Évaluation de méthodes et d'outils pour la lemmatisation automatique du français médiéval

Cristina Holgado, Alexei Lavrentiev, Mathieu Constant

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Évaluation de méthodes et d'outils pour la lemmatisation automatique du français médiéval

Cristina G. Holgado[1] Alexei Lavrentev[2] Mathieu Constant[3]
(1) Université de Strasbourg, F-67081, Strasbourg
(2) IHRIM, CNRS, ENS de Lyon, F-69007 Lyon, France
(3) ATILF, Université de Lorraine, CNRS, F-54063, Nancy
cristina.gholgado@gmail.com, alexei-lavrentev@ens-lyon.fr,
mathieu.constant@univ-lorraine.fr

RÉSUMÉ

Pour les langues historiques non stabilisées comme le français médiéval, la lemmatisation automatique présente toujours des défis, car cette langue connaît une forte variation graphique. Dans cet article, nous dressons un état des lieux de la lemmatisation automatique pour cette langue en comparant les performances de quatre lemmatiseurs existants sur un même jeu de données. L'objectif est d'évaluer où se situent les nouvelles techniques de l'apprentissage automatique par rapport aux techniques plus traditionnelles s'appuyant sur des systèmes de règles et lexiques, en particulier pour la prédiction des mots inconnus.

ABSTRACT

Evaluation of methods and tools for automatic lemmatization in Old French.

For non-stabilized historical languages such as old French, automatic lemmatization still presents challenges, as this language has a strong graphic variation. In this article we benchmark automatic lemmatization for this language by comparing the performances of four existing lemmatizers on the same dataset. Our goal is to evaluate where the new techniques of machine learning stand in regards to more traditional rule- and lexicon-based ones, especially for unknown words.

MOTS-CLÉS : lemmatisation, étiquetage morphosyntaxique, linguistique historique, français médiéval.

KEYWORDS: lemmatization, part-of-speech tagging, historic linguistics, Old French.

1 Introduction

La lemmatisation automatique a été l'objet de très nombreuses recherches dans le domaine du traitement automatique des langues (TAL). Elle consiste à produire, pour chaque occurrence de mots, leur forme de base telle qu'on peut la trouver dans des dictionnaires. Les nombreux outils qui sont désormais disponibles ont permis de populariser cette tâche pour un nombre important de langues.

Dans cet article, nous nous intéressons à la lemmatisation automatique du français médiéval qui fait face à de nombreux défis. L'absence de normalisation graphique rend la lemmatisation pour une forme plus difficile car elle peut apparaître sous plusieurs graphies. Le français médiéval se caractérise par une complexité morphologique plus importante que le français moderne et il est particulièrement

Volume 1 : conférence principale, pages 153–161.

marqué par la variation dialectale. Plusieurs corpus textuels lemmatisés, dictionnaires électroniques et lexiques morphologiques associés existent pour cet état de langue (Lavrentiev *et al.*, 2017). Cependant différents dictionnaires de référence utilisent parfois différentes formes d'entrée pour le même lexème, puisque certains privilégient les formes modernisées et d'autres conservent les formes médiévales. Il est par conséquent nécessaire de normaliser les lemmes avant de compiler un corpus d'apprentissage unifié.

Nous souhaitons dresser un état des lieux de la lemmatisation automatique du français médiéval en comparant différents outils sur un même ensemble de textes. Il existe deux grandes familles d'approches pour la lemmatisation : (1) les approches traditionnelles se fondant sur des lexiques et des règles (Silberztein, 1994; Beesley & Karttunen, 2003), (2) les approches reposant sur des techniques d'apprentissage automatique à partir de corpus annotés (Chrupala *et al.*, 2008; Müller *et al.*, 2015; Bergmanis & Goldwater, 2018), parfois complémentés de lexiques. Nous cherchons ici à évaluer où se situent les nouvelles techniques de l'apprentissage automatique par rapport aux techniques plus traditionnelles et éprouvées dans le cadre de la lemmatisation du français médiéval. Plus précisément, nous allons tester quatre outils : (1) TreeTagger (Schmid, 1994) ; (2) LGeRM (Souvay & Pierrel, 2009) ; (3) Pie (Manjavacas *et al.*, 2019) ; (4) UDPipe (Straka & Straková, 2017). Les trois premiers sont très utilisés pour la lemmatisation du français médiéval. Ils utilisent des approches assez différentes donc ils sont intéressants à comparer. Le dernier est un outil très populaire dans la communauté TAL actuelle, utilisant une méthode de lemmatisation encore différente des trois premiers.

