Impact de la correction automatique de l'OCR/HTR sur la reconnaissance d'entités nommées dans un corpus bruité

Ljudmila Petkovic, Motasem Alrahabi, Glenn Roe

Journal of Information Sciences
https://doi.org/10.34874/IMIST.PRSM/jis-v21i2.36599

Impact de la correction automatique de l'OCR/HTR sur la reconnaissance d'entités nommées dans un corpus bruité

Impact of Automatic OCR/HTR Correction on Named Entity Recognition in Noisy Corpora

Ljudmila PETKOVIC [1], Motasem ALRAHABI [2], Glenn ROE [3]

[1] ObTIC - Sorbonne Université ljudmila - petkovic@sorbonne-universite.fr
[2] ObTIC - Sorbonne Université - motasem.alrahabi@sorbonne-universite.fr
[3] ObTIC - Sorbonne Université - glenn.roe@sorbonne-universite.fr

Résumé

Nous présentons une expérience menée sur la correction d'orthographe automatique de textes issus de la reconnaissance optique des caractères (OCR), dans l'objectif de mesurer l'impact de la correction sur une tâche d'extraction d'informations. À partir d'un échantillon de documents d'archives numérisées (océrisées), nous avons appliqué un système de reconnaissance d'entités nommées avant et après une correction d'orthographe. Les résultats obtenus ont montré que le correcteur orthographique permet d'améliorer certaines tâches du traitement automatique du langage naturel. Une extension de l'approche proposée par un ré-entraînement sur un plus grand corpus est également présentée pour optimiser davantage les résultats obtenus.

Mots-clés : correction d'orthographe automatique ; OCR; documents d'archives numérisées; reconnaissance d'entités nommées; traitement automatique des langues.

Abstract

We present an experiment conducted on the automatic spelling correction of texts resulting from optical character recognition (OCR), with the objective of measuring the impact of corrections on an information extraction task. Using a sample of OCRed digitized archival documents, we applied a named entity recognition system before and after orthographic correction. The results obtained showed that the spelling checker improves certain automatic natural language processing tasks. An extension of the proposed approach by re-training on a larger corpus is also presented to further optimize the results obtained.

Keywords: Automatic Spelling Correction; OCR; Digitized Archival Documents; Named Entity Recognition; Natural Language Processing.

Introduction

Le volume des données numérisées en lettres et sciences humaines et sociales ne cesse de croître donnant naissance à de nouvelles techniques de traitement automatique des langues. Dans cet environnement dynamique, caractérisé par cette profusion de données d'une part, et la croissance des quantités d'informations qu'elle englobe de l'autre, les besoins d'analyse ont également évolué, requérant le développement de nouveaux outils de plus en plus puissants mais aussi capables d'intégrer des informations contextuelles adaptées aux types de données traitées (Liu et al., 2023). Dans quelle mesure ces outils exploitent-ils les données ? Une correction de ces données est-elle nécessaire, et par quel moyen ? Quel est l'impact d'une telle correction sur les tâches d'édition numérique, d'extraction d'informations et de fouille textuelle ?

Ce travail vise à améliorer la qualité des documents d'archives numérisées, dans le cadre du projet patrimonial de la Très Grande Bibliothèque (TGB)[1], par le biais d'algorithmes d'apprentissage automatique et de traitement automatique des langues (TAL). Le corpus en question a été numérisé grâce à la technologie OCR[2], qui convertit électroniquement les images contenant du texte dactylographié ou manuscrit, en format texte analysable par l'ordinateur. Certains systèmes de reconnaissance automatique des textes présentent des performances de transcription très solides, dont le taux d'erreur de caractères s'approche de 0.01% (Reul et al., 2018). Malgré ces performances et le gain de temps de transcription apporté par ces technologies, l'exploitation directe des données générées de cette manière reste problématique à cause du bruit issu de l'OCR (insertion, substitution, suppression, etc.), en particulier dans les documents historiques (Hamdi et al., 2022). Il est alors possible de mener des actions d'amélioration du processus OCR lui-même (par le biais de l'entraînement de modèles OCR), voire des actions correctives post-OCR.

Nous nous plaçons dans le second cadre en expérimentant des outils de correction automatique de la couche texte des documents océrisés, pour faciliter les tâches de TAL en aval, en l'occurrence, la Reconnaissance d'Entités Nommées (REN). En effet, l'extraction d'information, et plus particulièrement la REN nous semble un point d'entrée privilégié pour mesurer la qualité des textes océrisés (Sagot & Gábor, 2014 ; Koudoro-Parfait et al., 2021 ; Hamdi et al., 2022). Notre approche consiste à comparer la qualité de REN avant et après la correction automatique des documents d'archives océrisées. L'évaluation menée sur l'extraction des EN nous a permis de repérer celles qui ont été identifiées avant et après correction, ainsi que de dresser une typologie de modifications apportées par cette correction. Les résultats de notre étude montrent une certaine robustesse

[1] http://obvil.lip6.fr/tgb/.
[2] Angl. *Optical Character Recognition*, soit la reconnaissance optique de caractères.

du système de REN utilisé sur les textes bruités (`spaCy`)[3], ainsi que les avantages et les inconvénients de l'utilisation d'un correcteur orthographique (`JamSpell`)[4] en amont de la tâche de REN.

L'article est organisé de la manière suivante : après l'introduction, nous proposons d'abord une revue de la littérature qui s'appuie sur la problématique de la correction des sorties OCR et les approches utilisées à cet effet (Section 1). La Section 2 présente la méthodologie de l'évaluation de l'impact de la correction automatique d'OCR sur la REN, ainsi que ses limites. Finalement, la dernière section présente une conclusion et propose des pistes pour des recherches futures.

