Méthode alternative à la détection de « copier/coller » : intersection de textes et construction de séquences maximales communes

Jérémy Ferrero*, Alain Simac-Lejeune**

Compilatio
276, rue du Mont-Blanc
74520 Saint-Félix, France
* jeremyf@compilatio.net
** alain@compilatio.net

Résumé. La détection du plagiat passe le plus souvent par la phase de recherche de similitudes la plus naïve, la détection de « copier/coller ». Dans cet article, nous proposons une méthode alternative à l'approche standard de comparaison mot à mot. Le principe étant d'effectuer une intersection des deux textes à comparer, récupérant ainsi un tableau des mots qu'ils ont en commun et de ne conserver que les séquences maximales des mots se suivant dans l'un des textes et existant également dans l'autre. Nous montrons que cette méthode est plus rapide et moins coûteuse en ressources que les méthodes de parcours de textes habituellement utilisées. L'objectif étant de détecter les passages identiques entre deux textes plus rapidement que les méthodes de comparaison mot à mot, tout en étant plus efficace que les méthodes n-grammes.

1 Introduction

Lorsque l'on cherche à comparer deux documents, on recherche tout élément présent dans l'un qui est également présent dans l'autre, ces éléments sont dénommés "similitudes". Les plus évidentes à voir à l'œil humain sont les similitudes exactes, les parties copiées d'un document directement dans l'autre. Cependant, reproduire informatiquement cette capacité humaine est une opération délicate. Ce procédé est souvent gourmand en temps, car passant par une comparaison mot à mot afin d'identifier les séquences de mots identiques dans les deux textes. De ce fait, des méthodes beaucoup moins gourmandes ont vu le jour. Basées sur un système de n-grammes, elles extraient des séquences de n mots se suivant d'un texte et en cherche la présence dans l'autre. C'est dans l'optique de proposer une alternative à ces méthodes que nous allons décrire dans cet article une nouvelle approche de construction de séquences communes.

Après avoir présenté l'état de l'art, nous décrirons dans un premier temps le processus d'intersection des deux textes, ensuite, la phase de construction des plus longues séquences communes et pour finir, nous présenterons l'évaluation de notre approche en la comparant aux méthodes naïves de comparaison mot à mot et à la méthode classique n-grammes.

2 Le « copier/coller »

2.1 Le phénomène « copier/coller »

Le « copier/coller » touche particulièrement les étudiants, en Europe, 34,5% (Guibert et Michaut, 2011) d'entre eux auraient déjà recopié tout ou partie d'un document pour le présenter comme travail personnel. Cette fréquence rejoint celle de travaux américains (Park, 2003) estimant à environ 30% la proportion d'étudiants ayant produit un travail reprenant des phrases d'Internet sans en citer la source. Une étude européenne (Gibney, 2006) révèle que près d'un étudiant français sur deux (46%) a déjà fait usage du plagiat pendant son cursus, contre environ un tiers des étudiants anglais et 10% des étudiants allemands. Ces résultats, qui paraissent déjà impressionnants, pourraient pourtant encore être sous-évalués. En effet, toujours selon la même étude, 40% des étudiants ne comprennent pas ce que signifie réellement le plagiat et n'assimilent pas le « copier/coller » à de la tricherie. La recherche de « copier/coller » entre deux textes joue donc un rôle essentiel dans la prévention du plagiat et la protection du droit d'auteur.

2.2 État de l'art

Les « copier/coller » sont en théorie les similitudes textuelles les plus facilement repérables et identifiables. En effet, la détection de celles-ci équivaut à comparer l'égalité entre deux textes. Pour effectuer cette recherche automatiquement on est obligé de procéder à une comparaison mot à mot. Cette opération, étant beaucoup trop chronophage pour être intégrée dans des solutions à but commercial ou hébergées en ligne, comme des services anti-plagiat, des techniques alternatives ont dû être mises au point.