L'article est organisé comme suit. Tout d'abord, nous présentons les outils utilisés. Ensuite, nous précisons les caractéristiques du corpus et la procédure expérimentale. Enfin, nous proposons une discussion des résultats et des pistes pour améliorer la lemmatisation du français médiéval.

2 Description des lemmatiseurs utilisés

Nous décrivons maintenant les quatre outils de lemmatisation testés dans l'article pour le français médiéval. Les outils TreeTagger et LGeRM sont des systèmes à base de règles et de lexiques pour la lemmatisation. TreeTagger est, avant tout, un étiqueteur morphosyntaxique basé sur des arbres de décision appris à partir d'un corpus annoté. Il existe un module de lemmatisation qui consiste à récupérer le (ou les) lemme(s) du mot en entrée dans le lexique du corpus d'apprentissage et potentiellement d'un lexique externe. La prédiction de l'étiquette grammaticale permet de filtrer les lemmes du lexique qui ne sont pas de cette catégorie. Pour les mots inconnus du lexique, si l'option correspondante est activée, le lemme prédit correspond simplement à la copie du mot en entrée. La version récente de LGeRM repose sur un principe similaire à quelques différences notables près. Tout d'abord, il utilise TreeTagger pour prédire l'étiquette morphosyntaxique du mot à lemmatiser. Ensuite, il repose sur le lexique extrait (principalement) du Dictionnaire du Moyen Français (DMF) (Bazin-Tacchella *et al.*, 2016). Enfin, un système de règles complexes est appliqué pour prédire le (ou les) lemme(s) des mots inconnus du lexique. Les outils Pie et UDPipe s'appuient sur un apprentissage supervisé de leurs modèles de lemmatisation à partir de corpus annotés. Pie utilise un modèle neuronal encodeur-décodeur appris conjointement avec une tâche de prédiction des mots suivants et précédents afin de mieux tenir compte du contexte, sans avoir à prédire l'étiquette morphosyntaxique. L'outil, incluant également un étiqueteur morphosyntaxique, a démontré des résultats très prometteurs pour la lemmatisation des états anciens de différentes langues. UDPipe, quant à lui, incorpore un autre

type de lemmatiseur appris sur corpus annoté. Étant donné un mot à lemmatiser, le principe est de générer un ensemble de paires (lemme, étiquette morphosyntaxique) possibles à l'aide de règles de lemmatisation apprises automatiquement, puis d'appliquer un modèle de levée d'ambiguïté appris grâce à un perceptron moyenné.

3 Campagne de tests

3.1 Source des données et normalisation

Les textes annotés utilisés à l'entraînement et à l'évaluation sont issus du corpus BFMGOLDLEM rassemblant un total de 431 144 formes étiquetées et lemmatisées. Ce corpus fait partie de la BFM (Base de Français Médiéval) [1] (Guillot *et al.*, 2018), une collection de textes médiévaux qui recouvre la période du IX^{e} jusqu'au XV^{e} siècle et dont le nombre total d'occurrences-mots s'élève à 4,7 millions. Il est composé de deux sources dont une prédominante appartient à un seul auteur (Chrétien de Troyes). Il a été lemmatisé à l'ATILF dans le cadre du projet DECT (Souvay & Kunstmann, 2008). C'est donc un corpus important, mais peu diversifié (un seul auteur, un seul manuscrit, un seul genre). Il a son propre référentiel de lemmes, qui correspondent pour la plupart aux entrées du dictionnaire Tobler-Lommatzsch (TL) (Adolf, 2002) qui privilégie des formes anciennes. Le reste du corpus a été lemmatisé dans le cadre de la BFM et est beaucoup plus diversifié. Il est au format CONLL-U (Nivre *et al.*, 2016), et constitué par les formes fléchies tokenisées, accompagnées des étiquettes morphologiques du jeu d'étiquettes Cattex09 (Prévost *et al.*, 2009) et des lemmes. Ces lemmes sont majoritairement issus du DMF, qui utilise des formes modernes. D'autres référentiels ont été utilisés en cas d'absence de lemme dans le DMF.