1. Correction post-OCR : état de l'art

En reconnaissance automatique des textes, nous distinguons généralement les méthodes d'OCR et les méthodes de reconnaissance du texte manuscrit (abbr. *HTR*, angl. *Handwritten Text Recognition*) : la principale différence tient au fait que l'OCR se base sur la reconnaissance des *caractères* individuels, alors que l'HTR traite les blocs de texte, notamment les lignes (Gabay & SciCoS, 2022)[5]. Certains systèmes d'HTR de pointe, notamment Transkribus[6], ainsi que d'OCR (Kraken[7], Tesseract[8], OCRopus[9], Calamari[10] ou ABBYY Finereader[11]) (Reul et al., 2019), présentent des performances de transcription très solides, dont le taux d'erreur de caractères (angl. *Character Error Rate — CER*) pour les deux derniers s'approche de 0.01% (Reul et al., 2018). Selon le travail d'Oger (Oger et al., 2012), les erreurs d'OCR[12] peuvent être regroupées en 2 catégories : celles des *non-mots* (angl. *non-word errors*) qui ne représentent pas des mots valides de la langue, p. ex., si le mot « maman » est écrit « maaman » (Wisniewski et al., 2010), contrairement à celles des *mots réels* (angl. *real-word errors*), ce qui est le cas, par exemple, du mot « dessert », grammaticalement correct mais incorrectement saisi à la place de « désert », dans la phrase

[3] https://spacy.io/models/fr.
[4] https://github.com/bakwc/JamSpell.
[5] Il faut également souligner la double capacité de la technologie HTR de transcrire non seulement les textes manuscrits, mais également les imprimés (Chagué, 2021). Pourtant, l'HTR trouve sa véritable application dans le traitement des textes manuscrits, étant donné que ce type de document comprend très souvent des lettres liées l'une à l'autre, ce qui nécessite le traitement des lignes complètes (et non pas la séparation des caractères individuels fournie par la technologie OCR) pour rendre la transcription plus correcte.
[6] https://readcoop.eu.
[7] https://kraken.re/master/index.html.
[8] https://tesseract-ocr.github.io.
[9] https://github.com/ocropus/ocropy.
[10] https://github.com/Calamari-OCR/calamari.
[11] https://pdf.abbyy.com.
[12] Pour des raisons de brièveté, sous le terme *OCR* dans les expressions *modèle d'OCR*, *post-OCR*, *correction des sorties OCR*, *erreurs d'OCR* etc. nous entendons les transcriptions automatiques des textes en général, indépendamment de la méthode sous-jacente du modèle (OCR ou HTR).

« Le Sahara est un désert ». Dans le dernier cas il s'agit donc d'erreurs *grammaticales*, ou d'erreurs *sémantiques, dépendantes du contexte* (angl. *semantic/context-sensitive errors*) (Azmi et al., 2019)[13]. Du point de vue du calcul des permutations de caractères produisant des erreurs d'orthographe, la distance minimum d'édition de Levenshtein (Levenshtein, 1965) est très utilisée dans la littérature, qui permet de calculer les similitudes entre les caractères et les mots. Toutefois, son inconvénient majeur est la matrice des coefficients qui est figée à priori, indépendamment du contexte. D'autres algorithmes comme *wmd* (*Word Mover's Distance*) (Wei et al., 2022) ont l'avantage d'intégrer les significations implicites des mots lors du calcul des différences entre les mots dans les documents.

Le domaine de la correction automatique de texte est très actif et remonte à plusieurs décennies, depuis les travaux de (Damerau, 1964) jusqu'à nos jours. Plusieurs institutions patrimoniales (p. ex. la BNF)[14] et universitaires (p. ex. l'université Louis-et-Maximilien de Munich ou l'université de Leipzig)[15] se sont attelées à cette tâche, et les autres initiatives, comme les compétitions ICDAR[16] sur la post-correction d'OCR (Rigaud et al., 2019), ne manquent pas. Les travaux relevant de cette problématique incluent non seulement les langues traditionnellement placées au centre d'intérêt des chercheurs, comme l'anglais (Nguyen et al., 2020), le français (Sagot & Gábor, 2014), l'allemand (Martin et al., 2011), l'italien (Bolioli et al., 2014), l'arabe (Mubarak & Darwish, 2014), etc., mais également les langues et les dialectes moins étudiés, comme les langues africaines (Enguehard & Mbodj, 2014 ; Salifou & Naroua, 2014), le serbe (Krstev & Stankovic, 2019), le suisse allemand ou le romanche (Martin et al., 2011), voire les langues en voie de disparition : aïnou, griko, et yakkha (Rijhwani et al., 2020).

De nombreuses classifications (à différents niveaux de granularité) des approches de correction des textes bruités sont répertoriées dans l'état de l'art, mais elles ne sont pas unanimes, comme en témoignent certains travaux de recherche (Bassil & Alwani, 2012 ; Edwards, 2016 ; Nguyen et al., 2020 et Nguyen et al., 2021). Un dénominateur commun émerge cependant de toutes ces études, résultant en trois grandes méthodes qui se démarquent : méthodes lexicales, méthodes utilisant les modèles de langue, et méthodes à base d'apprentissage artificiel.

---

[13] Il existe aussi les erreurs dépendantes du contexte au sein d'une phrase, laquelle nécessite une analyse syntaxique afin d'être désambiguïsée. (Edwards, 2017) illustre ce phénomène avec la phrase « Le pilote ferme la porte », dont une décomposition possible (mais peut-être moins évidente) comporte l'adjectif « ferme », le pronom « la » suivi du verbe « porte ».

[14] Abr. de *Bibliothèque Nationale de France*. Cf. le projet AMELIOCR de la post-correction des ouvrages anciens en exploitant les associations lexicales de l'OCR bruité https://actions-recherche.bnf.fr/BnF/anirw3.nsf/IX01/A2016000030_post-correction-d-ocr-pour-les-ouvrages-anciens-en-exploitant-les-associations-lexicales-de-l-ocr-bruite.

[15] https://pro.europeana.eu/page/issue-13-ocr.

[16] Abr. angl. de *International Conference Document Analysis and Recognition*.