Les méthodes les plus efficaces restent les méthodes classiques dites n-grammes (Torrejón et Ramos, 2013), qui consistent à construire puis comparer à partir de textes, des séquences de n éléments pouvant être des syllabes, des mots, des entités nommées, etc. La recherche de Barron-Cedeño et Rosso (2009) prouve qu'en prenant des "n-words" (séquence de n mots se suivant) de petites tailles, deux ou trois par exemple, les résultats sont bien meilleurs qu'en utilisant des longues séquences avec un n important. Sur le même principe mais plus originale, on peut citer la méthode de Stamatatos (2009), utilisant des n-grammes mais lors d'une détection intrinsèque, c'est-à-dire sans utilisation de document externe, on ne cherche pas des similitudes avec d'autres documents mais on étudie l'intérieur même du document analysé pour y repérer des irrégularités, des zones suspectes. Les n-grammes les plus pertinents ne sont pas toujours des séquences de mots, comme en atteste le travail de Shrestha et Solorio (2013), des n-grammes de mots vides (stop words) et d'entités nommées sont également utilisés pour détecter des parties de textes similaires entre deux documents. Toutefois, les méthodes les plus répandues sont les méthodes "fingerprint", créant une empreinte du document pour la comparer avec celle d'autres documents. La plupart de ces méthodes (Kent et Salim, 2010) utilisent également des n-grammes pour construire l'empreinte des documents.

Les méthodes "fingerprint" divisent la plupart du temps le document en grammes de longueur n, ainsi les empreintes de deux documents peuvent être comparées et les points (i.e. grammes) concordants, identifiés comme étant des passages identiques dans les textes. Certaines méthodes de "fingerprint" (Stein et Eissen, 2006, 2007; Lyon et al., 2001) vont au-delà de la recherche de similitudes exactes et introduisent la notion de « similarités proches »

pouvant ainsi détecter les paraphrases. Toujours dans cette optique, des recherches plus récentes (Simac-Lejeune, 2013; Kong et al., 2013) ne se contentent pas de comparer des mots ou groupes de mots d'un document à un autre mais tentent d'établir une corrélation « sémantique » entre deux documents par une approche utilisant des mots-clefs.

3 Notre approche

3.1 Intersection de deux textes

L'idée de cette première étape est d'effectuer une intersection de deux textes, afin d'obtenir un tableau des mots présents dans les deux textes tout en conservant la position qu'ils ont dans l'un des deux. La procédure utilisée durant cette étude est la suivante :

1. passage en minuscule des deux textes à comparer ;
2. transformation en tableaux de ces deux phrases en segmentant sur les espaces et les caractères de ponctuation (lemmatisation) (chaque tableau représente une phrase et chaque cellule d'un tableau contient un lemme de la phrase à laquelle il correspond) ;
3. intersection des deux tableaux créés en conservant les offsets (positions) des mots du premier tableau et donc de la première phrase.

3.2 Construction de séquences maximales communes

La seconde et dernière étape consiste à construire, à partir du tableau obtenu à l'étape précédente, les séquences d'un minimum de n mots se suivant dans le premier texte, se suivant donc dans le tableau et étant également présentes dans le second texte. Le seuil n est le nombre de mots se suivant à partir duquel on peut déterminer qu'une séquence est la copie d'une autre et qu'elle n'est pas due au hasard. Nous pourrions dès lors nous poser la question : à partir de combien de mots se suivant une séquence peut être considérée comme réellement copiée ? En effet, il existe des séquences de trois mots ou plus, suffisamment fréquentes dans la langue, pour fausser la comparaison, comme les séquences « il était une fois » ou « nulle par ailleurs ». Cependant, les résultats des travaux de Barron-Cedeño et Rosso (2009) démontrent que sur de larges textes, il est tout aussi efficace de fixer un n petit, à deux ou trois par exemple.

La procédure de construction des séquences est la suivante :

1. on déplace une fenêtre de glissement de n éléments dans le tableau en fonction du seuil n choisi afin de constituer des "n-words" se suivant donc forcément dans le premier texte ;
2. pour chaque "n-word" constitué, on vérifie son existence dans le second texte ;
3. tant qu'une correspondance est trouvée et que la séquence existe bien dans les deux textes, on construit la séquence de taille $n + 1$ en y concaténant le mot suivant du tableau ;
4. dès que la séquence ne s'y trouve plus, on récupère la séquence maximale commune (la séquence essayée précédemment avant que le test échoue) et on recommence depuis l'étape 1 en déplaçant la fenêtre de glissement sur le mot suivant et en reprenant le n initial.