Afin d'harmoniser dans la mesure du possible les lemmes dans le corpus d'apprentissage, les lemmes DECT dans les textes de Chrétien de Troyes, ont été convertis en lemmes d'autres référentiels à partir du lexique FROLEX (Lavrentiev *et al.*, 2017) selon le procédé suivant : conversion vers un lemme DMF (s'il existe), sinon TL, sinon GDF (Godefroy, 1901). Le lemme DECT a été conservé en cas d'absence de correspondance. Dans le corpus normalisé, les 424 836 lemmes sont ainsi des lemmes DMF (98,54%), 4 512 DECT (1,04%), 965 BFM (0,22%) [2], 801 TL (0,18%) et 30 GDF (0,01%).

3.2 Protocole de tests

Afin d'évaluer les outils, nous avons mis en place un protocole de tests se voulant le plus réaliste possible. Il est divisé en dix expériences d'évaluation incluant des corpus de textes de caractéristiques linguistiques et tailles différentes, dont certains sont plus marqués dialectalement et/ou datant d'états plus anciens de la langue, ce qui arrive effectivement dans la pratique. Ainsi, notre corpus a été divisé en dix parties, chacune contenant un ou plusieurs textes cohérents en termes de date, de genre et de dialecte. Pour chaque expérience, les outils (étiquetage et/ou lemmatisation) étaient appris sur tout le corpus, sauf la partie testée (que l'on appelle corpus de contrôle). Le découpage est indiqué dans le tableau 2. Le tableau 1 indique les caractéristiques des textes de contrôle pour chaque test. Une

1. http://txm.bfm-corpus.org
2. Les lemmes BFM sont essentiellement des noms propres du corpus BFM. Aucun lemme DECT n'a été converti vers ce référentiel.

description plus détaillée du protocole de tests est accessible dans un dépôt GIT du projet [3].

Test	Date	Dialecte	Genre
1 et 2	fin 12e s.	champenois	roman
3	milieu 11e s.	normand	hagiographie
4	début 12e s.	normand	épique
5	milieu 12e s.	anglo-normand	chronique
6	début 14e s.	non marqué	chronique
7	fin 13e s.	non marqué	hagiographie
8	début 11e s.	franco-occitan	hagiographie
9	milieu 13e s.	hainaut	charte
10	fin 14e s.	non marqué	registre

TABLE 1: Métadonnées caractéristiques des textes de contrôle pour chaque test

La première partie (test 1) contient tous les textes de Chrétien de Troyes, ce qui constitue le corpus d'apprentissage avec la taille la plus réduite (41,1%) par rapport aux autres tests, pour lesquels le corpus de contrôle constitue entre 1,2% et 8,2% du corpus total en nombre de tokens. Cela permet d'évaluer l'impact du retrait du corpus d'apprentissage d'un volume de textes important mais très homogène. Dans le test 2, un seul texte de Chrétien de Troyes constitue le corpus de contrôle, les autres textes de cet auteur font partie du corpus d'apprentissage. Il convient de noter que la proportion des formes inconnues varie très fortement selon les tests et ce n'est pas dans le premier test qu'elle est la plus élevée (11,40% contre 31,85% dans le test 8 où le corpus de contrôle est composé de deux textes courts, mais très anciens et marqués dialectalement). Dans les tests 3 et 4, les textes de contrôle sont également très anciens, mais les graphies sont moins inhabituelles. Dans les tests 9 et 10, les textes de contrôle ne sont pas marqués sur le plan diachronique ou dialectal, mais appartiennent à un genre peu représenté dans le corpus d'apprentissage (acte juridique).

3.3 Configuration des outils

TreeTagger a été utilisé sans lexique externe. Cela implique que le lexique utilisé est celui du corpus d'apprentissage. Nous utilisons la dernière version de LGeRM qui nous a été fourni par le développeur de l'outil. Cette version repose sur l'étiqueteur TreeTagger pré-appris. Pour UDPipe, nous avons utilisé la bibliothèque associée dans R (UDPipe v0.8.3). L'entraînement du lemmatiseur est appris en utilisant 60 itérations (epochs), un batch de taille 100 et un taux d'apprentissage (learning rate) de 0,1. L'étiqueteur est appris en utilisant 20 itérations. Les autres paramètres de configuration correspondent aux valeurs par défaut. Ceux de Pie (v0.8.5) ont été fixés de la manière suivante : 50 itérations (epochs) et un batch de taille 50 (lemmatiseur) et 10 itérations et un batch de taille 50 (étiqueteur). Le lemmatiseur et l'étiqueteur utilisent tous deux un taux d'apprentissage de 0,0001.

