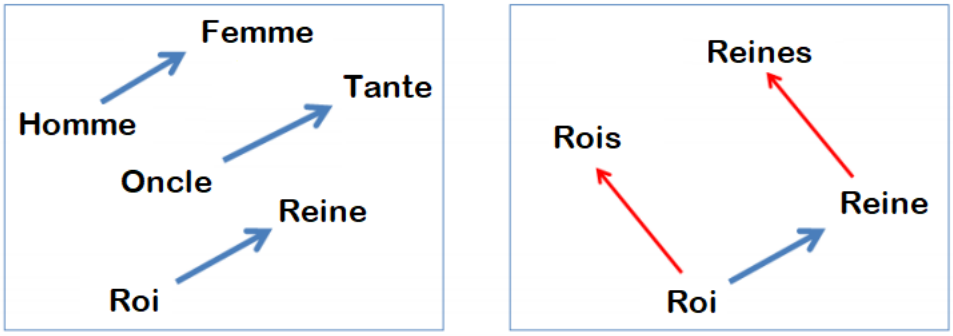


Figure 2.7: Exemple de relation genre et singulier/pluriel entre les vecteurs de mots issue de [MYZ13]

dire qu'on peut identifier la relation singulier/pluriel avec ce modèle. Ce phénomène est illustré dans la figure 2.7.

2.3.5 Détection du plagiat à base des citations

L'efficacité d'un système de détection du plagiat est mesurée en fonction du type de plagiat détecté. Idéalement, le système de détection du plagiat devrait détecter à la fois la similarité lexicale et sémantique entre documents. Par conséquent, la capacité de détection dépend fortement de la fonction de similarité. La fonction de similarité dépend également de caractéristiques des marqueurs textuels à analyser pour calculer le taux de similarité (score de similarité). Considérant chaque entité identifiable d'un texte comme un marqueur textuel potentiel, en effet, il existe deux types de marqueurs: des marqueurs *dépendants de la langue*, par exemple : les caractères, mots, n-grammes de caractères/mots, *etc.* Le deuxième type sont les marqueurs *indépendants de la langue*, comme par exemple : les formules, les dates et les citations [Gip14]. La *Détection du Plagiat par Citation (DPC)* est visiblement basée sur le deuxième type de marqueurs (les citations). Contrairement aux autres techniques précédemment décrites, cette technique est totalement indépendante du lexique et de la langue utilisée.

La détection du plagiat par les citations utilise la similarité structurelle à base des citations comme un indicateur de similarité sémantique entre documents. Un *Système de Détection du Plagiat par Citation (SDPC)* repose principalement sur un processus en trois étapes [Gip14]: (*i*) tout d'abord le *SDPC* extrait la séquence de citations du document suspect ($Seq_{suspect}$) et source (Seq_{source}); (*ii*) un algorithme d'alignement

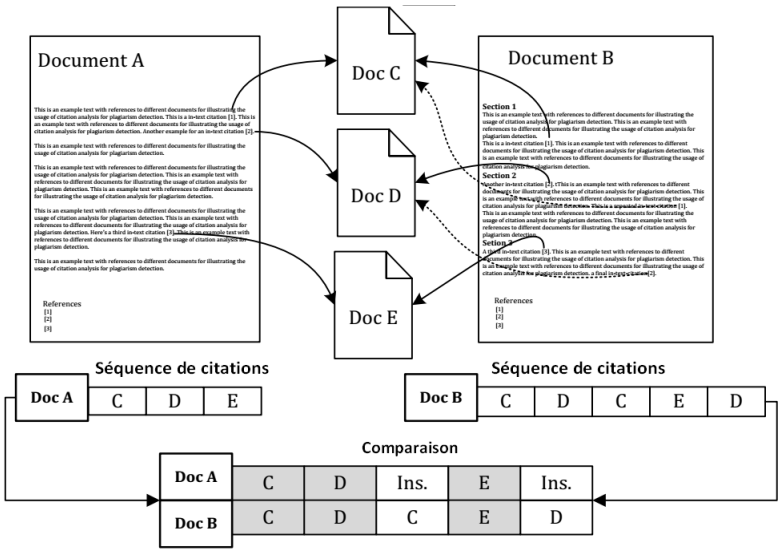


Figure 2.8: Concept général d'un SDPC issu de [Gip14]

de séquence est utilisé (p. ex. Needleman-Wunsch [NW70]) pour aligner les deux séquences Seq_{source}, $Seq_{suspect}$; (*iii*) les deux séquences alignées sont comparées pour détecter un potentiel plagiat.

La figure 2.8 illustre le concept général d'un système de détection du plagiat avec les citations. Dans cet exemple les séquences de citations des deux documents Doc_A (source) et Doc_B (suspect) sont les suivantes: $Seq_{source} = CDE$, $Seq_{suspect} = CDCED$, "*Ins.*" représente les citations non partagées et la sous-séquence de citations partagées entre Doc_A et Doc_B est CDE.

Il est important de noter l'existence de certaine techniques utilisées par les auteurs pour contourner les *SDPCs* comme : la transposition, écaillage (*Scaling*), l'insertion et la substitution de citations.

- **Transposition.** L'ordre de citations dans le document source peut être transposé par le plagieur. Cette transposition peut prendre plusieurs formes comme le changement de styles de citation ([1,2,3] $\Longrightarrow$ [1-3]) ou le réarrangement ([1,2,3] $\Longrightarrow$ [2,1,3]).
- **Écaillage.** C'est le fait d'utiliser la même citation plus d'une fois, par exemple, si la séquence de citations du document source est [1], [2], [3], [4] le document suspect

sera paraphrasé de façon que la séquence suspecte soit : [1], [3], [1], [2], [4], [1].

- **Insertion et Substitution** Les plagieurs peuvent insérer des citations non partagées supplémentaires ou remplacer les citations existantes par des citations sémantiquement similaires. Par exemple, si la séquence source est : [1], [2], [3], [4], le document suspect sera représenté par la séquence suspect suivante: [1], [x_1], [3'], [x_2], [2], [4'], [x_3], où les x_i représentent les citations non partagées insérées et [3'], [4'] sont des citations non partagées sémantiquement similaires à [3] et [4], i.e [3], [4] sont des citations contiennent les mêmes informations que [3] et [4].

Pour lutter contre ces trois techniques de plagiat déguisées, Gipp et al. [Gip14] proposent deux systèmes de détection du plagiat avec citation. lLe premier est basé principalement sur la proximité et l'ordre des citations [GB09], tandis que le deuxième *SDPC* [GB10] utilise le couplage bibliographique [AJ08], la mesure *GCT* (de l'anglais Greedy Citation Tiling) [Wis93], Citation Chunking [zG09] et la plus longue séquence de citations commune (LCS) [CR03].

2.4 Détection du plagiat translingue

Le plagiat translingue (*en anglais cross-lingual*) est une forme de plagiat par traduction, c'est-à-dire que le document plagié est le résultat d'une traduction (manuelle ou automatique) d'un document original dans une autre langue [BC10]. La tâche de détection du plagiat translingue est relativement difficile compte tenu de la diversité des langues et les propriétés lexicales, syntaxiques et sémantiques de chaque langue [FABS17b, NFSC17]. Dans la littérature plusieurs approches sont proposées pour la détection de similarité translingue; nous pouvons les classer en fonction de la stratégie utilisée pour détecter une telle similarité en cinq classes : approches basées sur la syntaxe, les corpus parallèles et comparables, les dictionnaires, et la traduction automatique [FABS16]. La figure 2.9 illustre une taxonomie de différentes approches de détection de similarité translingue. Dans ce qui suit nous fournirons une brève présentation de chacune de ces classes.

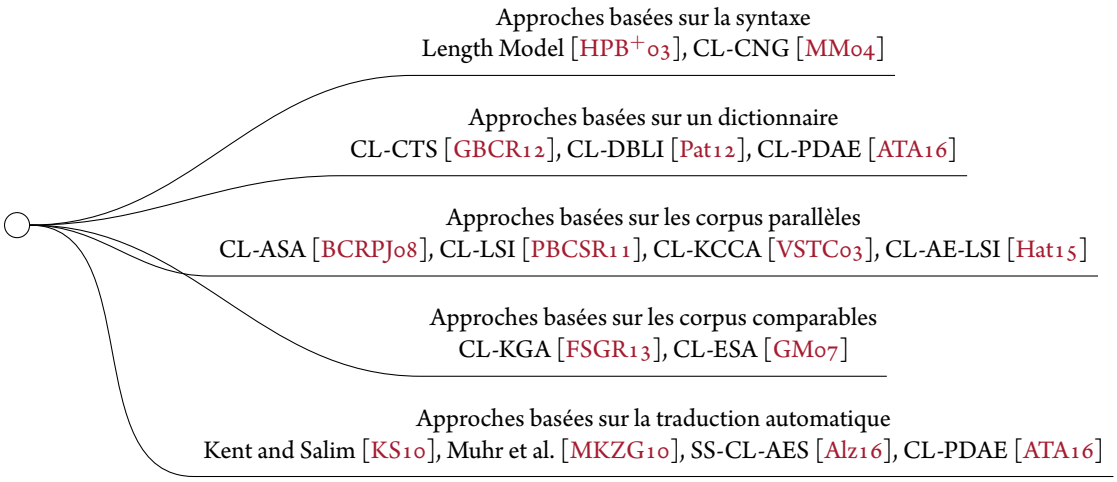


Figure 2.9: Taxonomie des différentes approches de détection de similarité translingue [NFSC17], issu des travaux de Potthast et al. [PBCSR11] et Danilova [Dan13]

2.4.1 Approches basées sur la syntaxe

Concernant les approches basées sur la syntaxe, l'idée clé consiste à comparer les textes multilingues sans traduction. Par exemple, Pouliquen et al. [HPB+03] proposent un modèle de longueur de textes *LM* (de l'anglais *Length Model*) pour estimer la similarité entre textes dans plusieurs langues. Il est principalement basé sur la comparaison de la taille des textes. Ils ont observé le fait que les longueurs des textes dans différentes langues sont étroitement liées par un facteur. Ainsi, ils ont montré l'existence d'un facteur différent pour chaque paire de langues.

McNamee et Mayfield [MM04] introduit le modèle *C-CLN* (*Cross-Language Character N-Gram*) afin de comparer deux textes en utilisant leur représentation par vecteurs de n-grammes. *C-CLN* permet d'obtenir de bonnes performances avec des langages proches les uns des autres, à cause des racines communes.

2.4.2 Approches basées sur un dictionnaire

Dans cette classe d'approche, la similarité entre textes est mesurée en construisant un modèle d'espace vectoriel des unités textuelles. En effet, un vecteur de concepts est construit pour chaque unité textuelle à l'aide d'un dictionnaire ou de thésaurus. Ensuite, la similarité entre les vecteurs de concepts est mesurée en utilisant la similarité cosinus, la distance euclidienne, ou n'importe quelle autre mesure de similarité

[Dan13].

Gupta et al. [GBCR12] présentent un modèle de similarité conceptuel multi-lingue basé sur un thésaurus *CL-CTS* (de l'anglais Cross Language Conceptual Thesaurus-Based Similarity model) pour mesurer la similarité entre textes écrits en trois différentes langues : espagnol, anglais et allemand. Après une phase de pré-traitement qui inclut la racinisation, lemmatisation et la suppression de mots vides. Le modèle *CL-CTS* projette les textes dans l'espace des domaines spécifiques *EuroVoc*[2], afin de construire des vecteurs de concepts. Ensuite, la similarité entre textes est estimée avec la similarité cosinus entre ces vecteurs.

Dans un autre travail, Pataki [Pat12] présente un modèle indépendant du langage basé sur un dictionnaire nommé *CL-DBLI* (de l'anglais *Cross Language Dictionary-Based Language Independent*). Le modèle *CL-DBLI* définit une fonction de distance entre les phrases basée sur un dictionnaire de synonymes et de traduction. Dans ce modèle, les phrases sont évaluées en plusieurs étapes afin d'améliorer la précision de calcul. Ainsi, les principales étapes sont : la désambiguïsation lexicale, la traduction des synonymes (*i.e.* la génération de toutes les traductions possibles) et enfin la comparaison entre les phrases traduites.

2.4.3 Approches basées sur les corpus comparables

Dans cette classe d'approches, les corpus comparables sont utilisés pour construire un modèle translingue. Par exemple, Gabrilovich et Markovitch [GBBMLY12] présentent un modèle d'analyse sémantique explicite translingue. Ce modèle repose sur l'analyse sémantique explicite des termes (*Explicit Semantic Analysis*, *ESA*). L'*ESA* permet d'indexer ou de classer un texte donné par rapport à un ensemble de catégories externes explicitement données. Ainsi, elle indique à quel point un mot donné dans le texte d'entrée est associé à un document spécifique dans ce corpus. Dans ce modèle, deux textes sont liés sémantiquement bien qu'ils n'aient aucun mot en commun. Potthast et al. [PBCSR11] utilisent *Wikipedia* comme un corpus comparable pour estimer la similarité de deux documents en calculant la similarité de leurs représentations *ESA*.

Franco-Salvador et al. [FSGR13] introduisent le modèle *CL-KGA* (de l'anglais *Cross-Language Knowledge Graph Analysis*). *CL-KGA* utilise les graphes de connais-

[2] http://eurovoc.europa.eu/ (consulté le 13/01/2018 à 11h)

sances (*Knowledge Graphs*) construits à partir du réseau sémantique multilingue *BabelNet* [NP12b] pour représenter les textes de langages différents. La comparaison entre deux textes est effectuée en comparant leur graphes de connaissances.

2.4.4 Approches basées sur les corpus parallèles

En ce qui concerne les modèles basés sur des corpus parallèles, Barrón-Cedeño et al. [BCRPJ08] proposent une approche d'analyse de similarité avec l'alignement translingue (*Cross-Language Alignment Similarity Analysis (CL-ASA)*). *CL-ASA* estime la similarité entre deux unités textuelles en utilisant un dictionnaire statistique bilingue extrait à partir d'un corpus parallèle. La même idée a été indépendamment présentée par Pinto et al. [PCBC+09].

Potthast et al. [PBCSR11] introduisent un modèle translingue basé sur l'indexation sémantique latente (en anglais *Latent Semantic Indexation (LSI)*) [DDF+90]. *LSI* permet de construire des relations entre un ensemble de textes et les termes qu'ils contiennent, en construisant un espace de concepts liés aux textes et aux termes. Ce modèle utilise un corpus parallèle avec la *LSI* pour estimer la similarité entre deux documents écrits avec différentes langues.

Vinokourov et al. [VSTC03] proposent aussi un autre modèle qui repose sur la construction des espaces de concepts par *LSI*; ce modèle est nommée *CL-KCCA* (de l'anglais *Cross-Language Kernel Canonical Correlation Analysis*). *CL-KCCA* définit une nouvelle mesure nommée la *F-corrélation canonique des noyaux* pour comparer deux espaces de concept.

Dans le contexte de la similarité sémantique translingue arabe-anglais. Hattab [Hat15] a aussi utilisée *LSI* pour construire un espace sémantique multilingue arabe-anglais à partir duquel il mesure la similarité sémantique deux textes donnés, l'un en arabe et l'autre en anglais.

2.4.5 Approches basées sur la traduction automatique

Dans la littérature, plusieurs approches sont proposées pour la détection de similarité textuelle translingue en utilisant la traduction automatique [BCGR13], [KS10], [MKZG10],[Alz16], [ATA16] et [NFSC17]. L'idée générale de ces approches repose sur la traduction automatique des textes vers la même langue (langue pivot), par la

suite une comparaison monolingue est effectuée [BCGR13].

Par exemple, Kent et Salim [KS10] exploitent *Google Translate API*[3] pour traduire les textes. Après la suppression des mots vides, la racinisation des mots et la segmentation des textes traduits, [KS10] utilisent la technique d'empreinte digitale (*cf. section 2.3.1*) pour effectuer une comparaison mono-lingue.

Après une traduction automatique des textes vers la langue pivot, Barrón-Cedeño et al. [BCRAL10] proposent d'utiliser un vecteur caractéristiques (sac de mots) avec la pondération *TF-IDF* [SB88] pour représenter chaque texte. Ensuite, une comparaison mono-lingue est effectuée par similarité cosinus entre vecteurs.

Au lieu de traduire les textes, Muhr et al. [MKZG10] remplacent chaque terme dans le texte par ses traductions les plus probables dans la langue pivot. Ils exploitent le corpus *Europarl* [Koe05] pour substituer chaque mot par les 5 traductions les plus probables.

2.5 Métriques d'évaluation

Afin d'évaluer au mieux un système de détection du plagiat, Potthast et al. [PSBCR10] ont proposé les quatre mesures orientées plagiat suivantes: *précision, rappel, granularité* et *plagdet*. Ces métriques sont calculées en utilisant deux ensembles S et R, où $S = \{s_1, s_2, \ldots, s_n\}$ représente les cas réels de plagiat annotés dans le document source et $R = \{r_1, r_2, \ldots, r_m\}$ désigne les cas détectés par le système de détection. Dans le but de mieux illustrer ces métriques, nous reprenons l'exemple de Barrón-Cedeño [BC10] (voir figure 2.10).

2.5.1 Précision

Représente la proportion des passages détectés par le système qui sont réellement des cas de plagiat sur l'ensemble de tous les passages détectés. La précision est déterminée par la formule (2.1).

$$\text{Précision}(S, R) = \frac{1}{|R|} \sum_{r \in R} \frac{|\bigcup_{s \in S}(s \cap r)|}{|r|} \tag{2.1}$$

[3] https://cloud.google.com/translate/ (consulté le 13/01/2018 à 19h)

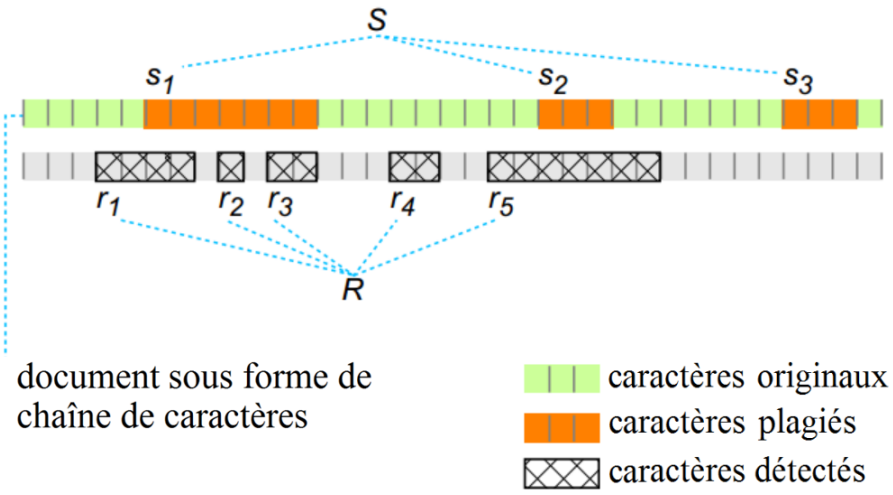


Figure 2.10: Résultats d'un système de détection du plagiat issu de [BC10]

2.5.2 Rappel

La proportion des passages détectés par le système par rapport à l'ensemble des passages plagiés. Le rappel est donné par la formule (2.2).

$$Rappel(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\bigcup_{r\in R}(s\cap r)|}{|s|} \tag{2.2}$$

2.5.3 Granularité

Il est important de noter que les systèmes de détection du plagiat détectent parfois des passages qui se chevauchent pour le même cas de plagiat. Pour résoudre ce problème, une métrique de granularité est également utilisée. La granularité détermine si un passage est détecté par morceaux ou en totalité.

$$\text{Granularité}(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_S| \tag{2.3}$$

avec $S_R \subseteq S$ est l'ensemble des cas plagiat qui ont été détectés, et $R_s \subseteq R$ sont les passages retrouvés plagiés pour un cas s donné, tel que:

$$\begin{cases} S_R = \{s \mid s \in S \wedge \exists r \in R : \ r \text{ détecte } s\}. \\ R_s = \{r \mid r \in R \wedge r \text{ détecte } s\}. \end{cases}$$

2.5.4 Plagdet

Le *plagdet* (*plagiarism detection*) est une mesure combinant le rappel, la précision ainsi que la granularité. Il permet un classement absolu entre les différents systèmes de détection du plagiat [PSBCR10].

$$Plagdet(S,R) = \frac{F_{mesure}}{\log_2(1 + \text{Granularité}(S,R))} \tag{2.4}$$

avec F_{mesure} combine la précision et le rappel est leur moyenne harmonique. Comme le montre la formule (2.5).

$$F_{mesure} = 2 * \frac{\text{Précision} * Rappel}{\text{Précision} + Rappel} \tag{2.5}$$

2.6 Conclusion

Nous avons présenté dans ce chapitre les méthodes fondamentales expérimentées dans la littérature pour la tâche de détection du plagiat. Deux grandes approches se distinguent : la détection extrinsèque et intrinsèque. La détection extrinsèque consiste à comparer un document suspect avec une collection référence pour trouver des passages similaires. La détection intrinsèque utilise une étude statistique fondée sur les caractéristiques linguistiques d'un texte afin de caractériser les variations stylistiques de l'auteur.