Afin de pallier au problème de la croissance exponentielle du temps de correction manuelle des sorties bruitées d'OCR, une pléthore d'outils a été conçue à cet effet. Dans le sillage des travaux de (Norvig, 2007)[17], `pySpellChecker`[18] vérifie l'orthographe d'un texte et suggère des corrections basées sur l'utilisation de l'algorithme de distance de Levenshtein et des dictionnaires pour trouver des permutations à une distance d'édition de 2 du mot d'origine.[19] D'un autre côté, `JamSpell`[20] est un correcteur orthographique basé sur les modèles de langue (statistiques, dans le cas de l'option *open source*, contrairement aux options payantes exploitant des modèles d'apprentissage artificiel du type *CatBoost*[21]). Cette librairie utilise les trigrammes pour corriger les textes multilingues et sélectionner les candidats de correction avec les scores les plus élevés en utilisant principalement les arbres de décision, tout en étant optimisé pour la vitesse (algorithme `SymSpell`[22] modifié) et la consommation de mémoire (filtre de *Bloom* et hachage parfait)[23]. Pour ce qui est de l'approche par réseaux de neurones, mentionnons la librairie `NeuSpell`[24] (Jayanthi et al., 2020) qui comprend plusieurs modèles de correction orthographique de ce type (LSTM[25], ELMo[26] et BERT[27]), mais aussi les correcteurs non-neuronaux (`Aspell`[28] ou `JamSpell` précédemment mentionné). La librairie `Contextual Spell Check`[29] de `spaCy` fournit également la correction d'orthographe à l'aide de l'architecture BERT, avec un focus sur la correction des *erreurs de non-mots* et de *mots hors vocabulaire*[30]. La méthodologie BERT avec les plongements des caractères (angl. *character embeddings*) a été également appliquée par (Nguyen et al., 2020) pour la correction d'OCR.

---

[17] Travail très souvent cité et utilisé comme le modèle *baseline* pour les évaluations des correcteurs automatiques – cf. p. ex. (de Amorim & Zampieri, 2013).
[18] https://pyspellchecker.readthedocs.io/en/latest/.
[19] Sous le terme *distance d'édition* nous entendons le nombre minimum d'opérations d'édition nécessaires pour réécrire un mot *w* en mot *w'* (Tantini et al., 2011). Par exemple, la distance d'édition entre les mots « mots » et « mode » est 2, en raison de deux substitutions qui s'effectuent ($t \rightarrow d$; $s \rightarrow e$).
[20] https://jamspell.com.
[21] https://catboost.ai. Ce modèle utilise le GPU, ce qui le rend rapide en termes d'exécution des tâches.
[22] https://github.com/wolfgarbe/SymSpell.
[23] D'après la discussion sur la page des *issues* GitHub https://github.com/bakwc/JamSpell/issues/15. Par ailleurs, les trigrammes sont considérés comme le modèle de langage le plus performant (Bendib, 2018).
[24] https://github.com/neuspell/neuspell.
[25] Abr. angl. *Long Short-Term Memory*.
[26] Abr. angl. *Embeddings for Language Model*.
[27] Abr. angl. *Bidirectional Encoder Representations from Transformers*.
[28] http://aspell.net/metaphone/. `Aspell` convertit le mot en anglais mal orthographié en son équivalent sonore (*métaphone*). Il s'appuie sur l'algorithme phonétique *Metaphone* qui renvoie une approximation de la prononciation d'un mot, qui devrait être la même pour les mots ou les noms qui se ressemblent, et peut être utilisée comme clé de recherche dans le processus de correction orthographique.
[29] https://spacy.io/universe/project/contextualSpellCheck.
[30] Angl. *Out Of Vocabulary* (*OOV*) *words* (Bouzidi et al., 2017). Ici, nous entendons les mots qui ne sont pas reconnus par le système de correction automatique d'OCR.

Le Tableau 1 résume les approches méthodologiques de la correction d'OCR en juxtaposant leurs points forts et faibles :

Approche	Avantages	Inconvénients
Manuelle	- Précision de l'étalon-or - Production collaborative (angl. *crowdsourcing*)	- Chronophage - Indisponibilité des documents originaux pour référence - Efficacité et expertise des correcteurs variables
Lexicale	- Création et extensibilité faciles du dictionnaire	- Chronophage si le dictionnaire ou le texte à corriger sont de grande taille - Incomplétude des dictionnaires - Indépendance du contexte
Modèles de langue	- Dépendance du contexte (à partir des bigrammes n-grammes) lors de la désambiguïsation	- Faible gestion des dépendances à longue distance
Apprentissage	- Dépendance du contexte (à partir des bigrammes n-grammes) lors de la désambiguïsation - Apprentissage par transfert ou par réglage fin - Meilleure gestion des dépendances à longue distance	- Nécessité d'un grand corpus pour l'entraînement - Surapprentissage si les hyperparamètres d'apprentissage sont mal définis - Données d'apprentissage parfois clairsemées

Tableau 1: récapitulatif des avantages et des inconvénients de différentes approches de correction d'OCR.

Parmi les méthodes proposées ci-dessus, nous avons choisi d'expérimenter des correcteurs récents, basés sur les modèles de langage, étant donné leur prise en main relativement rapide, mais aussi leurs meilleures performances par rapport à d'autres bibliothèques de correction d'orthographe, notamment celle de P. Norvig ou `Hunspell`[31] en termes de vitesse et de précision (Kolajo et al., 2022).

[31] http://hunspell.github.io/.