4 Analyse des résultats

Le tableau 2 affiche la précision moyenne (micro) obtenue pour chaque outil sur l'ensemble des tests.

3. https://gitlab.huma-num.fr/lemmatisation-fro/bfm-lem

La ponctuation n'est pas prise en compte dans le calcul. Il comprend, d'une part, la précision obtenue sur l'ensemble des lemmes et, d'autre part, celle obtenue pour les mots inconnus, c'est-à-dire, les unités absentes du corpus d'apprentissage (CA). Du fait que LGeRM peut proposer plusieurs lemmes, il a été évalué différemment en divisant le nombre de lemmes corrects par le nombre de lemmes proposés pour la forme. Ses résultats sur les mots inconnus doivent être regardés avec précaution car l'outil ne s'appuie pas sur le corpus d'apprentissage mais sur un lexique externe. Ainsi, l'évaluation des ces mots inclut des mots présents dans son lexique.

T	**CA**		**CC**			**TreeTagger**		**LGeRM**		**UDPipe**		**Pie**	
	tokens	%	tokens	m.inc.	%	tout	inc.	tout	inc.	tout	inc.	tout	inc.
1	177 050	41,1	254 094	28 976	11,4	0,75	0,07	0,83	0,82	0,67	0,12	0,74	**0,36**
2	383 164	88,9	47 965	1275	2,6	**0,86**	0,09	0,83	**0,84**	**0,76**	0,19	0,71	0,22
3	425614	98,7	5530	770	13,9	0,73	0,07	0,78	0,56	0,65	0,09	0,60	0,21
4	395 832	91,8	35 312	5399	15,3	0,72	0,07	0,85	0,75	0,70	0,13	0,61	0,20
5	413 123	95,8	18 021	2313	18,8	0,77	0,10	0,86	0,63	0,69	0,12	0,69	0,26
6	420 109	97,4	11 035	1405	12,7	0,81	0,20	**0,90**	0,71	0,71	0,20	0,69	0,23
7	408 375	94,7	22 769	2008	8,81	0,79	0,12	0,89	0,77	0,73	0,20	**0,75**	0,31
8	426 052	98,8	5092	1622	31,8	0,43	0,02	0,61	0,39	0,42	0,06	0,45	0,16
9	420 652	97,6	10 492	1711	16,3	0,74	0,13	0,82	0,54	0,68	0,09	0,67	0,16
10	419 163	97,2	11 981	2380	19,9	0,76	**0,29**	**0,90**	0,77	0,64	**0,21**	0,66	0,17
	Moyenne					0,74	0,12	0,83	0,68	0,66	0,14	0,66	0,23

TABLE 2: Découpage des expériences (CA corpus d'apprentissage ; CC corpus de contrôle). Moyenne des tests (hors ponctuation) (précision)