En second lieu, nous avons présenté en détail les méthodes de détection du plagiat monolingue et translingue. Nous avons présenté également les métriques d'évaluation utilisées afin d'évaluer au mieux un système de détection du plagiat, à savoir la précision, le rappel, la granularité et le plagdet.

Dans le chapitre suivant, nous décrivons les défis associés au traitement automatique de l'arabe, ainsi que le problème de détection du plagiat dans les documents arabes. Nous présenterons également les systèmes de détection du plagiat dans les documents arabes décrits dans la littérature.

v

" وسعْت كتاب الله لفظا وغاية ** وما ضقْت عن آي به وعظات
فكيف أضيق اليوم عن وصف آلة ** وتنسيق أسْماء لمخترعات
أنا البحر في أحشائه الدرّ كامن ** فهل سألوا الغوّاص عن صدفاتي"
قصيدة اللغة العربية تنعى حظها بين أهلها ، لحافظ إبراهيم

3

Détection de plagiat dans les documents arabes

Dans ce chapitre, nous introduisons les caractéristiques de la langue arabe. Nous discutons aussi les difficultés rencontrées et les défis associés au traitement automatique de la langue arabe. C'est sur cette langue en particulier, que nos travaux se porteront. Ainsi, nous présenterons dans la suite de ce chapitre les systèmes de détection du plagiat dédiés pour la langue arabe.

3.1 Particularité de la langue arabe

La langue arabe est une langue sémitique qui s'écrit et se lit de droite à gauche. Elle est une langue officielle dans 22 pays et la langue maternelle de plus de 330 millions de personnes. En 2017, l'arabe était considérée comme la quatrième langue la plus utilisée sur Internet [Saâ15]. Elle est aussi l'une des six langues officielles des Nations Unies (ONU)[1].

[1] http://www.un.org/fr/sections/about-un/official-languages/ (consulté le 18/12/2017 à 10h)

Début	Milieu	Fin	Isolée
ضـ	ـضـ	ـض	ض
ضوء	الضب	مريض	غرض

Table 3.1: Différentes représentation de la lettre ض selon sa position dans le mot.

3.1.1 Systèmes d'écriture de l'arabe

3.1.1.1 Alphabet

L'alphabet de l'arabe compte 28 lettres, avec 25 consonnes et 3 voyelles longues " ا", "و" et "ي" [Saâ15]. La forme morphologique de la langue arabe est très complexe en raison de la variation morphologique de ces lettres ainsi que le phénomène d'agglutination [KGK01]. Les lettres changent de forme d'écriture selon leurs positions dans le mot. On note quatre positions *début, milieu, fin* et *isolée*. Le tableau 3.1 illustre la représentation de la lettre "ض" selon sa position dans le mot.

3.1.1.2 Voyellation

L'arabe comporte aussi cinq signes diacritiques (voyelles courtes) : *Fatha* "ـَ" , *Damma* "ـُ", *Kasra* "ـِ", *Chadda* "ـّ" et *Sukûn* "ـْ" , ainsi que la combinaison de ces derniers engendre d'autres signes de voyellation comme: "ـٌ", "ـّ", "ـُ" , *etc.* Les signes de voyellation sont notés sous la forme de signes diacritiques placés au-dessous ou au-dessus des lettres, par exemple: "إِنَّمَا يُؤمِنُ بِآيَاتِنَا". De plus, les mots indéfinis (mots sans l'article défini "AL"(ال)) peuvent prendre des diacritiques spéciaux (la double voyelle) à la fin du mot notées par "nounatation" ou "tanwine" (التنوين) . En effet, le tanwine est représenté à la fin du mot par la double utilisation des trois signes diacritiques de base Fatha, Damma et Kasra: "ـً", "ـٌ", "ـٍ". On peut voir sur le tableau 3.2, les différents types de voyelles courtes en arabe avec leurs représentations.

Généralement, les diacritiques sont facultatifs, *c.-à-d.* un texte arabe peut être représenté avec ou sans diacritique. Le fait que la présence soit optionnelle des diacritiques peur générer une ambiguïté lexicale, car chaque lettre dans un mot arabe peut prendre une des voyelles courtes possibles, ce qui crée un nombre important de combinaisons de mots. D'après [Chaoo] l'absence de voyellation en arabe engendre

Voyelle courte	Nom en arabe	Nom en français	Exemple
ـَ	الفتحة	Fatha	سَراب
ـُ	الضمة	Damma	رُيُوع
ـِ	كسرة	Kasra	مبادِئ
ـْ	سكون	Sukun	صمْت
ـّ	شدة	Chadda	اللّيل
ـً	فتحتين (تنوين)	Fathayne (Tanwine)	رجاءً
ـٌ	ضمتين (تنوين)	Dammatayne (Tanwine)	حسنٌ
ـٍ	كسرتين (تنوين)	Kasratayne (Tanwine)	صديقٍ
ـَّ	شدة مفتوحة	Chadda+ Fatha	الشَّمس
ـُّ	شدة مضمومة	Chadda+ Damma	جدٌّ
ـِّ	شدة مكسورة	Chadda+ Kasra	معلِّم
ـًّ	شدة مع تنوين الفتح	Chadda+ Tanwine	حيًّا
ـٌّ	شدة مع تنوين الضم	Chadda+ Tanwine	دُريٌّ
ـٍّ	شدة مع تنوين الكسر	Chadda+ Tanwine	حَظٍّ

Table 3.2: Différents types de voyelle courte en arabe

	Interprétation I		Interprétation II		Interprétation III	
كتب	كَتَبَ	Il a écrit	كُتِبَ	Il a été écrit	كُتُبٌ	Des livres
علم	عَلَمُ	Drapeau	عِلْمٌ	Savoir	عَلَّمَ	Enseigne

Table 3.3: Ambiguïté causée par l'absence des diacritiques pour les mots كتب et علم

une complexité de calcul très importante et plus grande que ses homologues langues latines. Le tableau 3.3 montre deux exemples d'ambiguïté causée par l'absence de voyelles.

3.1.2 Morphologie de la langue arabe

Dans la langue arabe certains mots sont simples comme كَلَّمَ (Il a parlé à), certains sont plus complexes comme يتكلمون= يت + كلم + ون (Ils parlent) et d'autres sont invariables (sans diacritiques) comme par exemple: le verbe ذهب (aller) et le nom ذهب (or), le verbe كتب (a écrit) et le nom كتب (livres), *etc.*

La morphologie est l'étude de la structure interne et de la formation des mots et de leurs variation [Hab10]. Cette étude s'intéresse à la façon de décomposition d'un

mot en plusieurs unités nommées morphèmes ou unités morphologiques. En effet, la langue arabe dispose de trois façons pour déterminer les morphèmes d'un mot qui sont l'agglutination, la flexion et la dérivation [Ham15].

3.1.2.1 Morphologie agglutinante

La langue arabe est également très différente par rapport aux langues latines, elle est fortement agglutinée dans le sens où les pronoms et les prépositions et les articles collent aux verbes, noms et particules auxquels ils se rapportent [Kou91]. Ce phénomène d'agglutination en arabe peut générer des mots assez complexes, comme par exemple le mot ﴿فَأَسْقَيْنَاكُمُوهُ﴾ (سورة الحجر ﴿٢٢﴾) , ce mot est transcrit par quatre mots en français : *"dont nous vous abreuvons"*.

3.1.2.2 Morphologie flexionnelle

L'arabe est une langue hautement flexionnelle. Cette flexion est liée à la variation de la forme des mots en fonction de plusieurs facteurs grammaticaux [Ham15]. La flexion en arabe est employée pour la conjugaison du verbe, la déclinaison du nom, de mode, de genre, de temps, de nombre et de cas, qui sont en général des suffixes et préfixes [BGD60].

En effet, la flexion d'un mot repose sur la concaténation d'affixes à un radical (base) pour construire une forme fléchie. Le radical est obligatoire dans le mot, il représente la base du mot. Donc, les mots peuvent être générés par la concaténation d'une base et un affixe. Un affixe peut être un préfixe qui se situe avant la base, un suffixe (après la base) ou des circonfixes qui entourent la base. Par exemple pour la conjugaison des verbes : le mot يتوقفون(Ils s'arrêtèrent) est le résultat de la concaténation du préfixe "يـ" et le suffixe "ون" avec le radical "توقف". Dans ce cas, le préfixe "يـ" indique le présent et le suffixe "ون" indique que le verbe "توقف" (s'arrête) est conjugué à la troisième personne du masculin pluriel.

3.1.2.3 Morphologie dérivationnelle

Une autre difficulté de traitement automatique de la langue arabe réside dans le fait que l'arabe possède une morphologie dérivationnelle assez riche et complexe. En effet, la dérivation est un phénomène plus complexe que la flexion. Contrairement aux

Modèle	Dérivation	Sens
فعل	كَتَبَ	a écrit
فاعل	كَاتِبْ	écrivain
مفعل	مَكْتَبْ	bureau
مفعلة	مَكْتَبَة	bibliothèque
...	...	...

Table 3.4: Exemples de dérivation du mot كتب

autres langues latines, un mot arabe n'est pas simplement le résultat d'une concaténation de morphèmes lexicaux, comme c'est le cas pour le français (par exemple : infaillibleme nt= in+faillible+ ment). Le processus de dérivation en arabe consiste à former de nouveaux mots à partir des mots existants. Un mot arabe est le résultat de concaténation d'une racine جذر avec combinaison de préfixes (المزيد المقدم), suffixes (المزيد المؤخر), circonfixes (الزائدة أو المزيد) et d'un schème morphologique (وزن). La racine est trilitère ou quadrilitère et elle définit une notion abstraite [Ham15]. Par exemple, la racine صلى est liée à la notion de "*prier*", alors que علم est associée à la notion "*savoir*". En effet, à partir d'une racine de 3 à 4 lettres nous pouvons former jusqu'à 30 dérivés verbaux et nominaux par l'application des règles morphologique [Lar95a]. Le tableau 3.4 illustre un exemple de dérivation du mot كتب (écrire).

3.2 Systèmes de détection du plagiat en langue arabe

Ces dernières années, le problème de la détection du plagiat en langage naturel a suscité l'intérêt de nombreux chercheurs. Plusieurs outils de détection du plagiat ont été développés, notamment pour l'anglais. Les systèmes de détection du plagiat pour l'anglais ont été largement étudiés par Chen et al. [CFL+04] et Maurer et al. [MKZ06], Alzahrani et al. [ASA12], Bin-Habtoor et al. [BHZ12] et Gipp [Gip14].

La détection du plagiat en arabe est particulièrement difficile en raison de la morphologie complexe et la structure syntaxique de l'arabe [Men12]. Afin de surmonter ce défi, plusieurs systèmes de détection du plagiat sont proposés pour l'arabe. Nous présentons dans cette section les systèmes les plus connus de détection du plagiat dans

$<w_i , w_j>$
<وسائل،سيارة>
<نقل،سيارة>
<طائرة،سيارة>
<المواصلات،سيارة>
<وسائل،إحدى>
...

Figure 3.1: Extraction de couples (w_i,w_j) [AS09]

les documents arabes.

3.2.1 *FS-APD*

Alzahrani et Salim [AS09] proposent un système de détection du plagiat en langue arabe basé sur les fondements de la théorie des ensembles flous [OMK91]. Ce système est nommé : *FS-APD* de l'anglais *Fuzzy Statement-based Arabic Plagiarism Detection system*. Le processus de détection de plagiat avec le système *FS-APD* passe par les étapes suivantes :

1. Soient doc_i et doc_j un document original et un document suspect, respectivement. doc_i (*original*): السيارة هي إحدى وسائل الموصلات. (*La voiture est un moyen de transport*). doc_j (*suspect*): وسائل المواصلات تشمل السيارة والطائرة. (*Les moyens de transport comprennent la voiture et l'avion*). La première étape consiste à extraire toutes les paires de mots de doc_i et doc_j comme illustré par la figure 3.1.

2. Dans cette étape, *FS-APD* calcule le coefficient de corrélation C_{ij} pour chaque paire de mots. C_{ij} représente l'étendue de la corrélation entre chaque paire de mots (corrélation mot-à-mot) en utilisant la formule (3.1):

$$C_{ij} = \frac{n_{ij}}{(n_i + n_j - n_{ij})} \tag{3.1}$$

   où n_{ij} est le nombre de documents dans la collection de référence contenant les deux mots, et n_i, n_j représentent le nombre de documents contenant les mots w_i, w_j respectivement. Le résultat de cette étape est une matrice de corrélation mot-à-mot pour les mots dans les documents doc_i et doc_j comme illustré dans la figure 3.2.

$<w_i, w_j>$	سيارة	عربة	نقل	طائرة	احدى
سيارة	1	1	0.7	0.85	0.001
عربة	-	1	0.5	0.6	0.002
نقل	-	-	1	0.7	0.003
طائرة	-	-	-	1	0.002
احدى	-	-	-	-	1
...	-	-	-	-	-

Figure 3.2: Matrice de corrélation [AS09]

3. Ensuite, *FS-APD* calcule le coefficient de corrélation μ_{w_i,doc_j} entre chaque mot w_i dans le document original doc_i et le document suspect doc_j (une corrélation mot-à-document). La corrélation μ_{ij} est obtenue comme suit :

$$\mu_{w_i,doc_j} = 1 - \prod_{w_k \in doc_j} (1 - C_{ik}) \tag{3.2}$$

4. Le taux de similarité entre doc_i et doc_j est obtenu en appliquant la formule (3.3), où n est le nombre de mots dans doc_i:

$$Sim(doc_i, doc_j) = \frac{\sum_{k=1}^{n} \mu_{w_k,doc_j}}{n} \tag{3.3}$$

5. Enfin, le taux de similarité $Sim(doc_i, doc_j)$ est comparé à une valeur de seuil fixe SP pour juger si doc_i et doc_j sont similaires ou non. Le seuil SP est calculé en utilisant la technique proposée par Yerra et Ng [YN05].

Le système *FS-APD* a confirmé son efficacité dans la détection du copier-coller direct avec un *rappel* de 100%. Cependant, les résultats montrent que ce système n'est pas aussi efficace dans le cas d'une reformulation ou d'une substitution par des synonymes avec un *rappel* de 0.23% et 14% respectivement. Pour adresser le problème de détection du plagiat avec reformulation et substitution par des synonymes, Alzahrani et Salim [AS10] proposent aussi le système *FSS-EPD* (*Fuzzy Semantic-based Similarity for Extrinsic Plagiarism Detection*). Cependant, ce système a été testé sur l'anglais et non sur l'arabe.

3.2.2 Iqtebas

Ameera et Elnagar [JE12] présentent un système de détection du plagiat textuel dans les textes arabes basé sur la technique d'empreinte digitale (*cf. section 2.3.1*). Ce système est nommé *Iqtebas* qui vient du mot arabe إقتباس (*citation*). *Iqtebas* estime le taux de plagiat entre deux documents de la façon suivante :

1. Une première étape de pré-traitement des documents est faite. Elle inclut la suppression de mots vides et la racinisation des mots;
2. La segmentation des documents (suspects et sources) en phrases et la génération des 3-*grammes* de mots pour chaque phrase;
3. L'application de la fonction de hachage *Karp-Rabi* [KR87] sur les 3-*grammes* de mot;
4. Afin de construire l'empreinte digitale, Iqtebas utilise l'algorithme winnowing [SWA03] pour sélectionner un sous ensemble de hashes;
5. La construction d'un index inversé [MRS+08] pour accélérer le temps de calcul de similarité, de façon que le dictionnaire de l'index contient l'empreinte de chaque 3-$gramme_i$ et la liste de positions contient tous les identifiants des phrases ($phrases_{id}$) qui contiennent ce 3-$gramme_i$;
6. Enfin, *Iqtebas* utilise un Modèle de Fréquence Relative (MFR) [SGM95] et un Modèle de Proportion d'Inclusion (MPI) [BSLL03] pour estimer le degré de similarité entre l'empreinte digitale du document source (ED_{src}) et le document suspect (ED_{sus}). En effet, le Modèle de Fréquence Relative (MFR) [SGM95] propose une mesure de similarité asymétrique entre deux vecteurs (dans le cas d'*Iqtebas* entre deux empreintes). Cette mesure est dérivée de la similarité symétrique cosinus [SB88] comme illustrée par la formule (3.4). Concernant le Modèle de Proportion d'Inclusion (MPI) [BSLL03], les auteurs proposent aussi une similarité asymétrique exprimée par la formule (3.5).

$$Sim_{MFR}(ED_{src}, ED_{sus}) = \frac{\sum a_i^2 Freq_{src}(hach(3\text{-}gram_i)) Freq_{sus}(hach(3\text{-}gram_i))}{\sum a_i^2 Freq_{src}(hach(3\text{-}gram_i))^2} \quad (3.4)$$

$$Sim_{MPI}(ED_{src}, ED_{sus}) = \frac{\sum \mid Freq_{src}(hach(3\text{-}gram_i)) \cap Freq_{sus}(hach(3\text{-}gram_i) \mid}{\sum \mid Freq_{src}(hach(3\text{-}gram_i)) \mid} \tag{3.5}$$

où a_i est un poids donné pour $hach(3\text{-}gram_i)$; $Freq_{src}(hach(3\text{-}gram_i))$ et $Freq_{sus}(hach(3\text{-}gram_i))$ sont les fréquences d'apparition de $hach(3\text{-}gram_i)$ dans ED_{src} et ED_{sus} respectivement.

Malgré le fait que le système *Iqtebas* a marqué un taux de rappel intéressant (94%) dans la détection de duplication, il reste toutefois inefficace face aux reformulations, la substitution par des synonymes ainsi qu'à la suppression et l'ajout de mots.