2. Présentation des résultats et discussion

La REN se définit comme la tâche qui consiste à extraire automatiquement des éléments textuels (un mot ou un groupe de mots) dans plusieurs catégories prédéfinies, p. ex. : personne, organisation, localisation, événement, produit, valeur numérique, titre, date, etc. Notre approche consiste à évaluer l'impact de la correction post-OCR sur la tâche de REN. Pour ce faire, nous procédons par comparaison des sorties obtenues à partir d'un corpus annoté par un système de REN avant et après la correction post-OCR. Concrètement, à partir des documents d'archives de la TGB océrisées[32], nous avons extrait le texte brut et récupéré les EN à l'aide du modèle *large* de la librairie `spaCy`[33]. Ensuite, nous avons utilisé l'application `JamSpell`[34], dont le modèle de base est entraîné sur un corpus d'actualités et sur un corpus de Wikipédia (600 000 phrases en tout) pour générer les textes corrigés, à partir desquels nous avons extrait les EN homologues. Finalement, les listes des EN ont été juxtaposées et comparées à l'aide de la librairie `Difflib`[35] afin de trouver les différences entre elles. La chaîne de traitement en question est illustrée dans la Figure 1 :

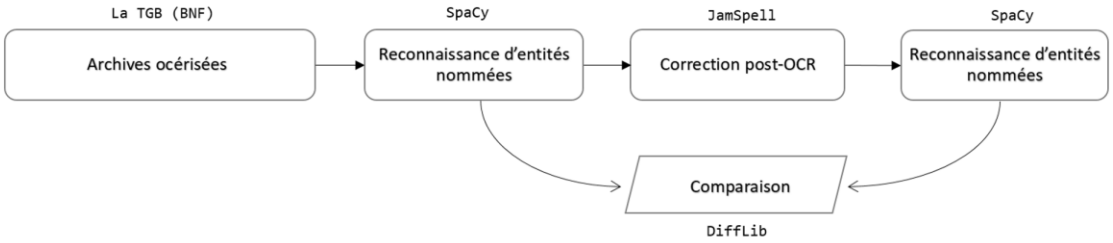

Figure 1: Chaîne de traitement pour l'évaluation de l'impact de la correction post-OCR sur la tâche de la REN.

2.1 Corpus de travail : la Très Grande Bibliothèque

La TGB[36] est une bibliothèque de 128 441 documents français en mode texte (reconnaissance optique des caractères non relue), issus des collections Gallica[37] de la BNF. Le corpus en XML-TEI provient majoritairement de l'édition du XIX^{e} siècle et couvre différentes thématiques (littérature, droit, philosophie, etc.). Les imprimés de cette

[32] Logiciel d'océrisation non renseigné dans la documentation de la TGB de la note de bas de page 1.
[33] https://spacy.io/models/fr#fr_core_news_lg. Le choix de ce modèle est dû au fait qu'il a été entraîné sur les plongements de mots (angl. *word embeddings*) et considéré comme le plus performant pour la REN dans (Koudoro-Parfait et al., 2021), en comparaison avec les modèles *small* et *medium*.
[34] L'implémentation de cette librairie est également intégrée à l'interface de l'application Toolbox (Cordova et al., 2022) (désormais Pandore) qui permet d'utiliser en ligne un ensemble d'outils de manipulation et de traitement de corpus textuels.
[35] https://docs.python.org/3/library/difflib.html.
[36] https://api.bnf.fr/documents-de-gallica-produits-au-format-tei-par-obvil.
[37] http://gallica.bnf.fr/.

période sont à la fois libres de droits et mieux reconnus automatiquement que les imprimés plus anciens (notamment au niveau des caractères comme le s long (ſ), ligatures, etc.).

Afin de créer notre échantillon pour cette expérience, nous avons aléatoirement choisi deux documents à partir du corpus TGB, d'environ un million de caractères. Ce choix a été guidé par le fait que `spaCy` applique une limite maximale d'un million de caractères pour la REN.

2.2 Évaluation

À partir de notre échantillon de texte brut, nous avons généré les EN à partir des textes corrigés et également des textes non corrigés.

Dans un premier temps, nous avons pu observer manuellement les bonnes corrections ou les *vrais positifs* (ex : « Térehce » → « Térence »). D'autres EN ont été identifiées uniquement grâce à la correction automatique (ex : « Eugène »). Néanmoins, le correcteur orthographique a effectué de nombreuses sur-corrections, c'est-à-dire des modifications des formes orthographiques déjà correctes dans le texte océrisé (ex : « Empédocle » en « L'Empésocle »). Ces résultats obtenus corroborent d'autres études basées sur des approches différentes de la nôtre (Huynh et al., 2020). Ces auteurs utilisent un correcteur `SymSpell`, et ils affirment que la correction post-OCR pouvait également dégrader les scores F_1 de la REN, en particulier lorsque le taux d'erreur OCR est très faible.

La comparaison des EN avant et après correction avec l'outil `Diffchecker`[38] nous a permis de faire une première évaluation, suite à laquelle nous avons exploité la librairie `Difflib`[39] afin de calculer le ratio des EN reconnues avant et après correction. Cet outil s'appuie sur la correspondance des motifs `Gestalt`[40] et compare deux chaînes de caractères en comptant les caractères qui correspondent. Il renvoie enfin un score compris entre 0 et 1 où 1 est une identité parfaite.

Le nombre d'EN reconnues avant et après la correction orthographique des textes est résumé dans le Tableau 2.

OCR corrigé	EN uniques	EN en total
Non	3918	6428
Oui	3901	6349

Tableau 2 : Le nombre d'EN reconnues avant et après la correction d'OCR.

En termes du ratio de similarité calculé entre les EN avant et après correction, ce taux s'élève à 90.4%. Nous trouvons dans la Figure 2 un extrait de l'échantillon (non corrigé /

[38] https://www.diffchecker.com/.
[39] https://docs.python.org/3/library/difflib.html.
[40] Angl. *Gestalt pattern matching* (Ratcliff & Obershelp, 1988).

corrigé) avec les changements orthographiques respectifs, visualisés dans la sortie HTML de la librairie `Difflib`.