Nous observons que les meilleurs résultats coïncident pour les quatre outils sur un groupe spécifique de tests, en particulier les tests 2, 6 et 7. Les tests 2 et 7 correspondent effectivement aux textes contenant le plus petit pourcentage de mots inconnus. En ce qui concerne l'ensemble des tokens (tout), LGeRM obtient le meilleur résultat parmi l'ensemble des outils avec une précision moyenne de 83%. Ceci s'explique en partie du fait qu'il possède un lexique très riche. Néanmoins, il est principalement conçu pour le Moyen Français ce qui explique la faible précision pour les tests 3 et 8 constitués de textes très anciens en français médiéval, et la précision la plus élevée pour les tests 6 et 10, constitués de textes plus tardifs. D'autre part, les autres outils ont été entrainés avec le même jeu de données en l'absence de lexique externe. De ce groupe, TreeTagger a atteint la précision la plus élevée avec 74% pour les lemmes, suivi par Pie avec 66% et UDPipe avec 64%. Dans le cas de TreeTagger, la taille du CA a un impact sur sa performance, ce qui n'est pas le cas dans UDPipe et Pie, comme cela est illustré dans les tests 1, 2 et 7, si nous tenons compte du pourcentage relativement faible de mots inconnus. En effet, le test 4 comporte un CA plus réduit que le test 7 mais le nombre de mots inconnus dans le CC est plus élevé. De ce fait, nous constatons que le nombre de mots inconnus a, en définitive, un impact sur l'ensemble des outils. Cependant, dans les outils UDPipe et Pie, et au contraire de TreeTagger, la taille des données n'est pas forcément significative pour l'amélioration de la précision mais plutôt le nombre assez représentatif d'échantillons dans les catégories pour obtenir une meilleure modélisation, permettant à ces outils d'être capables de prédire un lemme lorsqu'ils rencontrent de nouvelles formes. Pie est ainsi plus performant que TreeTagger dans le test 8 qui est caractérisé par deux textes très anciens et marqués dialectalement dont les tokens ne sont pas présents dans le corpus d'apprentissage. Afin d'examiner plus en détail ces résultats et mieux comprendre les performances des lemmatiseurs pour les mots inconnus, nous avons calculé la précision des lemmes pour chacune des étiquettes du corpus (cf. tableau 3).

Cat.	Tokens	%	m.inc.	%	TreeTagger		LGeRM		UDPipe		Pie	
					tout	inc.	tout	inc.	tout	inc.	tout	inc.
ADJ	14 773	4,03	2680	5,60	0,73	0,10	0,83	0,67	0,65	0,11	0,55	0,18
ADV	39 535	10,78	2435	5,09	0,71	0,10	0,63	0,75	0,81	0,13	0,62	0,18
CON	37 233	10,15	44	0,09	0,82	0,00	0,94	0,37	0,94	0,02	0,77	0,44
DET	35 853	9,77	812	1,70	0,68	0,06	0,81	0,55	0,72	0,05	0,65	0,15
Ncom	53 989	14,72	12 649	26,43	0,68	0,12	0,71	0,68	0,51	0,14	0,50	0,20
Npro	9268	2,53	6058	12,66	0,54	0,40	0,43	0,47	0,34	0,30	0,26	0,03
PRE	34 309	9,35	667	1,39	0,66	0,08	0,80	0,60	0,77	0,18	0,59	0,12
PRO	60 870	16,59	770	1,61	0,72	0,01	0,75	0,60	0,67	0,06	0,57	0,15
VER	80 522	21,95	21 413	44,74	0,62	0,02	0,84	0,80	0,57	0,13	0,60	0,36
Total	366 882		47 859									

TABLE 3: Précision des lemmes par catégorie

Tout d'abord, nous constatons que les mots inconnus appartiennent aux parties du discours qui possèdent le plus grand nombre de formes fléchies (d'abord, les verbes, puis les noms communs et les noms propres). Les performances relativement élevées de LGeRM s'expliquent par l'importance de son lexique (indépendant du corpus d'apprentissage) et le fait que les règles de substitution permettent le plus souvent de retrouver une forme attestée dans le lexique. TreeTagger prend la forme pour lemme en cas de mots inconnus lorsqu'on active l'option « -no-unknown ». Cette stratégie est assez efficace pour les noms propres, mais elle échoue systématiquement pour les verbes (à l'exception des infinitifs). Ainsi, il gagne en précision dans le cas des mots inconnus si le token et le lemme sont identiques (e.g. : le nom propre *Rome*), ce qui est rarement le cas des verbes. Ces résultats limités pour les mots inconnus peuvent également s'expliquer par le fait que, dans la très grande majorité des cas, les lemmes de référence sont modernisés, ce qui réduit encore les chances que la forme d'un mot soit identique au lemme. En particulier, l'outil obtient un score nul sur les conjonctions inconnues : ex. les formes *maiz* et *conbien* ont respectivement pour lemmes *mais* et *combien*.

