



HAL
open science

La production de corpus d'occitan médiéval et prémoderne

Jean-Baptiste Camps, Gilles Guilhem Couffignal

► **To cite this version:**

Jean-Baptiste Camps, Gilles Guilhem Couffignal. La production de corpus d'occitan médiéval et prémoderne. Actes du XIIe Congrès de l'Association internationale d'études occitanes Albi, 2017, Association internationale d'études occitanes (AIEO), Jul 2017, Albi, France. halshs-02050089

HAL Id: halshs-02050089

<https://shs.hal.science/halshs-02050089v1>

Submitted on 22 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La production de corpus d’occitan médiéval et prémoderne : problèmes et perspectives de travail*

Jean-Baptiste Camps
Centre Jean-Mabillon
École nationale des chartes | Université PSL

Gilles Guilhem Couffignal
Sorbonne Université | EA 4509, STIH

Actes du XII^e Congrès de l’Association internationale d’études occitanes
Albi, 2017

Le chercheur en langue et littérature occitanes est déjà bien bien pourvu en matière de corpus textuels électroniques, de façon même surprenante si l’on considère le statut social de l’occitan dans la société française actuelle. Le médiéviste occitaniste, notamment, a depuis longtemps à sa disposition la précieuse COM, fruit d’un travail pionnier mené en son temps par le regretté Peter Ricketts (Ricketts, Akehurst et al. 2001 ; Ricketts, Reed et al. 2005). D’autres projets de corpus (Field 2016 ; Scrivner et al. 2016) ou d’édition électronique (Lodge 2006 ; Carrasco et al. 2014–) touchant à la période médiévale ont également fleuri ces dernières années.

Pour les études modernes, les projets sont également nombreux et anciens. On peut les faire remonter, dans une perspective computationnelle, aux tables dialectométriques de Séguy 1971¹. Les projets ne manquent pas, ni même les sources déjà traitées et donc exploitables. Devant la disparité des techniques employées, nous voudrions montrer ici comment une équipe restreinte de deux ou trois chercheurs peut concrètement travailler à partir de corpus textuels électroniques. Deux points nous intéresseront particulièrement, l’acquisition et le traitement des données.

Pour l’acquisition des données, nous développerons la question de la reconnaissance optique des caractères (*optical character recognition* ou OCR) et de la reconnaissance des écritures manuscrites (*handwritten text recognition* ou HTR), qui est la technique permettant aujourd’hui à n’importe quel chercheur muni d’un texte numérisé et d’un terminal informatique de créer rapidement son propre corpus, moyennant quelques manipulations informatiques et opérations de transcription que nous détaillerons.

En termes de traitement des données, nous retiendrons l’exemple de la lemmatisation, encore peu souvent effectuée sur les corpus occitans, et pivot d’un grand nombre d’explo-

*Les données et scripts utilisés pour cet article sont disponibles sur Zenodo, <https://zenodo.org/record/XXXX>, doi : .

1. Pour les corpus *stricto sensu*, voir dans ce même volume les projets en cours (Batelòc, CoRLiG) ; on notera également des projets de bibliothèque virtuelle (Népoté et al. 2006).

tations possibles. Ce sera également l'occasion de montrer toute l'importance du partage des données et de formats interoperables entre les différents projets.

1 L'acquisition de corpus : transcription et reconnaissance optique de caractères

Le développement d'outils ouverts de reconnaissance optique des caractères, fondés sur des méthodes d'apprentissage profonds permet de les adapter à la situation d'un imprimé ancien, d'un incunable, voire d'un manuscrit, avec des résultats exploitables (Breuel 2014; Springmann et al. 2015), ce qui autorise la constitution rapide de *corpora* textuels.

La mise en œuvre d'une série de phases d'entraînement, correction, et nouvel entraînement permet à un chercheur de faire progresser la qualité des résultats, en accumulant rapidement des données textuelles brutes. Des outils libres et performants sont désormais à la portée des chercheurs en SHS (Ocropy/CLSTM, Tesseract 4) et permettent d'atteindre des niveaux d'erreur caractère (CER ou *character error rate*) qui peuvent relativement aisément descendre sous les 2% pour des imprimés anciens, et sous les 10% pour des manuscrits médiévaux. La seule limite est que si des modèles existent déjà pour les documents contemporains, il faut le plus souvent les développer soi-même pour les états de langue anciens, voire les adapter à chaque main en ce qui concerne les manuscrits.