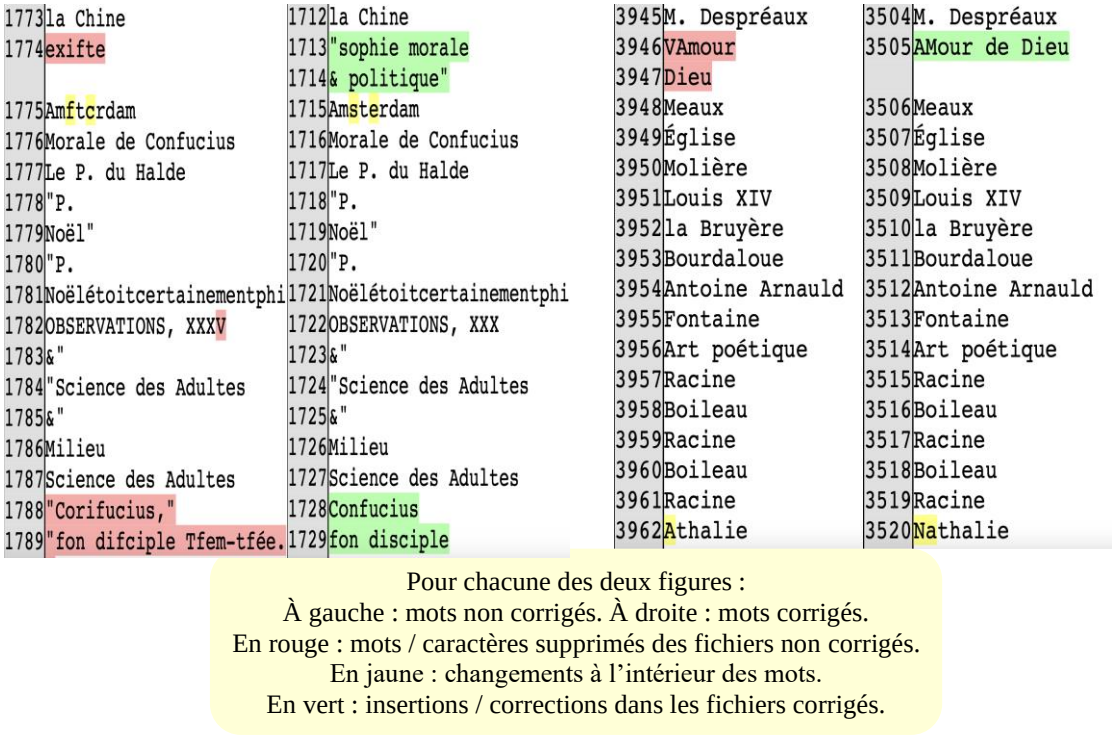

Figure 1 : Visualisations des EN et des changements orthographiques dans la sortie HTML de la librairie `Difflib`.

Les résultats de l'extraction (Figure 2a) indiquent que `spaCy` parvient à détecter des EN malgré les erreurs orthographiques (« Amftcrdam », « "Corifucius," »[41] pour « Am**ste**rdam », « Co**n**fucius »). Cela va dans le sens des résultats obtenus par (Koudoro-Parfait et al., 2021), qui montrent que `SpaCy` est relativement robuste pour cette tâche. La Figure 2b montre la présence des sur-corrections, p. ex. « Athalie » (tragédie de Racine) est incorrectement corrigé en « **Na**thalie ». Ce surapprentissage nous a poussé à chercher de nouvelles techniques hybrides comme la combinaison de : - *stemming* du mot tenant compte du contexte + identification par poids/ corpus. Par ex., après le pronom « les », le poids du dernier s/ x du mot suivant est plus grand : « les paris > pari » ; par contre après la préposition « à », le poids de la dernière lettre est faible : « à paris > Paris ».

Dans le cadre de l'analyse de *distance minimale d'édition* de Levenshtein[42], (Hamdi et al., 2022) indiquent que les techniques post-OCR devraient pouvoir corriger environ 81,49%

[41] À noter la présence des doubles apostrophes introduisant le discours direct dans la version non corrigée.
[42] Les opérations qui s'effectuent au sein des chaînes de caractères sont les suivantes : insertion d'un caractère (p. ex. le mot « mot » réécrit en « m**i**ot »), suppression (absence) d'un caractère (« mot » > « m**Ø**t »),

des entités nommées orthographiquement incorrectes avec un seuil de distance d'édition de 2. La librairie `Difflib` nous a permis de récupérer les types de changements des EN avant et après la correction d'OCR :[43] les EN non corrigés, sans changements (p. ex. « Molière »), les insertions (« paris » [sic] absent du texte non corrigé mais présente dans le texte corrigé, même si l'EN commence par une minuscule), les suppressions (« Alcippe »[44] présente dans le texte non corrigé mais absente dans le texte corrigé) et les remplacements (« Escar**g**ot » dans le corrigé au lieu de « Escar**b**ot »).

Pour calculer et évaluer l'impact des corrections orthographiques sur la REN, nous avons construit le tableau de contingence (matrice de confusion, Tableau 3), à partir duquel nous avons dérivé les métriques standards de précision, de rappel et de mesure F_1. Comme souligné dans (Reynaert, 2008), la détection des erreurs orthographiques s'apparente aux processus liés à la recherche d'information (angl. *information retrieval*, abbr. *IR*), champ disciplinaire où ces métriques sont largement utilisées.

Nous répertorions les catégories suivantes :

- *Vrais Positifs* (*VP*) : mots orthographiquement erronés qui sont correctement identifiés comme tels et corrigés (« Térehce » > « Téren**n**ce »)
- *Vrais Négatifs* (*VN*) : mots orthographiquement corrects qui n'ont pas été corrigés (typiquement la majorité des cas)
- *Faux Positifs* (*FP*) : mots corrigés qui sont originalement les mots orthographiquement corrects, où nous observons le phénomène des sur-corrections (« Empédocle » > « **L'**Empé**s**ocle »), mais aussi des suppressions des mots entiers (silence)
- *Faux Négatifs* (*FN*) : mots orthographiquement erronés qui ont été censés être corrigés (« l'olympe » n'a pas été corrigé en « l'Olympe »)
- Les tokens incorrectement classifiés comme les EN (*NOK*)[45], notamment les tokens commençant par une majuscule car en début de la phrase (« Est »), ceux écrits en capitales (« POUR »), les verbes (« étant ») etc.

substitution d'un caractère par un autre (« mot » > « mo**f** »). À ces traitements s'ajoute parfois celui de la transposition (inversion) des caractères (« mot » > « m**to** »), auquel cas nous nous référons à la distance d'édition Damerau-Levenshtein (Damerau, 1964).