Au contraire, UDPipe et Pie sont moins affectés par le nombre de formes fléchies et plus performants que TreeTagger pour quasiment toutes les catégories. Selon les résultats obtenus par ces deux outils, de manière générale, plus le nombre de tokens augmente, plus la précision augmente. Cela s'explique en partie par la capacité "générative" de ces deux outils qui sont capables de prédire une paire (forme, lemme) qui n'aura jamais été vue dans le corpus d'apprentissage (cf. section 2). La quantité d'exemples donnés à l'apprentissage est cruciale dans ce cadre-là pour que les procédures d'apprentissage puisse en extraire automatiquement des généralisations. Par exemple, dans le test 10, le pronom inconnu *luy* et le verbe inconnu *deposeroit* sont correctement lemmatisés en *lui* et *déposer* par Pie. Dans ce même test, le nom propre inconnu *Yvein* et l'adverbe inconnu *Meïsmes* sont correctement lemmatisés en *Yvain* et *même* pour UDPipe.

Pie est, en général, plus performant que UDPipe sur les mots inconnus, à l'exception toutefois des noms propres et des prépositions. Pie obtient ses performances les plus notables sur les conjonctions (44% de précision) et les verbes (36%). UDPipe est particulièrement performant sur les noms propres (27%). La différence de performances entre les deux outils est particulièrement marquante sur les conjonctions et les pronoms en faveur de Pie. Les résultats relativement plus élevés dans ces catégories s'expliquent par le fait que la plupart des tokens correspondent à un nombre très limité de types (e.g. conjonctions 'qua', 'comme', 'mais'). Certains mots grammaticaux courts et fréquents semblent avoir un impact important sur les analyses UDPipe. Par exemple, la contraction

‘du’ PRE.DETdef/de.le influence l’analyse de plusieurs formes qui se terminent en -du. Par exemple, *desfandu* : PRE.DETdef/*desfande.le*. Pie propose, dans ce cas, l’analyse correcte : VERppe/*défendre*. Une liste de catégories fermées fournie à l’outil en tant que paramètre aurait permis d’éviter ce genre d’erreur. Inversement, les noms propres obtiennent une précision particulièrement faible dans Pie relativement à UDPipe, en particulier pour les formes inconnues (3% vs. 27%).

Concernant Pie, la lemmatisation et l’étiquetage sont indépendants (cf. section 2). Cet outil produit donc parfois des analyses incohérentes du point de vue linguistique. Par exemple, pour la forme *Alexis*, l’étiquette ‘NOMpro’ est correcte, mais le lemme proposé *Aller* ressemble à un infinitif. La même situation se produit pour la forme *amfant* : NOMcom/*ampendre* (lemme correct : *enfant*). UDPipe, qui sélectionne des paires (lemme, étiquette), propose l’analyse VERppa/*amfer*, qui est erronée, mais cohérente. Nous indiquons, dans le tableau 4, quelques exemples d’erreurs de UDPipe et Pie dans la prédiction des lemmes pour des formes inconnues.

Forme	**Lemme et étiquette prédits**		**Lemme et étiquette gold**		**Outil**	**Type d’erreur**
Jehan	*Jehan*	NOMpro	*Jean*	NOMpro	Pie	Forme = lemme
Choisy	*Choisy*	NOMpro	*Choisu*	NOMpro	UDPipe	-y transformé en -u
Caisnoit	*Caisnir*	VER	*Quesnoy*	NOMpro	UDPipe	Mauvaise étiquette
dudit	*dudoulir*	VERinf	*de.ledit*	PRE.DET	Pie	Mauvaise étiquette
en.ii.	*en.ii*	DETcar	*ambedeux*	DETcar	UDPipe	Cas spécifique
Berthier	*Berthier*	NOMpro	*Berter*	VERinf	Pie	Mauvaise étiquette
Alexis	*Aller*	NOMpro	*Alexis*	NOMpro	Pie	Incohérence linguistique

TABLE 4: Quelques exemples d’erreurs sur les mots inconnus.

5 Conclusion

Dans cet article, nous avons évalué quatre outils de lemmatisation pour le français médiéval, utilisant différentes méthodes. Les résultats expérimentaux ont montré que les approches fondées sur des lexiques et systèmes de règles étaient les plus performantes du fait du manque de couverture des corpus annotés pour les méthodes supervisées. Néanmoins, les résultats sur les mots inconnus montrent que les approches par apprentissage automatique sont bénéfiques. Ces conclusions ouvrent de nombreuses pistes pour améliorer les performances. La richesse du lexique morphologique reste à ce jour le facteur déterminant dans la réussite de la lemmatisation. La première piste serait donc d’utiliser le lexique et les règles de LGeRM en association non pas avec TreeTagger, mais avec les étiqueteurs plus récents. L’augmentation du corpus d’apprentissage avec une meilleure représentation des périodes et des dialectes du français médiéval devrait également avoir un impact positif sur les performances de ces outils. L’utilisation simultanée de plusieurs outils et la mise en place d’un système de vote pour sélectionner le lemme et l’étiquette les plus probables est également très prometteuse. Le poids de chaque outil dans le vote devrait être calibré en fonction de ses performances pour chaque catégorie grammaticale. Enfin, des réglages et des post-traitements spécifiques pourraient être appliqués à chaque outil.