3.2.3 Aplag

Afin d'améliorer la détection du plagiat dissimulé comme la substitution par des synonymes et le changement de la structure de la phrase par l'ajout et la suppression de mots, Meni [Men12] présente un système de détection du plagiat nommé *Aplag* (de l'anglais *Arabic Plagiarism detection*). *Aplag* est aussi basé sur la technique de l'empreinte digitale et il partage presque les mêmes étapes que *Iqtebas* à savoir: le prétraitement, la génération des 3-grammes de mots et le hachage pour la construction des empreintes. Toutefois, *Aplag* utilise la base de données lexicales Arabic WordNet [BER+06] pour remplacer chaque mot dans le texte par le synonyme le plus fréquemment utilisé. De plus, *Aplag* propose aussi un modèle heuristique pour comparer le document suspect à différents niveaux hiérarchiques (phrase, paragraphe et document) afin d'éviter les comparaisons inutiles. Les résultats montrent que *Aplag* [Men12] donne de meilleurs scores dans la détection du plagiat dissimulé par rapport au *FS-APD* [AS08] et *Iqtebas* [JE12]. *Aplag* a obtenu un rappel de 94% pour la substitution par des synonymes et de 93% pour le changement de la structure de la phrase. Le tableau 3.5 donne une comparaison entre les deux systèmes *Aplag* et *Iqtebas*.

3.2.4 PDS-AD

Hussein [Hus15] propose un nouveau système de détection du plagiat pour les documents arabes nommé *PDS-AD* (de l'anglais *A Plagiarism Detection System for Arabic*

	Iqtebas [JE12]	*Aplag* [Men12]
Pré-traitement	supp. de mots vides & racinisation	supp. de mots vides & racinisation
Segmentation	phrase	phrase
Unité de Segmentation	3-gramme de mot	3-gramme de mot
Fonction de hachage	Karp-Rabi [KR87]	BKDR function [RKL88]
Fonction de similarité	MFR [SGM95] et MPI [BSLL03]	LCS [BHR00]
Niveau de compassion	phrase	phrase, paragraphe, document

Table 3.5: Iqtebas [JE12] Vs. Aplag [Men12]

Documents) basé sur la modélisation de la relation entre les textes et leurs n-grammes de phrases. Le système *PDS-AD* passe par une phase de pré-traitement qui comprend plusieurs étapes, y compris: la suppression de mots vides, l'étiquetage morpho-syntaxique et la substitution par des synonymes. Ensuite, il introduit un algorithme heuristique de correspondance des paires de phrases afin de construire pour chaque document un *modèle tf-idf* sous forme d'une matrice [SB88]. *PDS-AD* estime le taux de plagiat entre deux *modèles tf-idf*s par analyse sémantique latente (LSA, de l'anglais *Latent Semantic Analysis*) [SE06] et par décomposition en valeurs singulières (SVD, de l'anglais *Singular Value Decomposition*) [Ces08]. Le *PDS-AD* atteint un taux de rappel de 88% dans les cas de plagiat avec restructuration des phrases et de 86% pour la substitution par des synonymes respectivement.

3.3 Campagne d'évaluation AraPlagDet 2015

La campagne d'évaluation *AraPlagDet* (*Arabic Plagiarism Detection Shared Task*) [BBR+15] est la première campagne d'évaluation organisée pour la détection du plagiat en langue arabe. Un avantage très important d'*AraPlagDet* est de permettre l'évaluation de plusieurs systèmes de détection sur le même corpus. Deux sous-tâches de détection du plagiat ont été proposées lors de la campagne *AraPlagDet*.

La première sous-tâche est la détection de *plagiat extrinsèque* en langue arabe nommée *ExtAraDet*. Soient un ensemble de documents suspects (Doc_{sus}) et documents sources (Doc_{src}) potentiels, *ExtAraDet* consiste à repérer des passages plagiés dans les Doc_{sus} et les passages sources correspondants dans les Doc_{src}. La seconde sous-tâche est la détection de plagiat *intrinsèque* en langue arabe nommée *IntAraDet*. Le but de

la sous-tâche *IntAraDet* est d'extraire tous les passages plagiés dans un Doc_{sus} sans les comparer à des documents sources potentiels.

3.3.1 Corpus de la campagne d'évaluation AraPlagDet 2015

Dans le cadre de la campagne *AraPlagDet*, deux corpus ont été proposés pour la détection de plagiat. Le premier corpus, nommé *Cr-ExtAra*, est dédié à la détection de plagiat extrinsèque. Le second corpus, appelé *Cr-IntAra*, est conçu pour la sous-tâche de détection du plagiat intrinsèque.

Le corpus *Cr-ExtAra*[2] comprend 1171 documents écrits en arabe avec plus de 1727 cas de plagiat. Les documents sont issus principalement de Wikipedia Arabe[3] et le corpus *CCA* (Corpus of Contemporary Arabic) [ASA06]. *Cr-ExtAra* est divisé en deux ensembles: (1) 570 documents sources à partir desquelles des passages de texte sont extraits; et (2) 601 documents suspects dans lesquels les passages plagiés sont insérés directement ou après plusieurs types d'obfuscation comme la reformulation, la substitution par des synonymes ou la restructuration des phrases. Le tableau 3.6 présente des exemples des différents types de plagiat et d'obfuscations utilisées dans les corpus de *Cr-ExtAra*.

Concernant la sous-tâche de détection du plagiat intrinsèque, les organisateurs ont utilisé le corpus *Cr-IntAra*[2] proposé par Bensalem et al. [BRC13]. Dans ce corpus un cas de plagiat est défini par sa position et sa longueur dans le document suspect seulement. *Cr-IntAra* contient 1024 documents avec plus de 2714 cas de plagiat. De plus, Bensalem et al. [BBR+15] proposent aussi deux autres corpus d'entraînement *Cr-EnExtAra* et *Cr-EnIntAra* pour les sous-tâches *ExtAraDet* et *IntAraDet* respectivement. Le tableau 3.7 donne plus de détails sur le corpus *Cr-ExtAra*, à savoir, le nombre de cas de plagiat, le taux de plagiat par document, la taille des documents, *etc.* Plus de détails sur les autres corpus sont disponibles dans le papier de description de la campagne [BBR+15].

Afin d'évaluer les systèmes de détection du plagiat participants dans *AraPlagDet*, il faut que tous les passages plagiés soient étiquetés, de façon que chaque étiquette contient des informations sur la position du début et de fin du passage plagié dans

[2] http://misc-umc.org/AraPlagDet/?i=2datasets (consulté le 11/01/2017 à 10h)
[3] http://ar.wikipedia.org (consulté le 12/02/2017 à 10h)

le document source et suspect. C'est pourquoi Bensalem et al. [BBR+15] proposent d'associer chaque document suspect à un *fichier XML* qui localise la position exacte des passages plagiés (cas de détection du plagiat extrinsèque). Ainsi, pour chaque passage plagié, le fichier XML contient les méta-données suivantes :

- *this_offset*: la position du premier caractère de passage plagié détecté dans le document suspect.
- *this_length*: le nombre de caractères du fragment de plagiat détecté.
- *source_reference*: le document à partir duquel le fragment plagié a été pris.
- *source_offset*: la position du premier caractère du passage plagié détecté dans le document source.
- *source_length*: le nombre de caractères du fragment plagié.
- *obfuscation*: représente le type d'obfuscation utilisé, par exemple substitution de synonymes, paraphrase, *etc.*
- *type*: la façon de génération de ce cas de plagiat. En effet, il existe deux types de plagiat, plagiat artificiel (créé automatiquement) et plagiat simulé (créé manuellement).

```
<?xml version="1.0" encoding="UTF-8"?>
<document reference="suspicious-document0517.txt">
<feature name="plagiarism"
this_offset="345" this_length="226"
source_reference="source-document00072.txt" source_offset="17967" source_length="225"
obfuscation="manual synonym substitution" type="simulated"/>
<feature name="plagiarism"
this_offset="1095" this_length="444"
source_reference="source-document00048.txt" source_offset="4396" source_length="421"
obfuscation="word shuffling" type="artificial"/>
</document>
```

Figure 3.3: Le fichier XML associé au document suspicious-documento517.txt

Par exemple, la figure 3.3 présente les informations contenues dans le fichier XML associé au document suspect *suspicious-documento517.txt* dans le corpus *Cr-ExtAra*. Ces méta-données indiquent que le document suspect *suspicious-documento517.txt* comporte deux passages plagiés p_1 et p_2 à partir des documents *source-documento0072.txt* et *source-documento0048.txt* respectivement. Ces méta-

données nous informent que le passage p_1 commence à la position 1797 avec une taille de 255 caractères (1095 , 444 pour p_2). De plus, ce fichier nous informe également que le premier passage p_1 créé manuellement par substitution de synonymes, tandis que $p2$ est créé artificiellement par mélange de mots.

Type d'obfuscation	Description
Substitution manuelle de synonymes	Remplacer certains mots avec leurs synonymes en utilisant: le vérificateur d'orthographe grammaire et synonymes de Microsoft, le dictionnaire Almaany[4], Arabic WordNet Browser[5], et les synonymes fournis par Google traduction [6].
Ajout et/ou suppression des diacritiques	Les signes diacritiques en arabe sont facultatifs et leur présence ou leur absence est orthographiquement acceptable. Par conséquent, pour chaque mot w de longueur n, nous avons au moins 2^n représentations différentes. Par exemple: القضيّة الفلسطينية القضية الفلسطينية ≡ القضيّة الفلسْطينيّة ≡ القضيّة الفلسْطينيّة
Obfuscation automatique	Le mélange de mots et de phrases sont utilisés pour créer automatiquement des cas d'obfuscation, ex. القضية الفلسطينية مصطلح يشار به للخلاف السياسي والتاريخي. يشار مصطلح القضية الفلسطينية به للخلاف التاريخي و السياسي.
Paraphrase Manuelle	Les passages sont sélectionnés manuellement à partir des documents sources puis paraphrasés manuellement, ex. القضية الفلسطينية مصطلح يشار به للخلاف السياسي والتاريخي والمشكلة الإنسانية في فلسطين بدءا من عام ١٨٧٠. بدء الخلاف السياسي في فلسطين منذ أواخر القرن التاسع عشر مما أدى إلى مشكلة إنسانية أصبحت تعرف بالقضية الفلسطينية

Table 3.6: Différents types d'obfuscation utilisés dans Cr-ExtAra [BBR+15]

3.3.2 Systèmes participants dans AraPlagDet 2015

Dans la campagne d'évaluation *AraPlagDet 2015*, trois systèmes ayant participé à la sous-tâche *ExtAraDet*: Magooda et al. [MMR+15a], Alzahrani [Alz15] et Palkovskii[7]. Seulement deux participants (Magooda et al.[MMR+15a] et Alzahrani [Alz15]) parmi les trois systèmes soumis décrivent leurs méthodes.

Magooda et al. [MMR+15a] présentent un système de détection du plagiat extrinsèque nommé *RDI_RED*. Après une phase de segmentation des documents (suspects et sources), *RDI_RED* utilise un algorithme avec plusieurs paramètres et le moteur de recherche *Lucene* [MHG10] pour sélectionner une liste de documents sources candi-

[7] http://plagiarism-detector.com/ (consulté le 11/03/2018 à 11h)

Cr-ExtAra		
Information générique	Nombre de documents	1171
	Nombre de cas de plagiat	1727
	Documents sources	48.68%
	Documents suspects	51.32%
Plagiat par document	Sans plagiat	28.12%
	Avec plagiat	71.88%
	Peu (1%-20%)	36.94%
	Moyen (20%-50%)	32.95%
	Important (50%-80%)	2.00%
Taille du document	Très court(1 page)	17.51%
	Court (1-10 pages)	76.26%
	Moyen (10 -100 pages)	6.23%
Longueur de cas de plagiat	Très court (300 caractères)	21.25%
	Court (300-1k caractères)	42.50%
	Moyen (1k-3k caractères)	28.26%
	Longue (3k-30k caractères)	7.99%
Type de plagiat et d'obfuscation	Artificiel	88.94%
	Sans obfuscation	40.30%
	Mélange de phrases	10.42%
	Mélange de mots	38.22%
	Simulé	11.06%
	Subst. manuelle par des synonymes	9.79%
	Paraphrase Manuel	1.27%

Table 3.7: Détails du corpus Cr-ExtAra [BBR+15]

dats. Ensuite, les documents candidats sont alignés avec les documents suspects pour détecter les segments plagiés (parties alignées). Enfin, un ensemble de règles est appliqué par un module de filtrage pour éliminer les faux positifs.

Comme le système *RDI_RED* est basé principalement sur le moteur de recherche *Lucene*, il peut être facilement déployé en ligne. Cependant, un inconvénient majeur de *RDI_RED* est l'incapacité de détection du plagiat avec substitution de synonymes ou avec paraphrase.

Le système proposé par Alzahrani [Alz15] passe par cinq étapes principales : (*i*) une étape de pré-traitement des documents qui inclut la segmentation, la suppression des mots vides et la génération des n-grammes de mots, (*ii*) la création d'une empreinte digitale pour tous les documents sources et suspects, (*iii*) récupérer une liste de documents sources candidats pour chaque document suspect en calculant le coefficient de Jaccard [Jac12] entre les empreintes digitales des documents, (*iv*) comparaison approfondie entre les documents suspects et les documents sources candidats

associés en utilisant l'approche de *k-chevauchement* de Alzahrani et Salim [AS10], (v) les segments plagiés sont passés par un module de post-traitement pour fusionner les segments plagiés consécutifs.

Le système Mahgoub et al. [MMR+15b] est le seul participant à la sous-tâche de détection du plagiat intrinsèque *IntAraDet*. Ce système est similaire à celui proposé par Zechner et al. [ZMKG09]. Il est principalement basé sur la construction d'un modèle d'espace vectoriel (*VSM*) pour chaque segment (phrases, paragraphes, *etc.*) dans le document suspect. Ensuite, la détection de plagiat est effectuée par calcul de la distance cosinus [SB88] entre ces *VSM*s.

3.3.3 Résultats de l'évaluation

La campagne d'évaluation AraPlagDet 2015 [BBR+15] a permis de comparer trois systèmes de détection extrinsèque et une méthode référence (*Baseline*) sur le corpus *Cr-ExAra* en utilisant le même protocole d'évaluation. La méthode référence (*Baseline*) proposée par les organisateurs Bensalem et al. [BBR+15] consiste à considérer le chevauchement des 5-grammes de mot entre les documents suspects et sources. Le document suspect est considéré comme plagié si le chevauchement entre les 5-grammes de mot des deux documents est supérieur à un seuil prédéterminé. Le tableau 3.8 résume les performances des différents systèmes participant à *AraPlagDet* 2015.

Système	Rappel	Précision	Granularité	Plagdet
Magooda	**0.831**	0.852	1.069	**80.2**
Palkovskii	0.542	**0.997**	**1.062**	62.7
Baseline	0.535	0.990	1.209	60.8
Alzahrani	0.530	0.831	1.186	57.4

Table 3.8: Résultats de la campagne d'évaluation *AraPlagDet* 2015 sur ExtAraDet

3.4 Discussion

Nous avons présenté dans cette section les différents systèmes de la littérature autour de la détection de plagiat dans les documents en langue arabe. Nous avons ainsi cherché à illustrer les techniques utilisées ainsi que les différents types de plagiat détectable

par ces dernières. Le tableau 3.9 présente une synthèse des systèmes de détection du plagiat extrinsèques décrits ci-dessus en fonction des critères: la technique utilisée, le niveau de comparaison et leur efficacité dans la détection de différents types de plagiat. D'après le tableau 3.9, on constate qu'il y a un manque de systèmes qui traitent les problèmes de détection du plagiat avec reformulation. Par ailleurs, on remarque aussi que la détection de plagiat avec *word embeddings* n'est pas encore étudiée pour l'arabe.

		Systèmes					
		FS-APD [AS09]	Aplag [Men12]	Iqtebas [JE12]	Hussein [Hus15]	RDI-RED [MMR15a]	Alzahrani [Alz15]
Technique	Empreinte Digitale		×	×			×
	Ensembles flous	×					
	SVD				×		
	LSA				×		
	Moteur de recherche					×	
	Ressources linguistiques		×		×		
	Word Embedding						
Niveau de comparaison	Phrase	×	×	×	×	×	×
	Paragraphe		×			×	
	Document		×				
Type de plagiat	Copier/Coller	×	×	×	×	×	×
	Réordonner	×	×	×	×	×	×
	Substitution par des synonymes		×		×		
	Paraphraser						

Table 3.9: Systèmes de détection du plagiat extrinsèque pour les documents arabe [NKC18]

3.5 Conclusion

Dans ce chapitre, nous avons mis en évidence l'ensemble des caractéristiques de la langue arabe comme l'ambiguïté, l'agglutination, la morphologie dérivationnelle l'absence de voyelles, *etc.* Ces caractéristiques créent un ensemble de problèmes et

de difficultés qu'il faut prendre en considération lors de la conception d'un système de traitement automatique de l'arabe.

Par la suite, nous avons passé en revue les différents travaux de la littérature ayant traité le problème de détection du plagiat dans la langue arabe. Nous avons présenté également la seule campagne d'évaluation de détection de plagiat dans les documents arabes *AraPlagDet 2015*, ainsi que les différents corpus proposés durant cette compétition.

Dans le chapitre suivant, nous décrivons nos approches de détection du plagiat dans les documents arabes. Le but principal de ces approches est de fournir un système capable de détecter les différentes formes de plagiat, à savoir le copier-coller, la restructuration des phrases, la reformulation, la substitution par des synonymes, *etc*.

" من بركة العلم أن تضيف الشيء إلى قائله "
جامع بيان العلم ، لابن عبد البر ، ٨٩/٢

4

Systèmes proposés

Rappelons que l'un de nos objectifs vise à fournir un système de détection du plagiat capable de détecter les différentes formes de plagiat, à savoir le plagiat littéral (copier-coller, le changement d'ordre des mots) et le plagiat dissimulé (reformulation, la substitution par des synonymes ou la restructuration des phrases). Dans ce chapitre, nous exposons, dans un premier temps, une approche de détection de similarités sémantiques monolingue entre les phrases en arabe. Cette approche sera ensuite utilisée avec la traduction automatique pour mesurer la similarité translingue arabe-anglais. Nous décrivons ensuite deux systèmes de détection de plagiat pour les documents arabes. Notre premier système de détection est basé sur la technique d'empreinte digitale et les word embeddings afin de détecter les différentes formes de plagiat. Le deuxième système consiste à explorer l'apprentissage automatique pour classifier les textes en deux classes plagié ou non-plagié.

4.1 Similarité sémantique entre phrases en arabe

La détection de plagiat entre deux documents consiste à mesurer le degré de relation sémantique entre leurs unités textuelles (mots, phrases ou paragraphes). En effet, avant de développer nos systèmes de détection de plagiat, nous avons débuté par le développement d'une approche destinée à mesurer la similarité sémantique entre les phrases en arabe.

4.1.1 Représentation des phrases avec le modèle word embeddings

Soient deux phrases $S_1 = w_1, w_2, \ldots, w_i$ et $S_2 = w'_1, w'_2, \ldots, w'_j$, leurs vecteurs dans le modèle *word embeddings* étant respectivement $(v_1, v_2, \ldots, v_i)$ et $(v'_1, v'_2, \ldots, v'_j)$. Un moyen simple pour mesurer la similarité sémantique entre S_1 et S_2, consiste à effectuer la somme pondérée des vecteurs de leurs mots [NS17]. La pondération consiste à affecter un poids à chacun des mots en fonction de leur importance dans la phrase. Par ailleurs, nous avons proposé dans [NFS17a] d'utiliser quatre pondérations différentes durant cette somme : une pondération *unitaire* où tous les vecteurs ont le même poids, une pondération des mots avec leur *fréquence inverse en documents* (*Idf*), une pondération des mots dépendant de leur *étiquetage morpho-syntaxique* et une pondération *mixte*, combinaison des deux précédentes. La figure 4.1 donne un aperçu de notre approche de calcul de similarité entre deux phrases. Nous détaillons dans ce qui suit ces quatre fonctions de pondération.