[43] Cf. le fichier https://github.com/ljpetkovic/corr_OCR_NER/blob/main/diffs/changements_types.csv.

[44] La fille d'Arès et d'Aglaure dans la mythologie grecque.

[45] Abbr. angl. *Not OK*.

	Cible		Non-cible		
Corrigés	VP	15	FP	78	
Non corrigés	VN	600	FN	17	
En total	P	93	N	617	710

Tableau 3 : Matrice de confusion qui mesure l'impact des corrections orthographiques sur la REN. Les tokens incorrectement classifiés comme les EN sont exclus de ce tableau.

En l'occurrence, la précision représente le pourcentage d'EN correctement corrigées parmi les EN corrigées, c'est-à-dire. « l'aptitude » du système pour la correction automatique. Le rappel est le pourcentage d'EN correctement corrigées parmi les EN à corriger. Pour pondérer la précision et le rappel de manière égale, la mesure F1 a été également calculée.

Les calculs de ces mesures ont été faits de manière suivante :

$$Précision(P) = \frac{VP}{VP + FP} = \frac{15}{15 + 78} = 0.16 = 16\%$$

$$Rappel(R) = \frac{VP}{VP + FN} = \frac{15}{15 + 17} = 0.47 = 47\%$$

$$F_1 = \frac{2 \times P \times R}{P + R} = \frac{0.1504}{0.16 + 0.47} = 0.24 = 24\%$$

Le nombre de *faux positifs* (environ 11%) est considérable : nous répertorions les sur-corrections, p. ex. « Athalie » qui est incorrectement corrigée en « **Na**thalie », « Empédocle » en « **L'**Empé**s**ocle », « Bourette » (poète) en « Bou**l**ette », « Maratuech » (poète) en « Mar**r**a**k**ech », « Gringoire » (poète) en « Gri**m**oire », « M. de Maucroix » en « M. de **La**croix » etc. Nous soulignons également des cas où les EN sont reconnues dans le fichier non corrigé, mais omises dans le corrigé (silence) : « "Géorgiqucs, 1" » (de « Géorgiques », œuvre de Virgile), « Boirude » (omis de la phrase « Quand Brontin à Boirude adresse ce discours »), « "M. Aubran," » (complètement omise dans la version corrigée), « M. Jules Combarieu » (où « Combarieu » est omise) ou « Énéide ».

Toutefois, `JamSpell` parvient à bien corriger certaines EN (vrais positifs, mais seulement 2%, p. ex. « Térehce » > « Téré**n**ce », « Gahors » > « **C**ahors »). Dans d'autres cas, JamSpell a contribué à l'extraction des EN non reconnues dans le texte non corrigé : dans cette phrase : « Comme le Transylvain le Turc et le Hongrois », l'EN « Transylvain » est corrigée en « Transylva**nie** » et reconnue sous cette forme dans le fichier corrigé seulement. La mention des « Poésies de Valentin » n'existe aussi que dans le texte corrigé.

Les diagrammes à barres sur les Figures 3a et 3b montrent la répartition des types d'EN extraites. Pour avoir un aperçu plus clair sur la répartition des éléments en question, nous avons exclu les vrais négatifs et les tokens incorrectement classifiés comme les EN de la deuxième sous-figure.

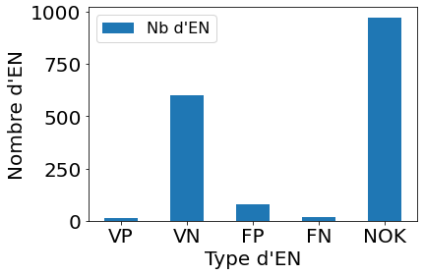

Figure 2a : Diagramme 1

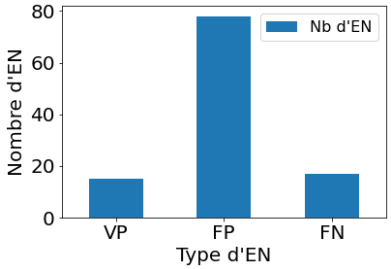

Figure 2b : Diagramme 2

Figure 2 : Répartition des vrais/faux positifs/négatifs et des tokens incorrectement classifiés comme les EN.

Comme dernière remarque, nous observons que les erreurs issues des corrections ne sont pas toujours systématiques (ex. « Pradon » sur-corrigé en « P**ar**don » et « Prado »).

Le nombre conséquent des FP et des FN est dû au fait que le corpus d'entraînement utilisé par `JamSpell` n'est pas suffisamment grand pour englober les EN particulières (p. ex. les noms de poètes ou les noms de lieux mythologiques). Pour atteindre une certaine robustesse, il serait donc nécessaire soit de ré-entraîner `JamSpell` sur un corpus plus adapté (documents historiques), soit de s'appuyer sur une approche hybride proposée précédemment.

Conclusion et perspectives

Dans ce travail, nous avons mené une expérience sur la correction d'orthographe automatique de textes issus de l'OCR, dans l'objectif de mesurer l'impact de la correction sur la tâche de REN. Du point de vue théorique, ce travail de recherche nous a permis de soulever la problématique de fouille des textes bruités dans le cadre de la REN, ainsi que d'établir un état de l'art par rapport aux différentes approches de correction post-OCR (avec leurs avantages et leurs limitations) à des fins d'amélioration de la qualité des documents d'archives numérisées et de l'optimisation de la chaîne de leur traitement numérique. Dans la partie pratique, nous avons d'abord utilisé le modèle `spaCy` *large* de REN pour le français afin d'extraire les EN à partir d'un échantillon non corrigé. Dans un deuxième temps, nous avons réalisé un système implémentant la correction post-OCR à l'aide de la librairie `JamSpell` basée sur des modèles de langue. Nous avons trouvé que le système de repérage d'entités nommées est relativement robuste ; d'autre part, nous montrons que le correcteur orthographique en question peut être amélioré davantage par l'intégration d'une approche hybride qui combine la racinisation d'un mot tenant compte du contexte avec l'identification par poids/ corpus (*rules-based analyzer*). Enfin, d'autres méthodes peuvent être mises en place en vue de l'amélioration du correcteur orthographique, notamment en passant par l'entraînement du modèle sur un corpus plus grand en utilisant des architectures d'apprentissage profond (`NeuSpell`).