4 Évaluation et tests

4.1 La base de tests et protocole

La base de tests est composée de 200 textes, allant de 100 mots à environ 20 000 mots (avec une moyenne de 1500 mots), représentant 500 comparaisons de textes deux à deux annotés manuellement afin de savoir quel passage est réellement la copie d'un autre. Pour tester correctement les performances des algorithmes évalués, la base comporte aussi bien des passages entièrement copiés que des paraphrases ou des reformulations plus complexes, ainsi que des textes « pièges » traitant du même sujet et donc employant le même vocabulaire mais n'étant pas pour autant un « copier/coller » ou une reformulation quelconque d'un autre texte présent dans le corpus. L'intégralité de ces textes est en français. Ci-dessous la répartition des comparaisons :

— 120 comparaisons effectuées afin de détecter des textes entièrement « copier/coller » de façon exact ;
— 80 comparaisons afin de détecter des textes entièrement paraphrasés ou reformulés ;
— 200 comparaisons entre des textes ne comportant que quelques passages rigoureusement identique (copier/coller) ;
— 100 comparaisons entre deux textes ne comportant que quelques passages « similaires » (paraphrases ou reformulations de phrases et/ou paragraphes).

Ces comparaisons sont réparties entre des travaux d'élèves (mémoires financiers et scientifiques) avec leurs sources, différentes versions à différentes dates d'un même article de Wikipédia et des extraits du corpus de la PAN-CLEF 2014 en matière d'alignement de textes.

4.2 Résultats

Les résultats obtenus sur le corpus de test, par la méthode naïve de comparaison mot à mot, la méthode classique des n-grammes et notre méthode, sont représentées dans le tableau 1. La méthode n-grammes évaluée est celle décrite dans l'article de Barron-Cedeño et Rosso (2009). L'algorithme de comparaison mot à mot est le plus efficace avec un rappel de 1 pour une

Méthodes	Précision	Rappel
mot à mot	0.76	1
n-grammes (avec n=2)	0.72	0.78
n-grammes (avec n=3)	0.78	0.64
n-grammes (avec n=4)	0.88	0.59
n-grammes (avec n=8)	1	0.49
nous (avec n=2)	0.76	1
nous (avec n=3)	0.83	0.93
nous (avec n=4)	0.92	0.84
nous (avec n=8)	1	0.58

TAB. 1 – *Précision et rappel des différents algorithmes avec différents n.*

précision de 0.76. L'utilisation de cet algorithme revient à utiliser notre méthode avec $n = 2$,

en effet toute séquence commune (de plus d'un mot) entre deux textes est alors considérée comme « copier/coller ».

Les résultats confirment le postulat de Barron-Cedeño et Rosso (2009) disant que prendre un n de petite taille augmente l'efficacité de détection, sachant que prendre des bigrams favorise le rappel, tandis que prendre un n supérieur favorise la précision. Ce phénomène s'explique par le fait que prendre un petit n forme des séquences courtes, on ne manque ainsi aucune correspondance mais on favorise les faux positifs, baissant alors la précision. En revanche prendre un n plus important construit des séquences plus longues, réduisant ainsi la correspondance de chaînes et donc le rappel mais augmentant le taux de certitude des concordances et donc la précision. Cet article ne pose pas la question d'optimisation de la détection en fonction du n choisi, on fixe donc $n = 2$ pour la suite de notre évaluation.

On constate dans le tableau 1 que notre méthode donne de meilleur résultat que celle des n-grammes (0.76 de précision contre 0.72 et 1 de rappel contre 0.78 avec $n = 2$ pour les deux méthodes). Toutefois, on peut voir dans le tableau 2 qu'en moyenne elle est 15% moins rapide et 30% plus coûteuse en mémoire que la méthode n-grammes (en allouant 6.48 Mbits de mémoire en 46.57 secondes pour une comparaison d'environ 20 000 mots contre 4.78 Mbits alloués en 39.68 secondes pour la méthode n-grammes).