4.1.2 Similarité sémantique avec pondération unitaire

Il s'agit de la méthode la plus naïve, celle généralement utilisée pour calculer la représentation vectorielle d'une phrase. Il s'agit de la somme des vecteurs des mots composants la phrase, comme le montre la formule (4.1):

$$V = \sum_{k=1}^{i} v_k \tag{4.1}$$

où v_k est le vecteur du $k^{\text{ième}}$ mot de la phrase.

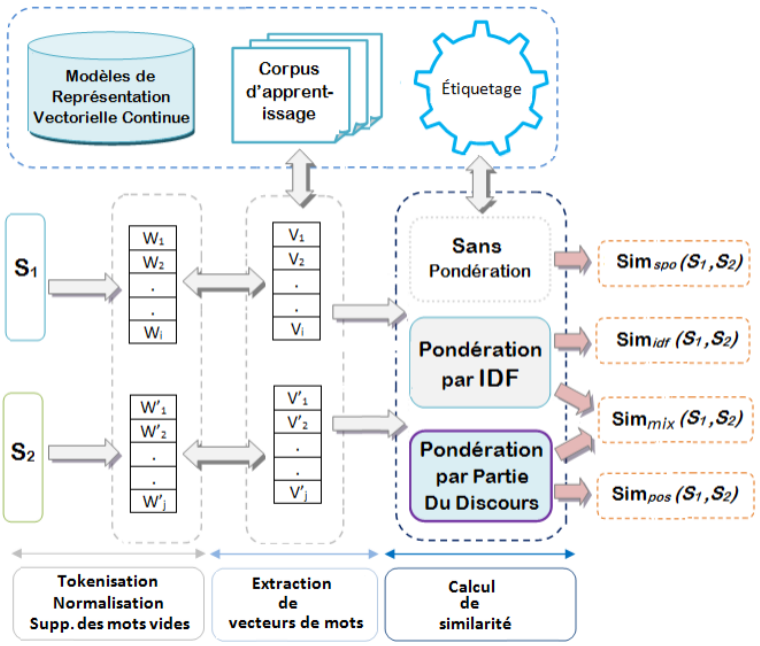


Figure 4.1: L'architecture de l'approche monolingue proposée [NFS17b]

Par exemple, soient les deux phrases: S_{src}=" ذهب يوسف إلى الكلية" (*Youssef est allé à la faculté*) et S_{sus}=" يوسف يمضى مسرعا للجامعة" (*Youssef va rapidement à l'université*). Le processus de mesure de similarité entre S_{src} et S_{sus} passe par deux étapes, comme illustré dans la figure4.2. Dans la première étape, on calcule les vecteurs de phrases V_{src} et V_{sus} par la formule (4.1). Ensuite, la similarité entre S_{src} et S_{sus} est obtenue en calculant la similarité cosinus entre V_{src} et V_{sus}, on aura donc: $Sim(S_{src}, S_{sus}) = Cos(V_{src}, V_{sus})$.

Nous avons présenté dans [NFS17a] qu'il y a des mots plus importants que d'autres dans une phrase. Ces mots ont plus de chance d'être conservés même lors d'une paraphrase ou d'une reformulation, méritant donc ainsi un poids plus important dans la représentation de cette phrase. On peut donc raisonnablement faire l'hypothèse que l'importance de ces mots peut être déterminée par leur rôle dans la phrase ou par leur rareté dans l'usage de leur langue.

4.1.3 Similarité sémantique avec pondération fréquentielle

La fréquence inverse en documents (*idf*) est une mesure statistique qui permet d'évaluer l'importance d'un mot dans un corpus [SB88]. C'est une mesure très util-

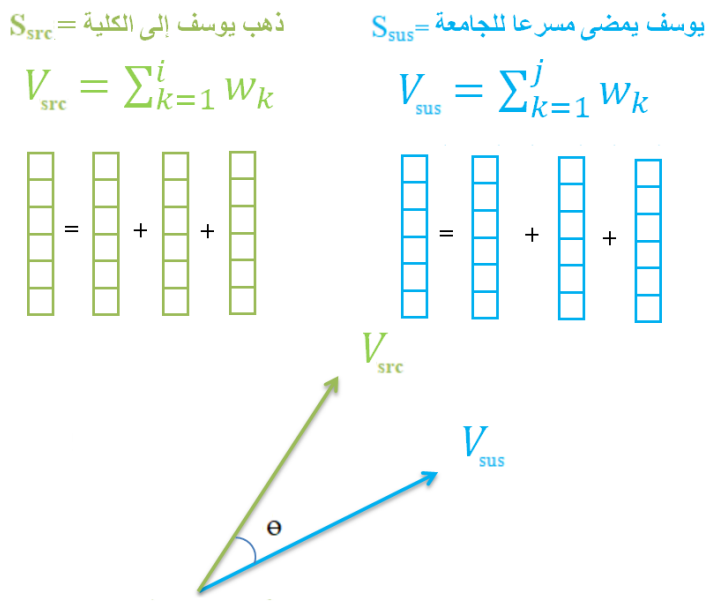


Figure 4.2: Mesure de la similarité entre deux phrases

isée dans le domaine de la recherche d'information [TP10]. L'idée sous-jacente est que les mots les moins fréquents sont considérés comme plus discriminants. Cette pondération donne ainsi un poids plus important aux mots moins fréquents. Ainsi, la représentation vectorielle d'une phrase est exprimée dans la formule (4.2):

$$V = \sum_{k=1}^{i} idf(w_k) * v_k \tag{4.2}$$

où v_k est le vecteur du $k^{\text{ième}}$ mot de la phrase et $idf(w)$ est la fonction qui donne la pondération *idf* du mot w_k. Elle est obtenue en appliquant la formule (4.3):

$$idf(w_k) = log(\frac{|D|}{|\{d_j : w_k \in d_j\}|}) \tag{4.3}$$

où $|D|$ représente le nombre total de documents dans le corpus et $|\{d_j : w_k \in d_j\}|$ le nombre de documents où le mot w_k apparaît. Les deux vecteurs de phrases construites sont ensuite comparées avec une similarité cosinus. La figure 4.3 illustre le fonctionnement de cette méthode sur les deux phrases de l'exemple précédent.

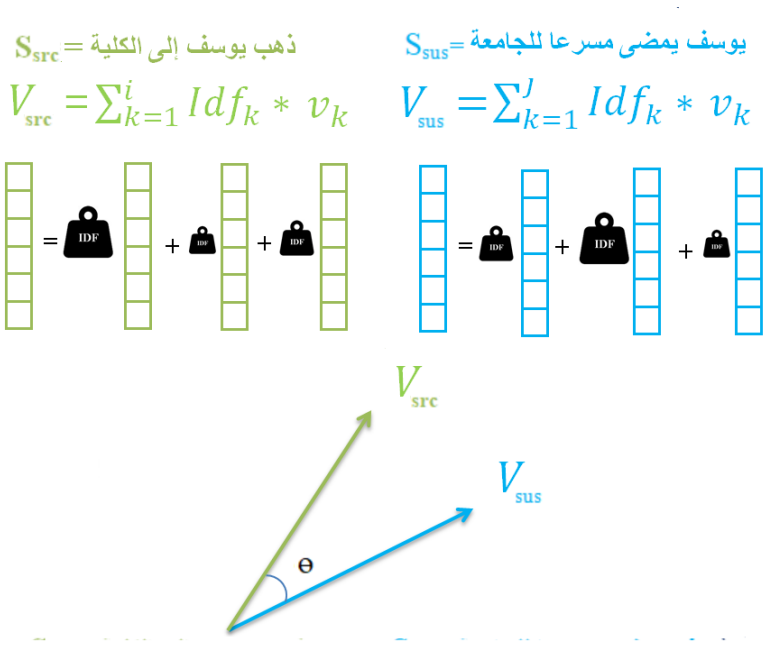


Figure 4.3: Mesure de la similarité sémantique entre deux phrases avec la pondération *idf*

4.1.4 Similarité sémantique avec pondération morphosyntaxique

Cette idée, inspirée de [Scho5] et plus récemment de [FABS17b], consiste à donner un poids au vecteur d'un mot en fonction de son étiquette morpho-syntaxique. Ainsi, la représentation vectorielle d'une phrase devient alors:

$$V = \sum_{k=1}^{i} Pos\_weight(Pos_{w_k}) * v_k \tag{4.4}$$

où v_k est le vecteur du $k^{\text{ième}}$ mot de la phrase et $Pos\_weight(Pos_w)$ est la fonction qui retourne le poids de la partie du discours du mot w. Le principe est d'étiqueter morpho-syntaxiquement les phrases en utilisant un analyseur morpho-syntaxique (p. ex. [GBBMLY12]), puis de calculer pour chacune des parties du discours sa fréquence inverse en documents et d'en déduire le poids à attribuer à chaque partie du discours. Si l'on considère le même exemple précédent, la similarité entre les deux phrases S_{src} et S_{sus} est obtenue comme illustré dans la figure 4.4.

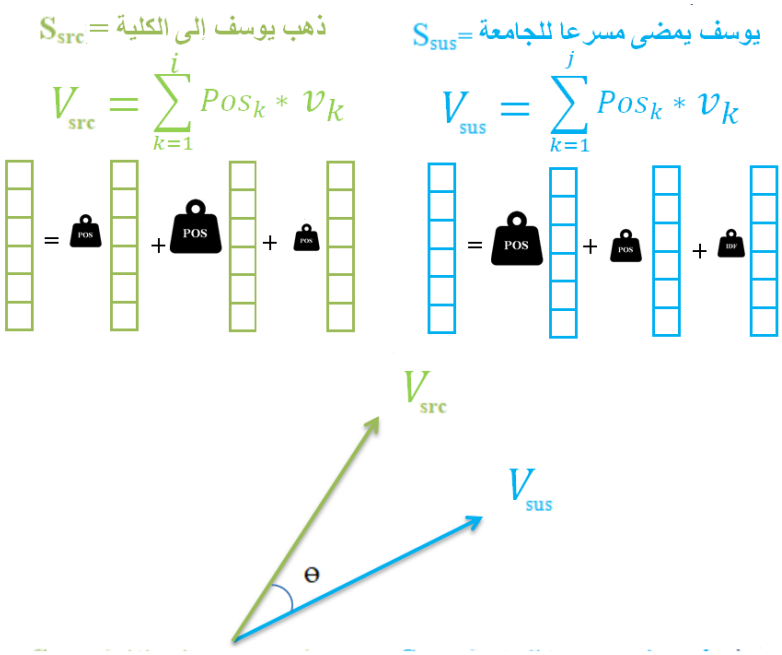


Figure 4.4: Mesure de la similarité sémantique avec la pondération morpho-syntaxique

4.1.5 Similarité sémantique avec pondération mixte

Cette dernière pondération utilise à la fois la pondération *idf* et la pondération en parties du discours. Ainsi, avec cette pondération, la représentation vectorielle d'une phrase devient alors:

$$V = \sum_{k=1}^{i} idf(w_k) * Pos\_weight(Pos_{w_k}) * v_k \tag{4.5}$$

où $Pos\_weight(Pos_w)$ retourne le poids de la partie du discours correspondant au mot w. La similarité entre les deux phrases avec cette méthode est montrée par la figure 4.5.

4.1.6 Expériences et résultats

Principe. Afin d'évaluer les performances de notre méthodes de pondération, nous avons utilisé quatre corpus (les trois corpus d'essai et le corpus de test) triés de la sous-tache: *détection de similarité textuelle sémantique sur des paires en langue arabe* de la campagne d'évaluation SemEval 2017 [CDA+17a]. Le Tableau 4.1 donne plus de détails sur les corpus utilisés.

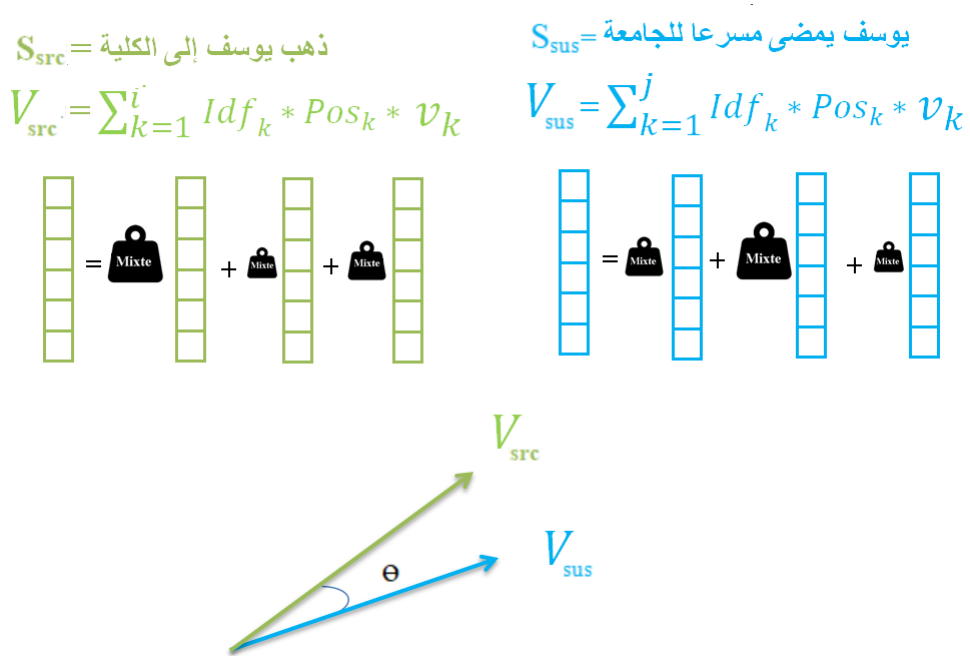


Figure 4.5: Mesure de similarité entre deux phrases avec une pondération mixte

Corpus	Nombre de paires
Microsoft Research Paraphrase Corpus (MSR-Paraphrase)	510 paires
Microsoft Research Video Description Corpus (MSR-Video)	368 paires
WMT2008 Development Dataset (SMT-europarl)	203 paires
Corpus d'évaluation SemEval 2017	250 paires

Table 4.1: Corpus utilisés pour évaluer nos méthodes de pondération [NFS17b]

Dans chacun de ces corpus, cinq annotateurs ont donné (manuellement) un score entre 0 et 5 aux paires de phrases, un score de 5 indiquant que les deux phrases ont exactement le même sens et un score de 0 indiquant que le sens des deux phrases est totalement différent. Le score final de chaque paire est la moyenne des scores des cinq annotateurs.

Évaluation. Les performances de nos quatre méthodes de pondération ont donc été testées sur plus de 1330 paires de phrases. Pour cela, nous avons calculé la *corrélation de Pearson* [BCHC09] entre nos scores de similarité sémantique et les jugements humains. Les résultats sont présentés dans le Tableau 4.2.

Corpus	MSRpar	MSRvid	SMTeuroparl	STS 2017	Global
Pondération unitaire	0,6745	0,7233	0,6233	0,5957	0,6653
Pondération avec IDF	0,7432	0,7820	0,7110	0,7309	0,7467
Pondération avec POS	0,7446	0,7951	0,7317	0,7425	0,7562
Pondération mixte	**0,7523**	**0,8276**	**0,7460**	**0,7646**	**0,7745**
Baseline de SemEval 2017	/	/	/	0,6045	/
BIT (gagnant de SemEval 2017)	/	/	/	0,7543	/

Table 4.2: Résultats de nos quatre méthodes ainsi que ceux des organisateurs et des vainqueurs de SemEval 2017 [NFS17b]

Discussion. les résultats reportés dans le tableau 4.2 montrent que, la pondération unitaire est nettement moins performante que les autres sur chacun des corpus considérés. La pondération par fréquence inverse en documents (IDF) et celle basée sur les parties du discours (POS) obtiennent des résultats comparables (respectivement $+8,14\%$ et $+9,09\%$ par rapport à la pondération unitaire), même si la seconde est légèrement meilleure. Enfin, la combinaison de ces deux pondérations donne les meilleurs résultats ($+10,92\%$ vis-à-vis de la pondération unitaire). En ce qui concerne la campagne d'évaluation SemEval 2017 [CDA+17b], nous avons participé à la sous-tâche *détection de similarité textuelle sémantique sur des paires en langue arabe*[1] avec la pondération *IDF* et *POS* (sans tenir compte conjointement des deux pondérations), ainsi, la méthode *POS* est classée deuxième sur 49 systèmes participants.

En effet, la pondération mixte (*IDF* et *POS*) serait arrivée en tête avec une corrélation de 0, 7646 contre 0, 7543 pour l'équipe gagnante dans cette compétition, BIT [WHJ+17b]. Il est également important de noter que le système BIT [WHJ+17b] utilise intensivement la base de données lexicales *Arabic WordNet* [BER+06] alors que notre approche n'exploite pas de telles données et pourrait être utilisée avec des langues qui n'en disposent donc pas.

4.1.7 Similarités sémantiques translingue entre les phrases arabe-anglais

Nous avons proposé aussi une approche pour mesurer la similarité translingue entre les phrases arabes-anglaises [NFSC17]. Cette approche exploite la traduction automatique, les *word embeddings*, l'alignement des mots et la pondération des vecteurs des mots.

[1] http://alt.qcri.org/semeval2017/task1/index.php?id=data-and-tools (consulté le 12/05/2018 à 14h)

Dans cette méthode on commence par la traduction automatique des phrases anglaises vers l'arabe (*langue pivot cf. section 2.4.5*). Dans ce sens, nous avons exploité *l'API Google Translate*[2] pour traduire ces phrases. Par la suite, une comparaison monolingue est effectuée en utilisant nos méthodes présentées ci-dessus (*cf. section 4.1.1, 4.1.3, 4.1.4 et 4.1.5*).

4.2 2L-APD : Un système de détection du plagiat à deux niveaux

L'idée de base de notre premier système de détection du plagiat *2L-APD* (de l'anglais *A Two-Level Arabic Plagiarism Detection System*) [NKC18] repose sur la décomposition du problème de détection du plagiat en deux sous-problèmes : détection du plagiat simple dit littéral, puis dans un second volet traiter le plagiat de type plus complexe. En effet, ce système propose la détection de plagiat extrinsèque sur deux niveaux : lexical et sémantique. En effet, le système *2L-APD* se compose de deux modules : un module de détection du plagiat utilisant les empreintes digitales (*cf. section 2.3.1*) et un module de détection basé sur les word embeddings (*cf. section 2.3.4*).

Le module d'Empreintes Digitales (ED_{mod}) est conçu pour détecter le plagiat littéral (*i.e.* niveau lexical), tel que le copier-coller, le changement d'ordres des mots et l'ajout de mots vides. Cependant, dans les cas réels de plagiat, plusieurs autres formes dissimulées de plagiat sont utilisées, y compris la reformulation, la substitution par des synonymes ou la restructuration des phrases. Ces techniques d'obfuscations génèrent souvent un changement important dans la structure du texte original, ce qui peut affecter considérablement l'empreinte digitale du document. Ce fait rend le module ED_{mod} très faible devant les modifications textuelles. Afin de faire face à ce problème, nous avons proposé un deuxième module principalement basé sur la technique des Word Embeddings (WE_{mod}). Le but principal de WE_{mod} est de mesurer la similarité sémantique entre deux passages de textes.