Contributions

Ce travail a été financé par l'équipe ObTIC-Sorbonne Université, dans le cadre d'un stage extracurriculaire de Ljudmila Petkovic, sous la direction de Motasem Alrahabi et Glenn Roe, lors de sa réalisation du programme « Certificat de spécialisation en linguistique » au Département de Linguistique à l'université de Genève, Suisse. Toutes les ressources utilisées dans ce travail sont librement accessibles en ligne.46

Index des figures

Index des tableaux

Références

Azmi, A. M., Almutery, M. N., & Aboalsamh, H. A. (2019). Real-Word Errors in Arabic Texts: A Better Algorithm for Detection and Correction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27*(8), 1308-1320. https://doi.org/10.1109/TASLP.2019.2918404

Bassil, Y., & Alwani, M. (2012). OCR Post-Processing Error Correction Algorithm using Google Online Spelling Suggestion. *arXiv preprint arXiv:1204.0191*. https://doi.org/10.48550/arXiv.1204.0191

Bolioli, A., Marchioni, E., & Ventaglio, R. (2014). Errori di OCR e riconoscimento di entità nell'Archivio Storico de La Stampa. In R. Basili, A. Lenci & B. Magnini (Eds.). *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & of the Fourth International Workshop EVALITA 2014 : 9-11 December 2014, Pisa* (pp. 78-82). Pisa University Press. http://digital.casalini.it/3043518

Boros, E., Hamdi, A., Pontes, E. L., Cabrera-Diego, L. A., Moreno, J. G., Sidere, N., & Doucet, A. (2021). Atténuer les erreurs de numérisation dans la reconnaissance d'entités nommées pour les

[46] Cf. https://github.com/ljpetkovic/corr_OCR_NER.

documents historiques. *Conférence en Recherche d'Informations et Applications (CORIA 2021)*, 1-7. https://doi.org/10.24348/coria.2021.mini_24

Bouzidi, K., Elloumi, Z., Besacier, L., Lecouteux, B., & Benzeghiba, M.-F. (2017). Traitement des Mots Hors Vocabulaire pour la Traduction Automatique de Document OCRisés en Arabe. *Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 - Articles longs*, 63-76. https://aclanthology.org/2017.jeptalnrecital-long.5

Chagué, A. (2021). *Comment faire lire des gribouillis à mon ordinateur?* [Diapositives]. ALMAnaCH - Inria Paris (ALMAnaCH). https://hal.archives-ouvertes.fr/hal-03170345

Cordova, J. M., Dupont, Y., Petkovic, L., Gawley, J., Alrahabi, M., & Roe, G. (2022). Toolbox : une chaîne de traitement de corpus pour les humanités numériques. *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 3 : Démonstrations*, 12-14. https://aclanthology.org/2022.jeptalnrecital-demo.4

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM, 7*(3), 171-176. https://doi.org/10.1145/363958.363994

de Amorim, R. C., & Zampieri, M. (2013). Effective Spell Checking Methods Using Clustering Algorithms. *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, 172-178. https://aclanthology.org/R13-1023

Edwards, L. D. M. (2016). Conception de formes de relecture dans les chaînes éditoriales numériques [Thèse de doctorat, Université de Technologie de Compiègne]. https://theses.hal.science/tel-01562039

Enguehard, C., & Mbodj, C. (2004). Des correcteurs orthographiques pour les langues africaines. *Bulletin de linguistique appliquée et générale*, 51-68. https://hal.archives-ouvertes.fr/hal-01094941

Gabay, S. & SciCoS (SCIentific COmputing Support). (2022). FoNDUE (FOrmes Numerisées et Détection Unifiée des Écritures) – An Handwritting Text Recognition infrastructure for Geneva [Diapositives]. Université de Genève. https://datascience.unige.ch/application/files/2316/3248/7621/3.3._Simon_Gabay__Jean-Luc_Falcone.pdf

Hamdi, A., Pontes, E. L., Sidere, N., Coustaty, M., & Doucet, A. (2022). In-depth analysis of the impact of OCR errors on named entity recognition and linking. *Natural Language Engineering*, 1-24. https://doi.org/10.1017/S1351324922000110

Huynh, V.-N., Hamdi, A., & Doucet, A. (2020). When to Use OCR Post-correction for Named Entity Recognition? In E. Ishita, N. L. S. Pang, & L. Zhou (Eds.), *International Conference on Asian Digital Libraries (ICADL 2020): Digital Libraries at Times of Massive Societal Transition. Lecture Notes in Computer Science, Vol. 12504* (pp. 33-42). Springer, Cham. https://doi.org/10.1007/978-3-030-64452-9_3

Issam, B. (2018). Recherche d'information parlée [Thèse de doctorat, Université Badji-Mokhtar-Annaba]. https://biblio.univ-annaba.dz/wp-content/uploads/2019/10/These-Bendib-Issam.pdf.