Remerciements

Le travail décrit dans cet article a été financé par l’Agence Nationale de la Recherche, via le projet PROFITEROLE (ANR-16-CE38-0010).

Références

ADOLF T. (2002). *Tobler-Lommatzsch : Altfranzôsisches Wôrterbuch*. Wiesbaden : Franz Steiner Verlag.

BAZIN-TACCHELLA S., MARTIN R. & SOUVAY G. (2016). *DMF 2015 - Dictionnaire du Moyen Français (version 2015)*. ATILF.

BEESLEY K. R. & KARTTUNEN L. (2003). *Finite State Morphology*. CSLI Publications. Google-Books-ID : 59RoAAAAIAAJ.

BERGMANIS T. & GOLDWATER S. (2018). Context Sensitive Neural Lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1391–1400, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1126.

CHRUPALA G., DINU G. & VAN GENABITH J. (2008). Learning Morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco : European Language Resources Association (ELRA).

GODEFROY F. (1901). *Lexique de l'ancien français*. H. Welter.

GUILLOT C., HEIDEN S. & LAVRENTIEV A. (2018). Base de français médiéval : une base de référence de sources médiévales ouverte et libre au service de la communauté scientifique. *Diachroniques. Revue de Linguistique française diachronique*, **7**, 168–184. Publisher : Presses de l'Université Paris-Sorbonne (PUPS).

LAVRENTIEV A., HEIDEN S. & DECORDE M. (2017). Building an Open Morphological Lexicon and Lemmatizing Old French Texts with the TXM Platform. In *Corpus linguistics - 2017*, Proceedings of the international conference "Corpus linguistics - 2017", p. 48–52, St-Pétersbourg, Russia : St-Petersburg State University and Institute for Linguistic Studies (RAS) and Herzen State Pedagogical University of Russia.

MANJAVACAS E., KÁDÁR A. & KESTEMONT M. (2019). Improving Lemmatization of Non-Standard Languages with Joint Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 1493–1503, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1153.

MÜLLER T., COTTERELL R., FRASER A. & SCHÜTZE H. (2015). Joint Lemmatization and Morphological Tagging with Lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2268–2274, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1272.

NIVRE J., DE MARNEFFE M.-C., GINTER F., GOLDBERG Y., HAJIČ J., MANNING C. D., MCDONALD R., PETROV S., PYYSALO S., SILVEIRA N., TSARFATY R. & ZEMAN D. (2016). Universal Dependencies v1 : A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, p. 1659–1666, Portorož, Slovenia : European Language Resources Association (ELRA).

PRÉVOST S., GUILLOT C., LAVRENTIEV A. & HEIDEN S. (2009). Jeu d'étiquettes CATTEX 2009.

SCHMID H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, p. 44–49, Manchester, UK : Routledge.

SILBERZTEIN M. (1994). INTEX : a corpus processing system. In *Coling'94. The 15th International Conference on Computational Linguistics. Proceedings*, volume 1, p. 579–583 : Association for Computational Linguistics. DOI : 10.3115/991886.991988.

SOUVAY G. & KUNSTMANN P. (2008). DÉCT (Dictionnaire Électronique de Chrétien de Troyes) : model for today's lexicography ? In *Proceedings of the XIII EURALEX International Congress (Barcelona, 15-19 July 2008), 2008, ISBN 978-84-96742-67-3, págs. 1203-1208*, p. 1203–1208. Section : Proceedings of the XIII EURALEX International Congress (Barcelona, 15-19 July 2008).

SOUVAY G. & PIERREL J.-M. (2009). LGeRM Lemmatisation des mots en Moyen Français. *Traitment automatique des langues*, **50**(2), 149–172.

STRAKA M. & STRAKOVÁ J. (2017). Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 88–99, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/K17-3009.