Le problème de détection de plagiat prend en entrée une collection de documents $D = \{d_1, d_2, ..., d_i\}$ et un document suspect d_{sus}. La tâche principale d'un système de détection du plagiat consiste à localiser les paires de passages suspects et sources (p_{sus},p_{src}) similaires à partir de d_{sus} et d_{src} ($d_{src} \in D$). Ces passages pourraient avoir

[2] https://cloud.google.com/translate/ (consulté le 13/05/2018 à 11h)

plusieurs niveaux de similarité, tels que p_{sus} est exactement similaire à p_{src} ou p_{sus} et p_{src} sont sémantiquement similaires, *i.e.* p_{sus} est obtenu à partir de p_{src} par reformulation, substitution par des synonymes ou restructuration. Le processus de détection du plagiat dans 2*L-APD* est comme suit : dans un premier temps p_{sus} est scanné par le premier module ED_{mod}, si aucun plagiat n'est détecté, alors, p_{sus} est envoyé au deuxième module WE_{mod} pour détecter le plagiat dissimulé s'il existe. La figure 4.6 illustre une vue d'ensemble sur le système 2*L-APD*. Avant de passer à la description des deux modèles ED_{mod} et WD_{mod} il est nécessaire de présenter la phase de segmentation et de pré-traitement appliquée sur tous les documents à analyser par le système 2*L-APD*.

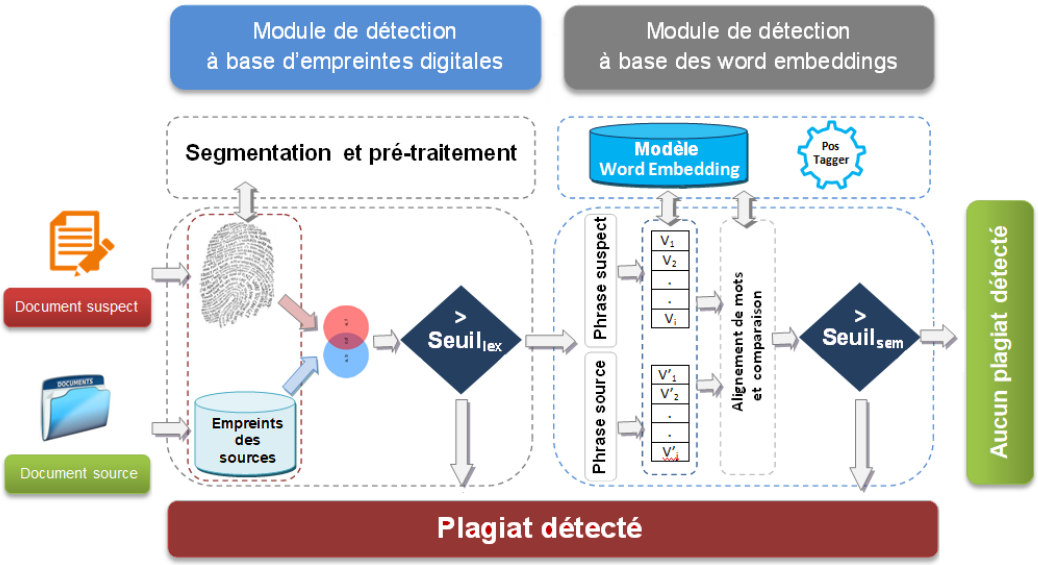


Figure 4.6: Architecture du système 2*L-APD*

4.2.1 Segmentation et pré-traitement

La stratégie de segmentation est une étape fondamentale dans la construction d'un système de détection du plagiat (*cf. section* 2.3.1.2). Cette étape définit la taille du segment (passage) utilisé par le système. La stratégie de segmentation intuitive consiste à utiliser une taille de segment fixe, par exemple, un nombre fixe de caractères ou de mots. En pratique, la segmentation structurelle est fréquemment adoptée; par exemple, le document est décomposé en phrases, paragraphes ou chapitres [HZ03].

Le système 2*L-APD* segmente les documents d_{sus} et d_{src} en phrases. Il faut bien noter

que, en pratique dans les documents arabes, la taille moyenne de la phrase est largement supérieure à celle des autres langues, elle contient environ 35 mots par phrase[3] [MLP06]. Par conséquent, nous avons choisi d'utiliser les signes de ponctuation (.), (,), (;), (:), (!) et (?) comme points de segmentation, à condition que la longueur de la phrase soit inférieure à 35 mots. Ensuite, un ensemble d'étapes de pré-traitement sont appliquées afin de normaliser les phrases pour les modules de détection :

(i) Tokenisation: décomposer chaque phrase en un ensemble de mots.

(ii) Suppression de mots vides.

(iii) Suppression des signes de ponctuation, diacritiques et les caractères non-alphanumériques.

(iv) Lemmatisation: l'outil *MADAMIRA* [PABD+14] est utilisé uniquement pour le module ED_{mod} afin de réduire les mots à leurs lemmes. Le module WD_{mod} utilise la forme normale des mots afin de préserver leurs propriétés sémantiques.

Dans ce qui suit, nous détaillons le processus de détection du plagiat entre deux documents d_{sus} et d_{src} par les deux modules ED_{mod} et WD_{mod}. Nous présentons ainsi un exemple de détection pour chacun d'eux.

4.2.2 Module de détection avec empreintes digitales

La détection du plagiat entre les phrases du document suspect (d_{sus}) et source (d_{src}) dans le module de détection à base d'empreinte digitale est effectuée comme suit :

1. **Création des empreintes digitales.** Dans cette étape une empreinte digitale est générée pour chaque phrase dans d_{sus} et d_{src} en itérant le processus suivant :

   (a) **Chunking.** Chaque phrase est représentée par un ensemble de n-grammes de caractères (*chunks*).

   (b) **Sélection.** Dans le but de sélectionner un sous-ensemble de *chunks*, nous avons considéré une nouvelle stratégie de sélection basée sur notre travail [BKC16]. L'idée clé de [BKC16] est d'exploiter la distribution hétérogène

[3] Ceci est dû principalement à l'absence de la ponctuation

des n-grammes dans un texte. En effet, nous avons montré que les n-grammes moins fréquents sont les plus significatifs dans un texte. Dans ce sens, nous proposons de sélectionner uniquement les n-grammes (*chunks*) ayant une fréquence inférieure à un seuil de sélection (T_{selec}) donné.

2. **Hachage.** La fonction de hachage de *Brian Kernighan et Dennis Ritchie (BKDR)* [RTR06] est appliquée aux n-grammes sélectionnés pour générer une empreinte digitale unique pour chaque phrase.

Détection de plagiat. Mesurer la similarité entre deux documents d_{sus} et d_{src} est effectuée en calculant le coefficient de Jaccard [Jac12] entre leurs empreintes digitales de phrases. Ensuite, le taux de similarité est comparé à un seuil fixe (T_{lex}: Seuil lexical) pour juger de l'existence d'un texte partagé et suggérer un éventuel plagiat. Si la similarité est inférieure à T_{lex}, alors la phrase suspecte est envoyée au module WD_{mod} pour détecter un éventuel plagiat dissimulé. La figure 4.7 montre un exemple de détection du plagiat entre deux textes arabes avec le module ED_{mod}.

Texte source		**Texte suspect**	
ذَهَبَ يُوسفُ إلى الكُلية لِلحُضُور إلى النَدْوة الْعِلمية		يمضي يوسف مسرعا للجامعة من أجل الحضور للدورة العلمية	
Segmentation			
S_2: لِلحُضُور إلى النَدْوة العِلمية	S_1: ذَهَبَ يُوسفُ إلى الكُلية ،	S'_2: من أجل الحضور للدورة العلمية	S'_1: يمضي يوسف مسرعا للجامعة
Prétraitement: Tokenisation, Supprimer les signes diacritiques, les non-lettres et les mots-vides			
للحضور الندوة العلمية	ذهب يوسف الكلية	الحضور للدورة العلمية	يمضي يوسف مسرعا للجامعة
Lemmatisation			
حضور ندوة علمي	ذهب يوسف كلية	حضور دورة علمي	مضى يوسف مسرع جامعة
Extraction des N-grammes			
حضوضور ورن رند ندو دوة ةعل علم لمي	ذهب هبي بيو يوس وسف سفك كلي لية	حضوضور ورد ردو دور ورة ةعل علم لمي	سرع مضى بيو يوس وسف سفم مسر رعج عجا جام امع معة
Sélection des N-grammes			
ضور دوة ةعل لمي	هبي يوس سفك كلي	ضور ردو ورة ةعل لمي	يوس سفم رعج جام معة
Hachage avec la fonction BKDR			
258 101 211 189 : $f(S_2)$	160 236 203 209 : $f(S_1)$	258 101 136 293189: $f(S'_2)$	256 87 741 781 166 : $f(S'_1)$
Jaccard (S'_j, S'_j) > S_{lex}= 15% ?			
Jaccard $(f(S_1), f(S'_1)) < T_{lex}$	Envoyer les phrases au module WD_{mod} (niveau sémantique)		
Jaccard $(f(S_1), f(S'_2)) < T_{lex}$	Envoyer les phrases au module WD_{mod}		
Jaccard $(f(S_2), f(S'_1)) < T_{lex}$	Envoyer les phrases au module WD_{mod}		
Jaccard $(f(S_2), f(S'_2)) > T_{lex}$	**Un plagiat potentiel est détecté entre S_2 et S'_2**		

Figure 4.7: Illustration de détection de plagiat par le module ED_{mod}

4.2.3 Module de détection avec word embeddings

La détection de plagiat dans le module Word Embeddings (WE_{mod}) est menée en mesurant la similarité sémantique entre phrases. Soient S_{src} et S_{sus} deux phrases source

et suspect respectivement, tel que : $S_{src} = w_1, w_2, \ldots, w_i$ et $S_{sus} = w'_1, w'_2, \ldots, w'_j$. Leurs vecteurs dans le modèle word embeddings étant respectivement $(v_1, v_2, \ldots, v_i)$ et $(v'_1, v'_2, \ldots, v'_j)$. Comme nous avons vu dans la *section 4.1*, la similarité sémantique entre phrases est obtenue en effectuons la somme des vecteurs de leurs mots [NS17]. Par exemple, soit les deux phrases: S_{src}="ذهب يوسف إلى الكلية" (*Joseph est allé à la faculté*) et S_{sus}="يوسف يمضى مسرعا للجامعة" (*Joseph va rapidement à l'université*). La similarité entre S_{src} et S_{sus} est obtenue comme suit :

1. **Calculer les vecteurs de phrases.** Pour calculer les vecteurs représentant les phrases V_{src} et V_{sus} nous effectuons la somme des vecteurs de leurs mots :

$$V_{src} = v(\text{الكلية}) + v(\text{يوسف}) + v(\text{ذهب}) \quad \text{(le mot vide إلى est supprimé)}$$
$$V_{sus} = v(\text{للجامعة}) + v(\text{مسرعا}) + v(\text{يمضى}) + v(\text{يوسف})$$

2. **Mesure de similarité.** La similarité entre S_{src} et S_{sus} est obtenue en calculant la similarité cosinus entre V_{src} et V_{sus}: $Sim(S_{src}, S_{sus}) = Cos(V_{src}, V_{sus}) = 0.71$

Afin d'améliorer les résultats de similarité, nous avons proposé une nouvelle méthode d'alignement des mots basée sur les word embeddings [NFSC17]. Ainsi, les mots des deux phrases S_{src} et S_{sus} sont alignés en fonction de leurs similarités sémantiques dans le modèle word embeddings. Par ailleurs, nous avons observé que les mots n'ont pas la même importance dans le sens des phrases. Par conséquent, nous avons utilisé notre fonction de pondération afin d'affecter à chaque mot (w_k) un poids [NFS17a] (*cf. section 4.1*). Cette fonction utilise à la fois une pondération avec fréquence inverse en documents (*idf*) et une pondération en parties du discours (*pos*). La similarité entre S_{src} et S_{sus} est mesurée par la formule (4.6) :

$$Sim(S_{src}, S_{sus}) = \frac{1}{2}\left(\frac{\sum_{w_k \in S_{src}} WT(w_k) * BM(w_k, S_{sus})}{\sum_{w_i \in S_{src}} WT(w_i)} + \frac{\sum_{w_k \in S_{sus}} WT(w_k) * BM(w_k, S_{src})}{\sum_{w_j \in S_{sus}} WT(w_j)}\right) \tag{4.6}$$

où $WT(w_k)$ est la pondération mixte $(idf(w_k) * Pos(w_k))$ correspondant au mot w_k. $BM(w_k, S_x)$ est le meilleur score de similarité entre w_k et tous les autres mots de la phrase S_x. La fonction *BM* (*Best Match*) aligne les mots en fonction de leurs similarités sémantiques dans le modèle word embeddings (*voir la formule (4.7)*). Enfin, la similarité $Sim(S_{src}, S_{sus})$ est comparée à un deuxième seuil fixe (T_{sem}: Seuil Sémantique) pour

juger de l'existence d'un plagiat potentiel.

$$BM(w_k, S_x) = Max\{Cos(v_k, v_r),\ w_r \in S_x\} \tag{4.7}$$

Illustration. Si l'on poursuit avec le même exemple S_{src}= "ذهب يوسف إلى الكلية" (*Joseph est allé à la faculté*). et S_{sus}= "يوسف يمضى مسرعا للجامعة" (*Joseph va rapidement à l'université*). La similitude entre S_{src}, S_{sus} dans le module WE_{mod} est obtenue en quatre étapes comme suit :

1. **L' étiquetage morpho-syntaxique.** Cette étape consiste à étiqueter morpho-syntaxiquement les phrases S_{src} et S_{sus} en utilisant l'analyseur morpho-syntaxique de Gahbiche-Braham et al. [GBBMLY12]. Le résultat de cette étape est :

$$\begin{cases} Pos\_tag(S_{src}) = verbe\ nom\_prop\ nom \\ Pos\_tag(S_{sus}) = nom\_prop\ verbe\ adj\ nom \end{cases}$$

2. **Alignement des mots.** Ensuite, nous alignons les mots de S_{src} et S_{sus} deux-à-deux. Pour cela, nous calculons la similarité entre chaque mot dans S_{src} et le mot sémantiquement le plus proche dans S_{sus} en utilisant la formule (4.7), par exemple :

$$BM(\text{يمضي}, S_{src}) = Max\{Cos(v(\text{يمضي}), v_r),\ w_r \in S_{sus}\} = Cos(v(\text{يمضي}), v(\text{ذهب})) = 0.85$$

3. **Pondération des mots.** Afin de pondérer les mots alignés, nous avons utilisé nos méthodes de pondérations présentées dans la *section 4.1*. Ainsi, nous récupérons pour chaque mot w_k dans S_x son poids $idf(w_k)$ tel que $idf(w_k) = log(\frac{S}{WS})$, où S représente le nombre total de phrases dans le corpus et WS est le nombre de phrases contenant le mot w_k. De plus, pour chacune des parties du discours nous avons calculé sa fréquence inverse en documents et d'en déduire le poids à attribuer à chaque partie du discours. La pondération *pos* de chaque partie du discours pos_i est obtenue en appliquant la formule $idf(pos_i) = log(\frac{S}{PS})$, où PS est le nombre de phrases contenant la partie du discours pos_i.

4. **Détection de plagiat.** La similitude entre S_{src} et S_{sus} est obtenue en utilisant la formule (4.6), ce qui nous donne: $Sim(S_{src}, S_{sus}) = 0.82$. Ce score est ensuite comparé au seuil T_{sem}, s'il est supérieur, cela signifie que S_{src} et S_{sus} sont sémantiquement similaires, *i.e.* l'existence d'un plagiat potentiel.

4.2.4 Modèle word embeddings utilisé

Dans le but de mesurer la similarité sémantique entre les phrases en langue arabe au niveau du module word embeddings, nous avons adopté le modèle CBOW[4] pour représenter les mots arabes proposé par Zahran et al. [ZMM+15]. Pour apprendre ce modèle, ils ont utilisé une grande collection de différentes sources qui contient plus de 5, 8 milliards de mots. Le tableau 4.3 présente les corpus utilisés pour apprendre le modèle CBOW de Zahran et al. [ZMM+15]. Par ailleurs, il faut souligner que les performances du modèle CBOW dépendent essentiellement de la bonne définition des paramètres d'apprentissage. Les paramètres du modèle CBOW de [ZMM+15] sont reportés dans le tableau 4.4.

Corpus	**Sources**
Wikipedia Arabe	[Wik06]
Corpus Arabe de BBC et CNN	[SA10]
Corpus parallèle ouvert	[Tie12]
Corpus Arabase	[RZR13]
Corpus OSAC (Open Source Arabic Corpus)	[SA10]
Corpus MultiUN	[CE12]
Corpus AGC (Arabic Gigaword Corpus)	[RPM01]
Corpus de l'universitaire King Saud (Ksu corpus)	[ksu12]
Crpus Arabe de Meedan	[Mee12]
LDC Arabic newswire	[GW01]
Texte brut du Coran	[Qur07]
Corpus KDE4	[Tie09]
Corpus de Khaleej et Watan 2004	[Kha04]

Table 4.3: Corpus utilisés pour apprendre le modèle CBOW de Zahran et al. [ZMM+15]

Paramètres	**Valeurs**
Taille des vecteurs	300
Taille de la fenêtre de contexte	5
Seuil de sous-échantillonnage des mots fréquents	10^{-5}
Exemples négatifs d'apprentissage	10
Nombre de mots les moins fréquents à supprimer	100

Table 4.4: Paramètres d'apprentissage du modèle CBOW de Zahran et al. [ZMM+15]

[4] https://sites.google.com/site/mohazahran/data (consulté le 11/04/2018 à 09h)

4.2.5 Paramétrage

Avant de présenter nos résultats, nous devons mentionner que le seuil lexical (T_{lex}) et le seuil sémantique (T_{sem}) sont fixés empiriquement en utilisant le corpus d'entraînement de AraPlagDet 2015 [BBR+15]. La description de ce corpus est détaillée dans la *section 3.3.1*. Rappelons ici que, dans ce corpus, chaque document suspect est associé à un fichier XML qui localise la position exacte des passages plagiés. De plus, les documents suspects sont classés en quatre ensembles selon le type de plagiat utilisé : documents sans plagiat, documents avec duplication (copier-coller), documents avec plagiat artificiel (mélange de phrases et de mots) et documents avec plagiat dissimulé (la substitution par des synonymes, l'ajout de diacritiques et le paraphrase).

En effet, nous avons utilisé les documents avec duplication et avec plagiat artificiel pour déterminer la valeur de seuil lexical (T_{lex}) et les documents avec plagiat dissimulés pour le seuil sémantique (T_{sem}). Ainsi, le seuil T_{lex} est fixé à 15%, ce qui signifie que deux empreintes digitales décrivant deux phrases différentes ont une intersection inférieure à 15%, et le seuil T_{sem} est fixé à 60% pour indiquer l'existence d'un plagiat dissimulé potentiel. En ce qui concerne la taille de n-gramme dans le module ED_{mod}, nous avons choisi la taille $n = 3$, en se basant sur notre étude empirique [BKC16].

Système	Précision	Rappel	Granularité	Plagdet
ED(F)	0.7315	0,4347	**1,055**	0,5255
ED(M)	0,7713	0,6251	1,058	0,6631
ED(C)	0,7856	0,6383	1,059	0,6882
ED(F)+WE	0,7521	0,6623	1,057	0,6769
ED(M)+WE	**0,8593**	0,8781	1,064	**0,8308**
ED(C)+WE	0,8413	**0,8867**	1,068	0,8236

Table 4.5: Performance du 2L-APD sur le corpus ExAra-2015

4.2.6 Tests et résultats

Dans le but de tester et de prouver l'efficacité de notre système *2L-APD* et comparer ses performances par rapport à d'autres systèmes sur le même jeu de données, nous avons utilisé le corpus de test *Cr-ExtAra*[5] de la campagne d'évaluation de détection du plagiat en langue arabe AraPlagDet 2015 (*cf. section 3.3.1*) [BBR+15].