Jayanthi, S. M., Pruthi, D., & Neubig, G. (2020). NeuSpell: A Neural Spelling Correction Toolkit. *arXiv preprint arXiv:2010.11085*. https://doi.org/10.48550/arXiv.2010.11085

Kolajo, T., Daramola, O., & Adebiyi, A. A. (2022). Real-time event detection in social media streams through semantic analysis of noisy terms. *Journal of Big Data, 9*(1), 1-36. https://doi.org/10.1186/s40537-022-00642-y

Koudoro-Parfait, C., Lejeune, G., & Buth, R. (2022). Reconnaissance d'entités nommées sur des sorties OCR bruitées : des pistes pour la désambiguïsation morphologique automatique. In Y. Estève, T. Jiménez, T. Parcollet, & M. Zanon Boito (Eds.), *Traitement Automatique des Langues Naturelles* (pp. 45-55). ATALA. https://hal.archives-ouvertes.fr/hal-03701476

Koudoro-Parfait, C., Lejeune, G., & Roe, G. (2021). Spatial Named Entity Recognition in Literary Texts : What is the Influence of OCR Noise? *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geospatial Humanities*, 13-21. https://doi.org/10.1145/3486187.3490206

Krstev, C., & Stanković, R. (2020). Old or new, we repair, adjust and alter (texts). *Infotheca - Journal For Digital Humanities, 19*(2), 61-80. https://doi.org/10.18485/infotheca.2019.19.2.3

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics Doklady, 10*(8), 707-710. https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf

Liu, Y., Alzahrani, I. R., Jaleel, R. A., & Al Sulaie, S. (2023). An efficient smart data mining framework based cloud internet of things for developing artificial intelligence of marketing information analysis. *Information Processing & Management, 60*(1), 103-121. https://doi.org/10.1016/j.ipm.2022.v

Mubarak, H., & Darwish, K. (2014). Automatic Correction of Arabic Text: a Cascaded Approach. *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, 132-136. https://doi.org/10.3115/v1/W14-3617

Nguyen, T. T. H., Jatowt, A., Coustaty, M., & Doucet, A. (2021). Survey of Post-OCR Processing Approaches. *ACM Computing Surveys, 54*(6), 1-37. https://doi.org/10.1145/3453476

Nguyen, T. T. H., Jatowt, A., Nguyen, N.-V., Coustaty, M., & Doucet, A. (2020). Neural Machine Translation with BERT for Post-OCR Error Detection and Correction. Proceed*ings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, 333-336. https://doi.org/10.1145/3383583.3398605

Oger, S., Rouvier, M., Camelin, N., Kessler, R., Lefèvre, F., & Torres-Moreno, J.-M. (2012). Système du LIA pour la campagne DEFT2010. In Grouin, C. & Forest, D. (Eds.), *Expérimentations et évaluations en fouille de textes : Un panorama des campagnes DEFT* (Chapitre 9). Coll. Systèmes d'information et organisations documentaires, Hermès. https://hal.archives-ouvertes.fr/hal-01433469

Ratcliff, J. W., & Metzener, D. E. (1988). Pattern-matching-the gestalt approach. *Dr Dobbs Journal, 13*(7), 46. https://www.drdobbs.com/database/pattern-matching-the-gestalt-approach/184407970?pgno=5.

Reul, C., Christ, D., Hartelt, A., Balbach, N., Wehner, M., Springmann, U., Wick, C., Grundig, C., Büttner, A., & Puppe, F. (2019). OCR4all—An Open-Source Tool Providing a (Semi-)Automatic OCR Workflow for Historical Printings. *Applied Sciences, 9*(22), 4853, 1-30. https://doi.org/10.3390/app9224853

Reul, C., Springmann, U., Wick, C., & Puppe, F. (2018). State of the Art Optical Character Recognition of 19th Century Fraktur Scripts using Open Source Engines. *arXiv preprint arXiv:1810.03436*. https://doi.org/10.48550/arXiv.1810.03436

Reynaert, M. (2008). All, and only, the errors : more complete and consistent spelling and ocr-error correction evaluation. Dans Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), 1867-1872 https://aclanthology.org/L08-1217/.

Reynaert, M. (2008). All, and only, the Errors: more Complete and Consistent Spelling and OCR-Error Correction Evaluation. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. LREC 2008, Marrakech, Morocco. http://www.lrec-conf.org/proceedings/lrec2008/pdf/477_paper.pdf

Rigaud, C., Doucet, A., Coustaty, M., & Moreux, J.-P. (2019). ICDAR 2019 Competition on Post-OCR Text Correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (1588-1593). IEEE. https://doi.org/10.1109/ICDAR.2019.00255

Rijhwani, S., Anastasopoulos, A., & Neubig, G. (2020). OCR Post Correction for Endangered Language Texts. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5931-5942. https://doi.org/10.18653/v1/2020.emnlp-main.478

Sagot, B., & Gábor, K. (2014). Détection et correction automatique d'entités nommées dans des corpus OCRisés. *Actes de la 21e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2014), Marseille (France)*, 437-442. http://talnarchives.atala.org/TALN/TALN-2014/taln-2014-court-009.pdf

Salifou, L., & Naroua, H. (2014). Étude et conception d'un correcteur orthographique pour la langue haoussa. *TALN-RECITAL 2014 Workshop TALAf 2014 : Traitement Automatique des Langues Africaines (TALAf 2014: African Language Processing)*, 147-158. https://aclanthology.org/W14-6506

Tantini, F., Terlutte, A., & Torre, F. (2011). Combinaisons d'automates et de boules de mots pour la classification de séquences. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle*, *25*(3), 411-434. https://hal.inria.fr/hal-00643057

Volk, M., Furrer, L., & Sennrich, R. (2011). Strategies for Reducing and Correcting OCR Errors. In C. Sporleder, A. van den Bosch, & K. Zervanou (Eds.), *Language Technology for Cultural Heritage. Theory and Applications of Natural Language Processing* (pp. 3-22). Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-20227-8_1

Wei, C., Wang, B., & Kuo, C. C. J. (2022). SynWMD: Syntax-aware Word Mover's Distance for Sentence Similarity Evaluation. arXiv preprint arXiv:2206.10029. https://doi.org/10.48550/arXiv.2206.10029

Wisniewski, G., Max, A., & Yvon, F. (2010). Recueil et analyse d'un corpus écologique de corrections orthographiques extrait des révisions de Wikipédia. *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, 121-130. https://aclanthology.org/2010.jeptalnrecital-long.13