	Mot à mot		n-grammes		Notre méthode	
	Temps	MA	Temps	MA	Temps	MA
200	0.1	0.34	0.01	0.02	0.02	0.14
1000	1.5	3.76	0.02	0.22	0.18	0.44
1800	4.4	13.01	0.34	0.38	0.57	0.67
4500	15.81	40.8	1.14	0.82	1.71	1.23
8400	94	220.86	8.27	1.72	10.68	3.13
16000	334	828.45	32.56	4.13	37.62	5.6
20000	502	1237.71	39.68	4.78	46.57	6.98

TAB. 2 – *Temps d'exécution en secondes et mémoire allouée (MA) en Mégabits de chaque algorithme utilisé avec n=2 en fonction du nombre moyen de mots des textes à comparer.*

5 Conclusions

Notre approche montre donc des résultats supérieurs aux méthodes n-grammes classiques, dans le sens où elle recherche une séquence de taille minimale n et agrandit si possible la séquence trouvée afin d'obtenir une séquence maximale commune. En revanche, elle est tout aussi dépendante du nombre n choisi que les méthodes n-grammes. Son temps d'exécution et son usage de la mémoire restent supérieurs à ceux des méthodes n-grammes bien que nettement inférieurs à ceux de la méthode mot à mot.

Pour des travaux futurs, nous envisageons de confronter nos résultats à des méthodes n-grammes plus sophistiquées comme celles décrites dans l'article de Shrestha et Solorio (2013).

Pour conclure, bien que moins rapide, notre méthode montre une précision équivalente aux méthodes n-grammes tout en proposant un rappel nettement supérieur.

Références

Barron-Cedeño, A. et P. Rosso (2009). On Automatic Plagiarism Detection Based on n-Grams Comparison. In M. Boughanem (Ed.), *Proceedings of the European Conference on Information Retrieval (ECIR'09)*, LNCS, pp. 696–700. Springer Berlin.

Gibney, E. (2006). I'm No Plagiarist, I Moved a Comma. *The Times Higher Education Supplement : THE*. No. 2104.

Guibert, P. et C. Michaut (2011). Le plagiat étudiant. Volume 28 of *Education et sociétés*, pp. 214. De Boeck Supérieur.

Kent, C. K. et N. Salim (2010). Features Based Text Similarity Detection. *Journal Of Computing*, 53–57. Volume 2, Issue 1.

Kong, L., H. Qi, C. Du, M. Wang, et Z. Han (2013). Approaches for Source Retrieval and Text Alignment of Plagiarism Detection. In *Notebook for PAN at CLEF 2013*.

Lyon, C., J. Malcolm, et B. Dickerson (2001). Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp. 118–125.

Park, C. (2003). In other (people's) words : plagiarism by university students - literature and lessons. *Assessment and Evaluation in Higher Education 28*(5), 471–488.

Shrestha, P. et T. Solorio (2013). Using a Variety of n-Grams for the Detection of Different Kinds of Plagiarism. In *Notebook for PAN at CLEF 2013*.

Simac-Lejeune, A. (2013). Calcul de distance inter-documents par approche mots-clés. In *EGC*, pp. 501–506.

Stamatatos, E. (2009). Intrinsic Plagiarism Detection Using Character n-gram Profiles. In *Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*, pp. 38–46.

Stein, B. et S. M. Z. Eissen (2006). Near Similarity Search and Plagiarism Analysis. In *From Data and Information Analysis to Knowledge Engineering*, pp. 430–437. Springer Berlin.

Stein, B. et S. M. Z. Eissen (2007). Fingerprint-based Similarity Search and its Applications. In *Forschung und wissenschaftliches Rechnen*, pp. 85–98.

Torrejón, D. A. R. et J. M. M. Ramos (2013). Text Alignment Module in CoReMo 2.1 Plagiarism Detector. In *Notebook for PAN at CLEF 2013*.

Summary

Plagiarism detection most commonly use the most naive phase of similarities search, the detection of copy and paste. In this paper, we propose an alternative method to the standard verbatim comparison approach. The idea is to carry out an intersection of two texts to get a table of common words and to keep only the maximum sequences of consecutive words in one of the texts which also exists in the other. We show that this method is faster and less expensive in memory that commonly used scan texts methods. The goal is to detect identical passages between two texts faster than verbatim comparison methods, while operating more efficient than the n-grams.