Plusieurs variantes du système *2L-APD* ont été testées pour mesurer l'impact de la résolution des empreintes digitales (*i.e. la stratégie de sélection*) et le module word

[5] http://misc-umc.org/AraPlagDet/datasets (consulté le 20/12/2017 à 16h)

embeddings sur le score de détection de plagiat. Ainsi, nous avons utilisé trois résolutions pour les empreintes du module ED_{mod}: *Fine (F), Médium (M) et Grossière (G)* avec 10%, 20% et 50% des 3-grammes sont sélectionnés respectivement. Nous avons également testé le système *2L-APD* sans et avec le module WE_{mod}. Concernant les métriques d'évaluation utilisées pour ces expérimentation, nous avons utilisé les mêmes que celles présentées dans la *section 2.5*. Le tableau 4.5 présente le taux de *précision, rappel, granularité et plagdet* pour les différentes résolutions d'empreintes digitales : *Fine (F), Médium (M) et Grossière (G)* avec ou sans le module WE_{mod}.

4.2.7 Discussion

D'après les résultats reportés au tableau 4.5, on observe que lorsque la résolution des empreintes est *Fine*, la *précision* est raisonnable avec 73% des cas détectés sont corrects, mais le rappel est très faible et égal à environ 43% et aussi un *plagdet* faible d'environ 52%.

En appliquant la résolution *Médium*, la *précision* de détection augmente légèrement à 77%, cependant, le *rappel* et le *plagdet* sont considérablement améliorés à 62% et 66%, respectivement. Ceci est dû à l'augmentation du nombre des n-grammes sélectionnés dans l'empreinte digitale, c'est-à-dire que plus d'informations sont codées et utilisées comme un indicatif sur les passages de texte plagiés.

Pour la résolution *Grossière*, le taux d'augmentation n'est pas significatif par rapport à la résolution *Médium*. Cela signifie que la résolution *Médium* est capable de coder suffisamment d'information sur les passages de textes pour assurer la détection. Fait intéressant à noter, l'utilisation du modèle word embeddings améliore significativement le *plagdet* (avec un gain moyen de + 15,33%). Cela est dû à l'incapacité du module d'empreinte digitale à détecter le plagiat dissimulé.

4.3 ML-APD : Détection de plagiat arabe avec apprentissage automatique

Notre deuxième système de détection du plagiat repose sur l'apprentissage automatique. La détection de plagiat avec cette technique se déroule en deux phases : une phase d'entraînement et une phase de détection. La phase d'entraînement permet de construire des modèles de plagiat et de non-plagiat à partir d'un ensemble de données d'apprentissage. Cette phase d'apprentissage est alimentée par des couples de phrases (source (S_{src}), suspect (S_{sus})) caractérisées. Chaque couple de phrase est étiquetée par deux labels : *plagié* ou *non plagié*. Nous proposons trois types de caractéristiques au

défi de la détection du plagiat dans la langue arabe à savoir : lexical, syntaxique et sémantique. Concernant la génération des modèles, nous explorons trois types de classificateurs, à savoir machine à vecteurs de support (*SVM*), les arbres de décision (*DT*) et les forêts d'arbres décisionnels (*RF*).

4.3.1 Extraction des caractéristiques

Trois ensembles de caractéristiques sont déployés dans notre système: lexical, syntaxique et sémantique. Ces caractéristiques sont principalement issus des meilleurs systèmes participants à *SemEval 2017* [CDA+17b].

1. **Caractéristiques lexicales.** Les caractéristiques de chevauchement lexical sont couramment appliquées dans la détection paraphrase et la similarité textuelle sémantique [DS09]. Ces caractéristiques sont basées sur le chevauchement entre les mots, les n-grammes de caractères ou les n-grammes de mots de deux unités textuelles. Dans cette catégorie, nous utilisons les quatre caractéristiques suivantes :

   (a) **Chevauchement des sacs de mots :** le coefficient de Jaccard [Jac12] est utilisé pour mesurer la similarité lexicale entre les sacs de mots (BoW) des phrases comme illustre la formule (4.8).

   $$BoW_{ovr} = \frac{BoW(S_{src}) \cap BoW(S_{sus})}{BoW(S_{src}) \cup BoW(S_{sus})} \tag{4.8}$$

   (b) **Chevauchement des n-grammes de mots** : dans cette caractéristique le coefficient de Jaccard [Jac12] est aussi utilisé pour mesurer la similarité lexicale entre les 3-grammes de mots de S_{src} et S_{sus}. Elle est exprimée par la formule (4.9).

   $$W\_Ngram_{ovr} = \frac{3\text{-}gram(S_{src}) \cap 3\text{-}gram(S_{sus})}{3\text{-}gram(S_{src}) \cup 3\text{-}gram(S_{sus})} \tag{4.9}$$

   (c) **Chevauchement des n-grammes de caractères** : nous calculons le coefficient de Jaccard [Jac12] entre les 3-grammes de caractères de S_{src} et S_{sus}.

   (d) **Distance de Levenshtein**: appelée aussi distance d'édition [Lev66], cette distance mesure la différence entre deux chaînes de caractères. Ainsi, elle

mesure le nombre minimum de substitutions, d'insertions et de suppressions nécessaires pour transformer S_{sus} en S_{src}.

2. **Caractéristiques syntaxiques.** Dans cette catégorie, nous avons considéré quatre caractéristiques syntaxiques, il s'agit de détecter la ressemblance de style syntaxique entre S_{sus} et S_{src}.

   (a) **Chevauchement des parties du discours** : mesure le chevauchement entre les sacs des parties du discours (BoP) de S_{src} et S_{sus}. Elle est obtenue par:

   $$POS_{Ovr} = \frac{BoP(S_{src}) \cap BoP(S_{sus})}{BoP(S_{src}) \cup BoP(S_{sus})} \tag{4.10}$$

   (b) **Chevauchement des n-grammes des parties du discours** : nous avons calculé le chevauchement entre les 3-grammes des parties du discours comme suit :

   $$POS_{Gr\_Ovr} = \frac{3\text{-}gram(Pos(S_{src}) \cap 3\text{-}gram(Pos(S_{sus})}{3\text{-}gram(Pos(S_{src}) \cup 3\text{-}gram(Pos(S_{sus})} \tag{4.11}$$

   (c) **Plus longue sous-séquence commune dans les parties du discours :** cette caractéristique calcule la taille de la plus longue sous-séquence commune entre les deux suites des parties du discours de S_{sus} et S_{src}. La valeur de cette taille est ensuite normalisée relativement au nombre total des parties du discours dans chaque phrase. LCS_{pos} est obtenue par la formule (4.12).

   $$LCS_{pos}(S_{src}, S_{sus}) = \frac{LCS(Pos(S_{src}), Pos(S_{sus}))}{|Pos(S_{src})| + |Pos(S_{sus})|} \tag{4.12}$$

   (d) **Chevauchement de mots vides :** Stamatatos [Sta11] a montré que les mots vides représentent un bon indicateur pour la détection du plagiat. Ainsi, cette caractéristique mesure le taux de chevauchement entre les mots vides de S_{src} et S_{sus}.

3. **Caractéristiques sémantiques**. Comme le plus grand défi à relever par un système de détection de plagiat est comment mesurer la similarité sémantique entre les textes, alors, les caractéristiques sémantiques sont requises pour un tel système. Ainsi, le résultat du module WE_{mod} de notre premier système *2L-APD* est aussi utilisé comme une caractéristique sémantique pour générer nos modèles supervisés.

4.3.2 Construction des modèles d'apprentissage

Afin de construire nos modèles, nous avons généré un ensemble de couples de phrases (S_{src}, S_{sus}) à partir du corpus d'entraînement *Cr-EnExtAra*[6] de la campagne d'évaluation *AraPlagDet* 2015 [BBR+15]. Plus de détails sur ce corpus apprentissage sont reportés dans le tableau4.6.

Nbr total de couples de phrases		57480	100%
Couples de phrases sans plagiat		29487	51,3%
Couples de phrases avec plagiat		27992	48,7%
	Sans obfuscations (C&P)	5654	20,2%
	Mélange de mots	3666	13,1%
	Substitution par des synonymes	3247	11,6%
	Paraphrase Manuel	1063	3,8%

Table 4.6: Corpus d'apprentissage extrait de *Cr-EnExtAra* [BBR+15]

En effet, nous avons étiqueté ces couples de phrases par 1 ou 0 pour désigner un couple plagié ou non plagié, respectivement. Par la suite, les caractéristiques lexicales, syntaxiques et sémantiques décrites dans la section précédente sont extraites à partir des couples (S_{src}, S_{sus}). Finalement, trois classificateurs sont entraînés sur ces caractéristiques, à savoir Machine à vecteurs de support (SVM), les arbres de décision (DT) et les forêts d'arbres décisionnels (RF).

La génération de nos modèles est réalisée avec la bibliothèque python *Scikit-learn*[7] [PVG+11]. De plus, il faut aussi noter que la mise en œuvre effective de notre approche de ML-SVM nécessite de fixer un certain nombre de paramètres nommés *hyper-paramètres* liés au classificateur SVM. Ainsi, nous avons utilisé la stratégie *Grid Search* [BdCBM13] pour l'optimisation du paramètre de régularisation (appelé C) entre le taux d'erreur et la complexité du modèle. Le deuxième paramètre estimé est le paramètre gamma (γ) qui correspond à la largeur de la fonction de base radiale (RBF) (de l'anglais *Radial Basis Function*).

4.3.3 Tests et résultats

Nous avons considéré le même corpus de test *Cr-ExtAraDet* [BBR+15] afin d'évaluer les performances de notre système. En effet, plusieurs variantes de *ML-APD* ont été testées sur ce corpus avec trois classificateurs (SVM, DT et RF) pour mesurer l'impact

[6] http://misc-umc.org/AraPlagDet/datasets (consulté le 21/01/2018 à 17h)
[7] http://scikit-learn.org (consulté le 13/05/2018 à 13h)

des caractéristiques lexicales, syntaxiques et sémantiques sur l'efficacité de la détection. Les performances en termes de *précision, rappel, granularité et plagdet* sont présentées dans le tableau 4.7.

Classificateur	Caractéristiques	Rappel	Précision	Granularité	Plagdet
SVM	Lex	0.7875	0.8524	1.057	0.7867
	Lex+Syn	0.8849	0.8671	**1,056**	0.8423
	Lex+Syn+Sem	**0.9215**	0.8863	1.059	**0.8671**
RF	Lex	0.7413	0.8139	1.059	0.7432
	Lex+Syn	0.7747	0.8387	1.065	0.7681
	Lex+Syn+Sem	0.8191	0.8552	1.0612	0.8014
DT	Lex	0.7425	0.8139	1.058	0.7465
	Lex+Syn	0.7789	0.8327	1.062	0.7709
	Lex+Syn+Sem	0.8513	**0.9107**	1.063	0.8423

Table 4.7: Les performances de détection du système *ML-APD*

4.3.4 Discussion

Les résultats rapportés dans le tableau 4.5 montrent que les classificateurs *SVM, DT* et *RF* avec toutes les caractéristiques réussissent à détecter les différentes formes plagiat avec un score de *plagdet* entre 80% et 87%. De plus, nous pouvons facilement observer que *SVM* améliore le *plagdet* par +6% et +2% par rapport aux deux autres variantes basées sur les classificateurs *RF* et *DT*, respectivement. Cependant, le classificateur *DT* avec toutes les caractéristiques atteint une meilleur *précision* avec plus de 91%. Par ailleurs, on observe que *SVM* avec toutes les caractéristiques a obtenu un meilleur *rappel* de 92%.

En ce qui concerne l'impact des caractéristiques extraites, on observe que toutes ces caractéristiques améliorent les résultats de la détection du plagiat. De plus, nous remarquons que les caractéristiques syntaxiques et sémantiques jouent un rôle clé dans l'amélioration des résultats de détection pour tous les classificateurs avec une moyenne de $+7,3\%$ dans le *plagdet*. Par ailleurs, on observe aussi que le classificateur *SVM* avec toutes les caractéristiques propose un meilleur score de détection par rapport à la meilleure variante $(ED(M)+WE)$ du premier système *2L-APD* avec un gain de $+3,65\%$ dans la métrique *plagdet* (*voir le tableau 4.8*).

Système	Rappel	Précision	Granularité	Plagdet
ML-SVM	0.921	0.886	1.059	**86.7**
ED(M)+WE	0.859	0.878	1.064	83.1

Table 4.8: *2L-APD* Vs. *ML-SVM*

4.4 Comparaison

Afin de comparer nos systèmes aux systèmes proposés récemment dans l'état de l'art, nous avons utilisé le corpus d'évaluation *Cr-ExtAra* de la campagne *AraPlagDet* 2015 [BBR+15]. Rappelons que, l'avantage principal de cette campagne est de permettre l'évaluation de plusieurs systèmes de détection sur le même corpus. Trois systèmes ont participé dans cette campagne d'évaluation: Magood et al. [MMR+15a], Alzahrani [Alz15] et Palkovskii[8] (*cf. section* 3.3). En effet, nous avons comparé la meilleure variante dans chacun de nos systèmes, à savoir *ED(M)+WE* et *ML-SVM* avec ces trois systèmes et la méthode *baseline* proposée par les organisateurs de la campagne Bensalem et al. [BBR+15]. Cette méthode est basée sur la détection de chevauchement des 5-grammes de mot entre les documents suspects et sources.

Le tableau 4.9 reporte plus de détails sur les performances en termes de *rappel*, *précision* et *granularité* de nos meilleures variantes, Magooda et al. [MMR+15a], Alzahrani [Alz15], Palkovskii[9] ainsi que la méthode référence *Baseline*. De plus, la figure 4.8 illustre les résultats comparatifs en terme de *plagdet*.

Système	Rappel	Précision	Granularité
ML-SVM	**0.921**	0.886	**1.059**
ED(M)+WE	0.859	0.878	1.064
Magooda	0.831	0.852	1.069
Palkovskii	0.542	**0.997**	1.062
Baseline	0.535	0.990	1.209
Alzahrani	0.530	0.831	1.186

Table 4.9: Résultats de comparaison

Le tableau 4.9 montre que nos deux méthodes *ED(M)+WE* et *ML-SVM* améliorent les résultats de la méthode *Baseline* en termes de *rappel*, *précision* et *granularité*. En

[8] http://plagiarism-detector.com/ (consulté le 21/01/2018 à 17h)
[9] http://plagiarism-detector.com/ (consulté le 21/01/2018 à 17h)

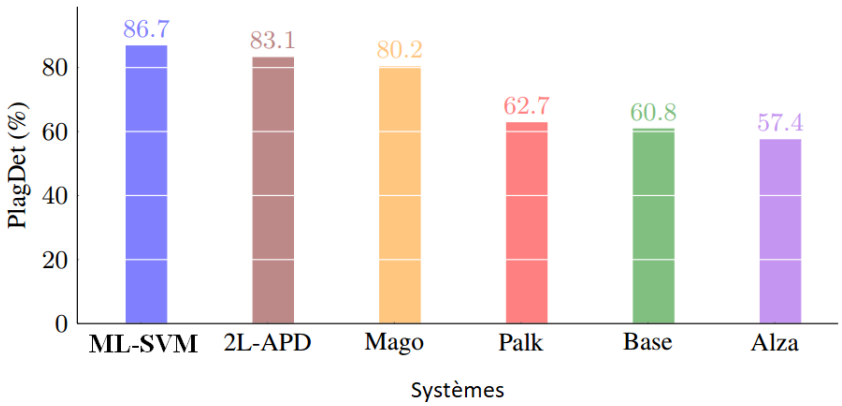


Figure 4.8: Résultats comparatifs en terme de plagdet
(Abv. Mag: Magooda, Alz: Alzahrani, Bas: Baseline, Pal: Palkovskii)

outre, on constate que la méthode ML-SVM surpasse le meilleur système de la campagne AraPlagDet 2015 Magooda et al. [MMR+15b] avec des gains de + 9% et de +3.4% pour le *rappel* et la *précision* respectivement. Par ailleurs, d'après la figure 4.8, on observe aussi que nos deux méthodes *ED(M)+WE* et *ML-SVM* proposent un meilleur *plagdet* par rapport au système de Magooda et al. [MMR+15b] avec un gain de +6.5% et de +2.9% respectivement.

4.5 Conclusion

Dans ce chapitre, nous avons proposé deux systèmes de détection plagiat extrinsèques pour les documents arabe traitant le plagiat dissimulé.

Le premier système nommé *2L-APD* propose la détection de plagiat en deux niveaux : lexical et sémantique. Dans le premier niveau la détection est effectuée en déployant la technique d'empreinte digitale, tandis que le deuxième exploite un modèle de représentation en word embeddings afin d'analyser les relations sémantiques entre les textes sources et suspects. Dans ce système, nous avons fait l'hypothèse que la sélection des n-grammes moins fréquents pour construire les empreintes digitales permet d'obtenir de meilleurs résultats de détection. Par ailleurs, nous avons aussi montré que l'alignement des mots et la pondération des vecteurs de mots améliorent le taux de détection.

Notre deuxième système *ML-APD* est basé sur l'apprentissage automatique supervisé. Afin de détecter le plagiat avec *ML-APD*, nous avons expérimenté trois classificateurs : machine à vecteurs de support (SVM), les arbres de décision (DT) et les forêts d'arbres décisionnels (RF), entraînés avec trois types de caractéristiques lexicales, syn-

taxiques et sémantiques. Nous avons observé que les caractéristiques syntaxiques et sémantiques augmentent significativement le taux de détection.

Les performances de détection de 2*L-APD* et de *ML-APD* sont évaluées sur le corpus *Cr-ExtAra* de la campagne d'évaluation AraPlagDet 2015 [BBR+15]. Ainsi, l'efficacité des deux systèmes a été prouvée dans la détection de plagiat littéral et dissimulé, y compris le copier-coller, la restructuration des phrases, la reformulation et la substitution par des synonymes. Nos résultats de détection sont comparés également à trois systèmes participants à AraPlagDet 2015. Notre meilleure approche *ML-SVM* obtient un *plagdet* de 86, 7% avec une amélioration de +6.5% par rapport au meilleur système de la campagne AraPlagDet 2015.

Conclusion générale

Les travaux présentés dans cette thèse portent sur la question de la détection du plagiat textuel dans les documents arabes. Nous avons commencé nos recherches par une étude des approches et systèmes existants. Ces études nous ont permis de constater que la détection de plagiat dissimulé est très peu traitée dans la littérature. Ainsi, Nous avons rapporté dans ce manuscrit nos approches explorées pour construire un système d'analyse textuelle destiné à repérer les passages de texte partagés entre les documents en arabes. Nos travaux se déclinent dans les trois contributions suivantes:

La première contribution concerne l'analyse de la similarité textuelle sémantique. En effet, nous avons combiné les techniques d'alignement des mots avec les word embeddings et la pondération mixte (fréquentielle et morphosyntaxique) des mots afin d'améliorer la détection de la similarité sémantique entres les phrases en langue arabe. Par ailleurs, nous avons participé avec cette technique à la sous tâche monolingue arabe-arabe de détection de similarité textuelle sémantique (STS) de SemEval 2017 [CDA+17a]. Cette technique a prouvé son efficacité en se classant deuxième parmi 49 systèmes proposés pour la compétition.

La deuxième contribution concerne l'utilisation des word embeddings dans le contexte de détection de plagiat dans les documents arabe. Ainsi, nous avons présenté notre système de détection de plagiat à deux niveaux pour l'arabe (2*L-APD*). Ce système est basé sur deux modules de détection. Le premier module utilise la technique d'empreinte digitale afin de détecter le plagiat littéral (niveau lexical), tel que le copier-coller et le changement dans la structures des phrases. Le deuxième module est conçu pour détecter le plagiat dissimulé (niveau sémantique), y compris la reformulation et la substitution par des synonymes en utilisant les word embeddings, l'alignement et la pondération des mots.

Comme troisième contribution, nous avons également étudié la détection de plagiat dans les documents arabe par apprentissage automatique supervisé (*ML-APD*). Dans la phase d'entraînement nous avons caractérisé les documents au niveau phrase. En effet, nous avons proposé trois types de caractéristiques : lexical, syntaxique et sémantique extraites à partir des couples de phrases sources et suspects. Concernant la génération des modèles nous avons exploré trois types de classificateurs : Machine à vecteurs de support (SVM), les arbres de décision (DT) ou les forêts d'arbres décisionnels (RF).

Finalement, afin de comparer nos approches à un état de l'art récent, nous avons utilisé le corpus d'évaluation de la campagne AraPlagDet 2015 [BBR+15]. L'avantage principal de cette campagne est de permettre l'évaluation de plusieurs systèmes de détection sur le même corpus. En effet, nous avons utilisé ce corpus afin de prouver l'efficacité de nos approches dans la détection de plagiat littéral y compris le copier-coller et la restructuration des phrases, *etc.* Nous avons également réussi à détecter plusieurs types de plagiats dissimulés comme la reformulation et la substitution par des synonymes. Notre meilleure approche (*ML-SVM*) obtient un *plagdet* de $86,7\%$ avec une amélioration de $+6.5\%$ par rapport au meilleur système de la campagne AraPlagDet 2015.

Perspectives

L'utilisation des approches proposées dans cette thèse ouvre le champ à plusieurs perspectives, parmi celles qui nous paraissent être les plus intéressantes :

- Nous envisageons d'améliorer notre module de détection par word embeddings avec la technique de désambiguïsation lexicale des mots afin de faire plus d'amélioration dans la détection de similarité sémantique [SVB+18];
- Nous souhaitons également tester notre système *ML-APD* avec d'autres caractéristiques et explorer d'autres méthodes d'apprentissage comme l'apprentissage profond (Deep Learning) en vue d'améliorer d'avantage les performances de détection [SAAM17];
- Enfin, nous prévoyons aussi d'aborder le problème de détection de plagiat translingue notamment pour l'arabe-anglais et l'arabe-français [Alz16].

Références

[ABC+15] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado , USA, 2015.

[AJ08] Per Ahlgren and Bo Jarneving. Bibliographic coupling, common abstract stems and clustering: A comparison of two document-document similarity approaches in the context of science mapping. *Scientometrics*, 76(2):273–290, 2008.

[Alz15] Salha Alzahrani. Arabic plagiarism detection using word correlation in n-grams with k-overlapping approach, working notes for panaraplagdet at fire 2015. In *FIRE Workshops*, pages 123–125, 2015.

[Alz16] Salha Alzahrani. Cross-language semantic similarity of arabic-english short phrases and sentences. *Journal of Computer Science*, 12(1):1–18, 2016.

[AS08] Salha Mohammed Alzahrani and Naomie Salim. Plagiarism detection in arabic scripts using fuzzy information retrieval. In *Student Conf. Res. Develop., Johor Bahru, Malaysia*, pages 281–285, 2008.

[AS09] Salha Mohammed Alzahrani and Naomie Salim. On the use of fuzzy information retrieval for gauging similarity of arabic documents. In *Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the*, pages 539–544. IEEE, 2009.

[AS10] Salha Alzahrani and Naomie Salim. Fuzzy semantic-based string similarity for extrinsic plagiarism detection. *Braschler and Harman*, pages 1–8, 2010.

[ASA06] Latifa Al-Sulaiti and Eric Steven Atwell. The design of a corpus of contemporary arabic. *International Journal of Corpus Linguistics*, 11(2):135–171, 2006.

[ASA12] Salha M Alzahrani, Naomie Salim, and Ajith Abraham. 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, 2012.

[ATA16] Zaid Alaa, Sabrina Tiun, and Mohammedhasan Abdulameer. Cross-language plagiarism of arabic-english documents using linear logistic regression. *Journal of Theoretical and Applied Information Technology*, 83, January 2016.

[Bak95] B. S. Baker. On finding duplication and near-duplication in large software systems. In *Proceedings of 2nd Working Conference on Reverse Engineering*, pages 86–95, July 1995.

[Bau68] Frédéric Baudry. *Grammaire comparée des langues classiques: contenant la théorie élémentaire de la formation des mots en sanscrit, en grec et en latin avec références aux langues germaniques*, volume 1. A. Durand et Pedone Lauriel, 1868.

[BBC+09] Chiara Basile, Dario Benedetto, Emanuele Caglioti, Giampaolo Cristadoro, and MD Esposti. A plagiarism detection procedure in three steps: Selection, matches and "squares". In *Proc. SEPLN*, pages 19–23, 2009.

[BBR+15] Imene Bensalem, Imene Boukhalfa, Paolo Rosso, Lahsen Abouenour, Kareem Darwish, and Salim Chikhi. Overview of the araplagdet pan@ fire2015 shared task on arabic plagiarism detection. In *FIRE Workshops*, pages 111–122, 2015.

[BC10] Alberto Barrón-Cedeño. On the mono-and cross-language detection of text reuse and plagiarism. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 914–914. ACM, 2010.

[BCGR13] Alberto Barrón-Cedeño, Parth Gupta, and Paolo Rosso. Methods for cross-language plagiarism detection. *Knowledge-Based Systems*, 50:211–217, 2013.

[BCHC09] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[BCR09] Alberto Barrón-Cedeño and Paolo Rosso. On automatic plagiarism detection based on n-grams comparison. In *European Conference on Information Retrieval*, pages 696–700. Springer, 2009.

[BCRAL10] Alberto Barrón-Cedeno, Paolo Rosso, Eneko Agirre, and Gorka Labaka. Plagiarism detection across distant language pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45. Association for Computational Linguistics, 2010.

[BCRPJ08] Alberto Barrón-Cedeno, Paolo Rosso, David Pinto, and Alfons Juan. On cross-lingual plagiarism analysis using a statistical model. In *PAN*, pages 1–10, 2008.

[BdCBM13] Igor Braga, Laís Pessine do Carmo, Caio César Benatti, and Maria Carolina Monard. A note on parameter selection for support vector machines. In *Mexican International Conference on Artificial Intelligence*, pages 233–244. Springer, 2013.

[BDGM95] Sergey Brin, James Davis, and Hector Garcia-Molina. Copy detection mechanisms for digital documents. In *ACM SIGMOD Record*, volume 24, pages 398–409. ACM, 1995.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[BE01] Bob S Brown and Dennis Emmett. Explaining variations in the level of academic dishonesty in studies of college students: Some new evidence. *College Student Journal*, 35(4), 2001.

[BER+06] William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. Introducing the arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300, 2006.

[BGD60] Regis Blachere and Maurice Gaudefroy-Demombynes. *Grammaire de l'arabe classique, morphologie et syntaxe / par R. Blachere et M. Gaudefroy-Demombynes*. Maisonneuve Paris, 3 ed., rev. et remaniee. edition, 1960.

[BGMZ97] Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 1997.

[BHR00] Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

[BHZ12] AS Bin-Habtoor and MA Zaher. A survey on plagiarism detection systems. *International Journal of Computer Theory and Engineering*, 4(2):185, 2012.

[BKC16] Nagoudi El Moatez Billah, Ahmed Khorsi, and Hadda Cherroun. Efficient inverted index with n-gram sampling for string matching in arabic documents. In *The 13th IEEE/ACS International Conference on Computer Systems and Applications*, Agadir, Morocco, 2016.

[BPS13] Steven Burrows, Martin Potthast, and Benno Stein. Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):43, 2013.

[Bra15] Chloé Braud. *Identification automatique des relations discursives implicites à partir de corpus annotés et de données brutes*. PhD thesis, Universite Paris Diderot-Paris VII, 2015.

[BRC13] Imene Bensalem, Paolo Rosso, and Salim Chikhi. A new corpus for the evaluation of arabic intrinsic plagiarism detection. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 53–58. Springer, 2013.

[BS08] Steven B Bird and Marco LA Sivilotti. Self-plagiarism, recycling fraud, and the intent to mislead. *Journal of Medical Toxicology*, 4(2):69–70, 2008.

[BSLL03] Jun-Peng Bao, Jun-Yi Shen, Xiao-Domg Liu, and Hai-Yan Liu. Quick asymmetric text similarity measures. In *Machine Learning and Cybernetics, 2003 International Conference on*, volume 1, pages 374–379. IEEE, 2003.

[BZ04] Yaniv Bernstein and Justin Zobel. A scalable system for identifying coderivative documents. In *International Symposium on String Processing and Information Retrieval*, pages 55–67. Springer, 2004.

[C+03] Paul Clough et al. Old and new challenges in automatic plagiarism detection. In *National Plagiarism Advisory Service, 2003; http://ir. shef. ac. uk/cloughie/index. html*. Citeseer, 2003.

[CDA+17a] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[CDA+17b] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[CE12] Yu Chen and Andreas Eisele. Multiun v2: Un documents with multilingual alignments. In *LREC*, pages 2500–2504, 2012.

[Ces08] Zdenek Ceska. Plagiarism detection based on singular value decomposition. In *Advances in natural language processing*, pages 108–119. Springer, 2008.

[CFGM02] Abdur Chowdhury, Ophir Frieder, David Grossman, and Mary Catherine McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems (TOIS)*, 20(2):171–191, 2002.

[CFL+04] Xin Chen, Brent Francia, Ming Li, Brian Mckinnon, and Amit Seker. Shared information and program plagiarism detection. *IEEE Transactions on Information Theory*, 50(7):1545–1551, 2004.

[Cha00] Achraf Chalabi. Mt-based transparent arabization of the internet tarjim. com. In *Conference of the Association for Machine Translation in the Americas*, pages 189–191. Springer, 2000.

[CL94] William I. Chang and Eugene L. Lawler. Sublinear approximate string matching and biological applications. *Algorithmica*, 12(4-5):327–344, 1994.

[CNP+12] Francisco Claude, Gonzalo Navarro, Hannu Peltola, Leena Salmela, and Jorma Tarhio. String matching with alphabet sampling. *Journal of Discrete Algorithms*, 11:37–50, 2012.

[CR03] Maxime Crochemore and Wojciech Rytter. *Jewels of stringology: text algorithms*. World Scientific, 2003.

[CS75] Václáv Chvatal and David Sankoff. Longest common subsequences of two random sequences. *Journal of Applied Probability*, 12(02):306–315, 1975.

[CW08] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[Dan13] Vera Danilova. Cross-language plagiarism detection methods. In *RANLP*, pages 51–57, 2013.

[DD03] Zhendong Dong and Qiang Dong. Hownet-a hybrid language and knowledge resource. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 820–824. IEEE, 2003.

[DDF+90] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[Dre07] Heinz Dreher. Automatic conceptual analysis for plagiarism detection. *Journal of Issues in Informing Science and Information Technology*, 4(2007):601–614, 2007.

[DRRA10] Sobha Lalitha Devi, Pattabhi RK Rao, Vijay Sundar Ram, and A Akilandeswari. External plagiarism detection. *Lab Report for PAN at CLEF*, 2010.

[DS09] Dipanjan Das and Noah A Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 468–476. Association for Computational Linguistics, 2009.

[E+05] Norzima Elbegbayan et al. Winnowing, a document fingerprinting algorithm. *TDDC03 Projects, Spring*, 2005.

[EHF+07] Mounir Errami, Justin M Hicks, Wayne Fisher, David Trusty, Jonathan D Wren, Tara C Long, and Harold R Garner. Déjà vu—a study of duplicate citations in medline. *Bioinformatics*, 24(2):243–249, 2007.

[ER90] Leo Egghe and Ronald Rousseau. *Introduction to informetrics: Quantitative methods in library, documentation and information science*. Elsevier Science Publishers, 1990.

[ESG+10] Mounir Errami, Zhaohui Sun, Angela C George, Tara C Long, Michael A Skinner, Jonathan D Wren, and Harold R Garner. Identifying duplicate content using statistically improbable phrases. *Bioinformatics*, 26(11):1453–1457, 2010.

[ESL+08] Mounir Errami, Zhaohui Sun, Tara C Long, Angela C George, and Harold R Garner. Deja vu: a database of highly similar citations in the scientific literature. *Nucleic acids research*, 37(suppl_1):D921–D924, 2008.

[Est09] Saleem Ahmed Saleem Estith. *Spatial Planning Of Health Services In Tulkarm City And Its Suburbs Using The Tool Of Geographic Information Systems GIS*. PhD thesis, 2009.

[FABS16] Jérémy Ferrero, Frédéric Agnes, Laurent Besacier, and Didier Schwab. A multilingual, multi-style and multi-granularity dataset for cross-language textual similarity detection. In *10th edition of the Language Resources and Evaluation Conference*, 2016.

[FABS17a] Jérémy Ferrero, Frederic Agnes, Laurent Besacier, and Didier Schwab. Compilig at semeval-2017 task 1: Cross-language plagiarism detection methods for semantic textual similarity. *arXiv preprint arXiv:1704.01346*, 2017.

[FABS17b] Jérémy Ferrero, Frédéric Agnes, Laurent Besacier, and Didier Schwab. Using word embedding for cross-language plagiarism detection. *arXiv preprint arXiv:1702.03082*, 2017.

[Fer17] Jérémy Ferrero. *Similarités Textuelles Sémantiques Translingues: vers la Détection Automatique du Plagiat par Traduction*. PhD thesis, LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES, 2017.

[Fis09] Teddi Fishman. "we know it when we see it" is not good enough: toward a standard definition of plagiarism that transcends theft, fraud, and copyright. 2009.

[FNBF17] Patricia I Fusch, Lawrence R Ness, Janet M Booker, and Gene E Fusch. The ethical implications of plagiarism and ghostwriting in an open society. *Journal of Social Change*, 9(1):4, 2017.

[Fre06] Heather Freegard. *Ethical practice for health professionals*. Thomson, 2006.

[FSGR13] Marc Franco-Salvador, Parth Gupta, and Paolo Rosso. Cross-language plagiarism detection using a multilingual semantic network. In *35th European Conference on Information Retrieval (ECIR'13)*, LNCS 7814, pages 710–713. Springer Berlin Heidelberg, 2013.

[FSL15] Jérémy Ferrero and Alain Simac-Lejeune. Détection automatique de reformulations-correspondance de concepts appliquée à la détection du plagiat. In *15e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances*, 2015.

[G+99] Ove Granstrand et al. The economics and management of intellectual property. *Books*, 1999.

[Gas49] Bachelard Gaston. La psychanalyse du feu. *Paris, Gallimard*, 1949.

[GB09] Bela Gipp and Jöran Beel. Citation proximity analysis (cpa): a new approach for identifying related work based on co-citation analysis. In *ISSI'09: 12th International Conference on Scientometrics and Informetrics*, pages 571–575, 2009.

[GB10] Bela Gipp and Jöran Beel. Citation based plagiarism detection: a new approach to identify plagiarized work language independently. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 273–274. ACM, 2010.

[GBBMLY12] Souhir Gahbiche-Braham, Hélene Bonneau-Maynard, Thomas Lavergne, and François Yvon. Joint segmentation and pos tagging for arabic using a crf-based classifier. In *LREC*, pages 2107–2113, 2012.

[GBCR12] Parth Gupta, Alberto Barrón-Cedeno, and Paolo Rosso. Cross-language high similarity search using a conceptual thesaurus. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 67–75. Springer, 2012.

[GBYS92] Gaston H. Gonnet, Ricardo A. Baeza-Yates, and Tim Snider. *Information Retrieval*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[Gib06] E Gibney. I'm no plagiarist, i moved a comma. *The Times Higher Education Supplement: THE*, (2104), 2006.

[Gip14] Bela Gipp. Citation-based plagiarism detection. In *Citation-based Plagiarism Detection*, pages 57–88. Springer, 2014.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'07)*, pages 1606–1611, Hyderabad, India, January 2007. Morgan Kaufmann Publishers Inc.

[GM11] Pascal Guibert and Christophe Michaut. Le plagiat étudiant. *Education et sociétés*, (2):149–163, 2011.

[Gon00] Ricardo Erkki S. Jorma T Gonzalo, N. Indexing methods for approximate string matching. *IEEE Data Engineering Bulletin*, 24, 2000.

[Gus97] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press, 1997.

[GW01] David Graff and Kevin Walker. Arabic newswire corpus (1-58563-190-6). In *Linguistic Data Consortium (LDC)*, 2001.

[H+96] Nevin Heintze et al. Scalable document fingerprinting. In *1996 USENIX workshop on electronic commerce*, volume 3, 1996.

[Hab10] Nizar Y Habash. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187, 2010.

[Ham15] Ahmed Hamdi. *Traitement automatique du dialecte tunisien à l'aide d'outils et de ressources de l'arabe standard: application à l'étiquetage morphosyntaxique*. PhD thesis, Aix-Marseille, 2015.

[Hat15] Ezz Hattab. Cross-language plagiarism detection method: Arabic vs. english. In *Developments of E-Systems Engineering (DeSE), 2015 International Conference on*, pages 141–144. IEEE, 2015.

[HKF+10] Shanmugasundaram Hariharan, Sirajudeen Kamal, Abdul Vadud Mohamed Faisal, Sheik Mohamed Azharudheen, and Bhaskaran Raman. Detecting plagiarism in text documents. In *Information Processing and Management*, pages 497–500. Springer, 2010.

[How99] Rebecca Moore Howard. *Standing in the shadow of giants: Plagiarists, authors, collaborators*. Number 2. Greenwood Publishing Group, 1999.

[HPB+03] André Happe, Bruno Pouliquen, Anita Burgun, Marc Cuggia, and Pierre Le Beux. Automatic concept extraction from spoken medical reports. *International Journal of Medical Informatics*, 70(2):255–263, 2003.

[Hus15] Ashraf S Hussein. A plagiarism detection system for arabic documents. In *Intelligent Systems' 2014*, pages 541–552. Springer, 2015.

[HZ03] Timothy C Hoad and Justin Zobel. Methods for identifying versioned and plagiarized documents. *Journal of the Association for Information Science and Technology*, 54(3):203–215, 2003.

[Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

[Jac86] Peter Jackson. Introduction to expert systems. 1986.

[JE12] Ameera Jadalla and Ashraf Elnagar. A plagiarism detection system for arabic text-based documents. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, pages 145–153. Springer, 2012.

[JMDL16] Killian Janod, Mohamed Morchid, Richard Dufour, and Georges Linarès. Réseaux de neurones pour la représentation des contextes continus des mots. In *CORIA-CIFED*, pages 656–668, 2016.

[KB10] Jan Kasprzak and Michal Brandejs. Improving the reliability of the plagiarism detection system. *Lab Report for PAN at CLEF*, pages 359–366, 2010.

[KCA04] Aleksander Kołcz, Abdur Chowdhury, and Joshua Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–610. ACM, 2004.

[KGH06] NamOh Kang, Alexander Gelbukh, and SangYong Han. Ppchecker: Plagiarism pattern checker in document copy detection. In *International Conference on Text, Speech and Dialogue*, pages 661–667. Springer, 2006.

[KGK01] Shereen Khoja, Roger Garside, and Gerry Knowles. *An Arabic tagset for the morphosyntactic tagging of Arabic*. PhD thesis, Lancaster University, 2001.

[Kha04] Khaleej. Khaleej and watan corpus, 2004.

[Kle10] Étienne Klein. L'instant présent unique, mais banal. *Pour la science*, (397):28–32, 2010.

[Kle14] E Klein. Le temps est-il affaire de conscience? *European Psychiatry*, 29(8):615, 2014.

[Kle16] Etienne Klein. *Le pays qu'habitait Albert Einstein*. Éditions Actes Sud, 2016.

[Koe05] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[Kou91] Djamel Eddine Kouloughli. *Lexique fondamental de l'arabe standard moderne*. Editions L'Harmattan, 1991.

[KR87] Richard M Karp and Michael O Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, 1987.

[KS10] Chow Kok Kent and Naomie Salim. Web based cross language plagiarism detection. In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on*, pages 199–204. IEEE, 2010.

[ksu12] ksucorpus. King saud university corpus, http://ksucorpus.ksu.edu.sa/ar/ (accessed january 20,2017), 2012.

[KT03] Dmitry V Khmelev and William J Teahan. A repetition based measure for verification of text collections and for text categorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 104–110. ACM, 2003.

[KWL03] Linda Achey Kidwell, Karen Wozniak, and Jeanne Phoenix Laurel. Student reports and faculty perceptions of academic dishonesty. *Teaching Business Ethics*, 7(3):205–214, 2003.

[Lar95a] Pierre Larcher. Où il est montré qu'en arabe classique la racine n'a pas de sens et qu'il n'y a pas de sens à dériver d'elle. *Arabica*, 42:291–314, 1995.

[Lar95b] Serge Larivée. La notion de plagiait scientifique. *Les Cahiers de Propriété Intellectuelle*, 8(1):159–190, 1995.

[Lar04] Birger Larsen. *References and citations in automatic indexing and retrieval systems-experiments with the boomerang effect*. PhD thesis, Department of Information Studies, Royal School of Library and Information Science, 2004.

[LBM04] Caroline Lyon, Ruth Barrett, and James Malcolm. A theoretical basis to the automated detection of copying between texts, and its practical implementation in the ferret plagiarism and collusion detector. *Plagiarism: Prevention, Practice and Policies*, 2004.

[LC11] Shengbo Liu and Chaomei Chen. The effects of co-citation proximity on co-citation analysis. In *Proceedings of ISSI*, pages 474–484, 2011.

[LEG+09] Tara C Long, Mounir Errami, Angela C George, Zhaohui Sun, and Harold R Garner. Responding to possible plagiarism. *Science*, 323(5919):1293–1294, 2009.

[Lev66] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[LL+11] Kenneth C Laudon, Jane Price Laudon, et al. *Essentials of management information systems*. Pearson Upper Saddle River, 2011.

[LMD01] Caroline Lyon, James Malcolm, and Bob Dickerson. Detecting short passages of similar text in large document collections. In *Proceedings of the*, 2001.

[M+94] Udi Manber et al. Finding similar files in a large file system. In *Usenix Winter*, volume 94, pages 1–10, 1994.

[MBC+05] Donald Metzler, Yaniv Bernstein, W Bruce Croft, Alistair Moffat, and Justin Zobel. Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 517–524. ACM, 2005.

[MBT06] Donald L McCabe, Kenneth D Butterfield, and Linda Klebe Trevino. Academic dishonesty in graduate business programs: Prevalence, causes, and proposed action. *Academy of Management Learning & Education*, 5(3):294–305, 2006.

[McC76] Edward M McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM (JACM)*, 23(2):262–272, 1976.

[McC05] Donald L McCabe. Cheating among college and university students: A north american perspective. *International Journal for Educational Integrity*, 1(1), 2005.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track*, pages 1301–3781, 2013.

[Mee12] Meedan. Meedan's open source arabic english, https://github.com/anastaw/meedan-memory, (accessed january 20,2017), 2012.

[Men12] Mohamed El Bachir Menai. Detection of plagiarism in arabic documents. *International journal of information technology and computer science (IJITCS)*, 4(10):80, 2012.

[MFHDC14] Vítor T Martins, Daniela Fonte, Pedro Rangel Henriques, and Daniela Da Cruz. Plagiarism detection: A tool survey and comparison. In *OASIcs-OpenAccess Series in Informatics*, volume 38. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[MH09] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009.

[MHG10] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in action: covers Apache Lucene 3.0*. Manning Publications Co., 2010.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Mil97] Olivier Millet. *Dictionnaire des citations*. Librairie générale française, 1997.

[MKB+10] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[MKZ06] Hermann A Maurer, Frank Kappe, and Bilal Zaka. Plagiarism-a survey. *J. UCS*, 12(8):1050–1084, 2006.

[MKZG10] Markus Muhr, Roman Kern, Mario Zechner, and Michael Granitzer. External and intrinsic plagiarism detection using a cross-lingual retrieval and segmentation system. In *Notebook Papers of CLEF 2010 LABs and Workshops*, 2010.

[MLP06] Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 216–220. Association for Computational Linguistics, 2006.

[MM93] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993.

[MM04] Paul Mcnamee and James Mayfield. Character n-gram tokenization for european language text retrieval. *Information retrieval*, 7(1):73–97, 2004.

[MMA01] Kasnawi Mahmoud Mohamed Abdulla. *Diriger la recherche scientifique à l'université pour répondre aux exigences du développement économique et social*. Symposium on Graduate Studies in Saudi Universities, Future Directions, King Abdulaziz University, Jedda, 2001.

[MMR+15a] Ahmed Magooda, Ashraf Y Mahgoub, Mohsen Rashwan, Magda B Fayek, and Hazem M Raafat. Rdi system for extrinsic plagiarism detection (rdi_red), working notes for panaraplagdet at fire 2015. In *FIRE Workshops*, pages 126–128, 2015.

[MMR+15b] Ashraf Y Mahgoub, Ahmed Magooda, Mohsen Rashwan, Magda B Fayek, and Hazem Raafat. Rdi system for intrinsic plagiarism detection (rdi_rid). 2015.

[MRS+08] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[MWSC15] Jean-Marc Marty, Guillaume Wenzek, Eglantine Schmitt, and Jocelyn Coulmance. Analyse d'opinions de tweets par réseaux de neurones convolutionnels. *Actes de DEFT, Caen, France: TALN*, 2015.

[MYZ13] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751, 2013.

[MZS00] Krisztián Monostori, Arkady Zaslavsky, and Heinz Schmidt. Parallel and distributed document overlap detection on the web. In *International Workshop on Applied Parallel Computing*, pages 206–214. Springer, 2000.

[Nas06] Susan Smith Nash. *Leadership and the e-learning organization*. texture press, 2006.

[NCA18] El Moatez Billah Nagoudi, Hadda Cherroun, and Ali Alshehri. Disguised plagiarism detection in arabic text documents. Alger, Algeria, 2018.

[NFS17a] El Moatez Billah Nagoudi, Jérémy Ferrero, and Didier Schwab. Limlig at semeval-2017 task1: Enhancing the semantic similarity for arabic sentences with vectors weighting. In *International Workshop on Semantic Evaluations*, pages 125–129, Vancouver, Canada, 2017.

[NFS17b] El Moatez Billah Nagoudi, Jérémy Ferrero, and Didier Schwab. Amélioration de la similarité sémantique vectorielle par méthodes non-supervisées. In *24e conférence sur le Traitement Automatique des Langues Naturelles (TALN 2017)*, Orléans, France, 2017.

[NFSC17] El Moatez Billah Nagoudi, Jérémy Ferrero, Didier Schwab, and Hadda Cherroun. Word embedding-based approaches for measuring semantic similarity of arabic-english sentences. In *International Conference on Arabic Language Processing*, pages 19–33, Fez, Morocco, 2017. Springer.

[NKC18] El Moatez Billah Nagoudi, Ahmed Khorsi, and Hadda Cherroun. A two-level plagiarism detection system for arabic documents. *Cybernetics and Information Technologies*, 20, 2018.

[NP12a] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[NP12b] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. In *Artificial Intelligence Proceedings*, volume 193, pages 217–250, 2012.

[NS17] El Moatez Billah Nagoudi and Didier Schwab. *Proceedings of the Third Arabic Natural Language Processing Workshop*, chapter Semantic Similarity of Arabic Sentences with Word Embeddings, pages 18–24. Association for Computational Linguistics, Valencia, Spain, 2017.

[NW70] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[OMK91] Yasushi Ogawa, Tetsuya Morita, and Kiyohiko Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy sets and systems*, 39(2):163–179, 1991.

[OSA12] Ahmed Hamza Osman, Naomie Salim, and Albaraa Abuobieda. Survey of text plagiarism detection. *Computer Engineering and Applications Journal (ComEngApp)*, 1(1):37–45, 2012.

[Oth01] Saad Eddine Othmani. *Les actions du Prophète la paix soit sur lui par l'imam*. Institut scientifique de la pensée islamique, 2001.

[OV13] Gabriel Oberreuter and Juan D VeláSquez. Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications*, 40(9):3756–3763, 2013.

[PABD+14] Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101, 2014.

[Par03] Chris Park. In other (people's) words: Plagiarism by university students–literature and lessons. *Assessment & evaluation in higher education*, 28(5):471–488, 2003.

[Pat12] Máté Pataki. A new approach for searching translated plagiarism. 2012.

[PBCSR11] Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. Cross-language plagiarism detection. *Language Resources and Evaluation*, 45(1):45–62, 2011.

[PCBC+09] David Pinto, Jorge Civera, Alberto Barrón-Cedeno, Alfons Juan, and Paolo Rosso. A statistical approach to crosslingual natural language tasks. *Journal of Algorithms*, 64(1):51–60, 2009.

[PN11] Maria Soledad Pera and Yiu-Kai Ng. Simpad: A word-similarity sentence-based plagiarism detection tool on web documents. *Web Intelligence and Agent Systems: An International Journal*, 9(1):27–41, 2011.

[Pou11] Fabien Poulard. *Détection de dérivation de texte*. PhD thesis, Université de Nantes, 2011.

[PSBCR10] Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pages 997–1005. Association for Computational Linguistics, 2010.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[Qur07] Quran. Raw quran text, 2007.

[R+05] David B Resnik et al. *The ethics of science: An introduction*. Routledge, 2005.

[RAY97] Patty Roberts, Janet Anderson, and Paula Yanish. Academic misconduct: Where do we start?. 1997.

[RKL88] Dennis M Ritchie, Brian W Kernighan, and Michael E Lesk. *The C programming language*. Prentice Hall Englewood Cliffs, 1988.

[RPM01] Ke Chen Junbo Kong Robert Parker, David Graff and Kazuaki Maeda. Arabic gigaword fifth edition (1-58563-595-2). In *Linguistic Data Consortium (LDC)*, 2001.

[RTR06] Anna Ritchie, Simone Teufel, and Stephen Robertson. How to find better index terms through citations. In *Proceedings of the workshop on how can computational linguistics improve information retrieval?*, pages 25–32. Association for Computational Linguistics, 2006.

[RZR13] Hazem M Raafat, Mohamed A Zahran, and Mohsen Rashwan. Arabase-a database combining different arabic resources with lexical and semantic information. In *KDIR/KMIS*, pages 233–240, 2013.

[SA10] Motaz K Saad and Wesam Ashour. Osac: Open source arabic corpora. In *6th ArchEng Int. Symposiums, EEECS*, volume 10, 2010.

[Saâ15] Houda Saâdane. *Le traitement automatique de l'arabe dialectalisé: aspects méthodologiques et algorithmiques*. PhD thesis, Grenoble Alpes, 2015.

[SAAM17] D. Suleiman, A. Awajan, and N. Al-Madi. Deep Learning Based Technique for Plagiarism Detection in Arabic Texts. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 216–222, Oct 2017.

[SALL93] Judith P Swazey, Melissa S Anderson, Karen Seashore Lewis, and Karen Seashore Louis. Ethical problems in academic research. *American Scientist*, 81(6):542–553, 1993.

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[SC08] Jangwon Seo and W Bruce Croft. Local text reuse detection. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 571–578. ACM, 2008.

[Sch05] Didier Schwab. *Approche hybride-lexicale et thématique-pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte*. PhD thesis, Université Montpellier II, 2005.

[SD87] NJ Scheers and C Mitchell Dayton. Improved estimation of academic cheating behavior using the randomized response technique. *Research in Higher Education*, 26(1):61–69, 1987.

[SE06] Sabrina Simmons and Zachary Estes. Using latent semantic analysis to estimate similarity. In *Proceedings of the Cognitive Science Society*, pages 2169–2173, 2006.

[SEL+10] Zhaohui Sun, Mounir Errami, Tara Long, Chris Renard, Nishant Choradia, and Harold Garner. Systematic characterizations of text similarity in full text biomedical publications. *PLoS One*, 5(9):e12704, 2010.

[Sér15] Gilles Sérasset. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web*, 6(4):355–361, 2015.

[SGM95] Narayanan Shivakumar and Hector Garcia-Molina. Scam: A copy detection mechanism for digital documents. 1995.

[SGWG06] Daria Sorokina, Johannes Gehrke, Simeon Warner, and Paul Ginsparg. Plagiarism detection in arxiv. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 1070–1075. IEEE, 2006.

[SKS07] Benno Stein, Moshe Koppel, and Efstathios Stamatatos. Plagiarism analysis, authorship identification, and near-duplicate detection pan'07. In *ACM SIGIR Forum*, volume 41, pages 68–71. ACM, 2007.

[SLL97] Antonio Si, Hong Va Leong, and Rynson WH Lau. Check: a document plagiarism detection system. In *Proceedings of the 1997 ACM symposium on Applied computing*, pages 70–77. ACM, 1997.

[Smi11] Jonathan C Smith. *Pseudoscience and extraordinary claims of the paranormal: a critical thinker's toolkit*. John Wiley & Sons, 2011.

[SN02] Patrick M Scanlon and David R Neumann. Internet plagiarism among college students. *Journal of College Student Development*, 43(3):374–385, 2002.

[SSJ11] Ron Scollon, Suzanne Wong Scollon, and Rodney H Jones. *Intercultural communication: A discourse approach*. John Wiley & Sons, 2011.

[Sta11] Efstathios Stamatatos. Plagiarism detection using stopword n-grams. *Journal of the Association for Information Science and Technology*, 62(12):2512–2527, 2011.

[Ste42] ZWEIG Stefan. Le monde d'hier: souvenirs d'un européen, 1942.

[SVB+18] Marwa Salah, Loïc Vial, Hervé Blanchon, Mounir Zrigui, Benjamin Lecouteux, and Didier Schwab. La désambiguïsation lexicale d'une langue moins bien dotée, l'exemple de l'arabe. In *25e conférence sur le Traitement Automatique des Langues Naturelles*, 2018.

[SWA03] Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85. ACM, 2003.

[SZE06] Benno Stein and Sven Meyer Zu Eissen. Near similarity search and plagiarism analysis. In *From data and information analysis to knowledge engineering*, pages 430–437. Springer, 2006.

[Tie09] Jörg Tiedemann. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248, 2009.

[Tie12] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218, 2012.

[TP10] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

[TR85] Victor Nabhan Thomas Ramsauer. L'abc du droit d'auteur. 1985.

[TRB10] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[TVGK10] George Tsatsaronis, Iraklis Varlamis, Andreas Giannakoulopoulos, and Nikolaos Kanellopoulos. Identifying free text plagiarism based on semantic similarity. In *Proceedings of the 4th International Plagiarism Conference*. Citeseer, 2010.

[TWY+14] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

[TZLW17] Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. Ecnu at semeval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, 2017.

[UD03] Özlem Uzuner and Randall Davis. Content and expression-based copy recognition for intellectual property protection. In *Proceedings of the 3rd ACM workshop on Digital rights management*, pages 103–110. ACM, 2003.

[Ukk95] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.

[VCR15] Vedran Vukotic, Vincent Claveau, and Christian Raymond. Irisa at deft 2015: supervised and unsupervised methods in sentiment analysis. In *DeFT, Défi Fouille de Texte, joint à la conférence TALN 2015*, 2015.

[VDK+98] Arie Van Deursen, Paul Klint, et al. Little languages: Little maintenance? *Journal of software maintenance*, 10(2):75–92, 1998.

[VH04] Hans Van Halteren. Linguistic profiling for author recognition and verification. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 199. Association for Computational Linguistics, 2004.

[VHBT+05] Hans Van Halteren, Harald Baayen, Fiona Tweedie, Marco Haverkort, and Anneke Neijt. New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics*, 12(1):65–77, 2005.

[VSTC03] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a Semantic Representation of Text via Cross-Language Correlation Analysis. *NIPS-02: Advances in Neural Information Processing Systems*, pages 1473–1480, July 2003.

[Wei73] Peter Weiner. Linear pattern matching algorithms. In *Switching and Automata Theory, 1973. SWAT'08. IEEE Conference Record of 14th Annual Symposium on*, pages 1–11. IEEE, 1973.

[Whi98] Bernard E Whitley. Factors associated with cheating among college students: A review. *Research in higher education*, 39(3):235–274, 1998.

[WHJ+17a] Hao Wu, Heyan Huang, Ping Jian, Yuhang Guo, and Chao Su. Bit at semeval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84, 2017.

[WHJ+17b] Hao Wu, Heyan Huang, Ping Jian, Yuhang Guo, and Chao Su. In proceedings of the 11th international workshop on semantic evalua- tion (semeval 2017). (SemEval 2017), 2017.

[Wik06] WikiAr. Arabic Wikipedia Corpus, www.linguatools.org /tools/corpora/wikipedia-monolingual-corpora/. 2006.

[Wil02] Jeremy B Williams. The plagiarism problem: are students entirely to blame. In *Proceedings 19th ASCILITE Conference*. Citeseer, 2002.

[Win99] William E Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[Wis93] Michael J Wise. String similarity via greedy string tiling and running karp-rabin matching. *Online Preprint, Dec*, 119, 1993.

[WW10] Debora Weber-Wulff. Test cases for plagiarism detection software. In *Proceedings of the 4th International Plagiarism Conference*, 2010.

[YC05] Hui Yang and Jamie Callan. Near-duplicate detection for erulemaking. In *Proceedings of the 2005 national conference on Digital government research*, pages 78–86. Digital Government Society of North America, 2005.

[YMO17] Xiao Yang, Craig Macdonald, and Iadh Ounis. Using word embeddings in twitter election classification. *Information Retrieval Journal*, pages 1–25, 2017.

[YN05] Rajiv Yerra and Yiu-Kai Ng. A sentence-based copy detection approach for web documents. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 557–570. Springer, 2005.

[ZES06] Sven Meyer Zu Eissen and Benno Stein. Intrinsic plagiarism detection. In *European Conference on Information Retrieval*, pages 565–569. Springer, 2006.

[zG09] Karl-Theodor zu Guttenberg. *Verfassung und Verfassungsvertrag: Konstitutionelle Entwicklungsstufen in den USA und der EU*. Duncker & Humblot, 2009.

[ZMKG09] Mario Zechner, Markus Muhr, Roman Kern, and Michael Granitzer. External and intrinsic plagiarism detection using vector space models. In *Proc. SEPLN*, volume 32, pages 47–55, 2009.

[ZMM+15] Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer, 2015.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.