1.1 Présentation générale

Le processus de reconnaissance d'un texte (OCR ou HTR) se décompose généralement en plusieurs étapes (fig. 1),

1. le traitement des images : recadrage, redressement, suppression du bruit, et binarisation (en différenciant le texte de l'arrière-plan de l'image) ;
2. l'identification de la mise en page : zones de texte, colonnes, lignes, etc. ;
3. la reconnaissance proprement dite du texte (i.e., la "transcription automatique"), à partir d'un modèle préexistant, ou bien en alternant phase d'entraînements d'un modèle et phase d'accroissement des données (reconnaissance de nouvelles lignes, correction,...) ;
4. d'éventuels post-traitements, visant à améliorer les résultats, en se fondant sur des modèles linguistiques (lexiques, ...) ou en traitant en série des erreurs fréquentes.

La première étape peut être réalisée avec un outil dédié au traitement des images, avec ou sans interface graphique. Certains de ces outils sont d'un maniement très simple pour le néophyte, tels que ScanTailor (Artsimovich et al. 2016). Cette étape n'est pas du tout anodine, et la qualité des résultats que l'on obtiendra en OCR/HTR est directement liée à la qualité des images traitées. Il est donc primordial non seulement d'obtenir des fichiers de bonnes qualité (300 ou 600 points par pouces), mais encore de les rendre les plus lisibles possible : redressement de l'image ; suppression de sources de bruit (automatiquement ou par intervention humaine pour supprimer la décoration ou les partitions musicales, comme pour l'OCR des *Psaumes* d'Arnaud de Salette) ; binarisation, pour accentuer les contrastes et isoler le texte du bruit d'arrière-plan ou des lettres du verso (quand l'encre corrosive les fait apparaître en transparence).

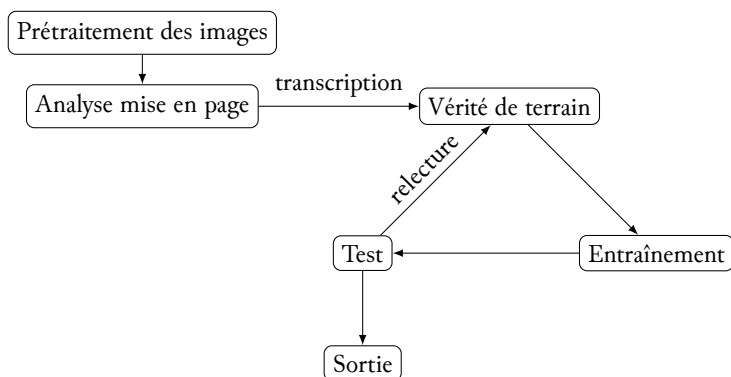


FIGURE 1 – Vue simplifiée des étapes de l'entraînement d'un outil d'OCR/HTR

Enfin, le texte est segmenté par colonne et par ligne, unité de base du traitement de reconnaissance. L'identification de la mise en page, et particulièrement des colonnes et des lignes, peut être plus délicate, notamment à cause des déformations possibles des images, des irrégularités que présentent les manuscrits, ou de la présence d'éléments tels que les lettrines, qui occupent plusieurs lignes de réglure. Pour cette phase d'analyse de la mise en page, une nouvelle génération d'algorithmes, fondés sur de l'apprentissage machine, est en train d'émerger.

Pour ce qui est de la reconnaissance du texte proprement dite, les outils existants peuvent être subdivisés en deux catégories, selon qu'ils aient recours à des méthodes segmentées ou non segmentées (Breuel et al. 2013 ; Ul-Hasan 2016, p. 11-18). Les méthodes segmentées visent tout d'abord à décomposer la ligne en une série de caractères, que l'on tentera ensuite d'identifier en les comparant individuellement à des formes enregistrées pour chaque caractère. Les méthodes non segmentées se fondent sur une analyse de la ligne, sans segmentation préalable, permettant ainsi une meilleure prise en compte du contexte. Elles peuvent, par exemple, faire défiler une fenêtre de largeur fixe sur la ligne en combinaison avec des modèles probabilistes (modèle de Markov caché). Les méthodes non segmentées actuellement les plus prometteuses ont recours à l'apprentissage de séquence, grâce à des réseaux de neurones, et tout particulièrement des réseaux récurrents à mémoire court et long terme (*long short-term memory* ou LSTM), uni- ou bidirectionnels, en combinaison avec un algorithme de classification temporelle connexionniste (CTC) (Breuel et al. 2013 ; Ul-Hasan 2016).

Nous avons ainsi choisi d'utiliser un outil libre implémentant ce dernier type de méthode, OCRopy (Breuel 2014), de pair avec son implémentation en C, CLSTM.

1.2 Un outil libre : OCRopy

OCRopy est un outil *open source*, avec lequel il est aisé d'entraîner et réutiliser des modèles de reconnaissance des caractères ou des écritures. L'entraînement de modèles par apprentissage machine repose sur des données initiales transcrites par l'humain (la vérité de terrain, en anglais, *ground truth*). Concrètement, après avoir pré-traité le document numé-

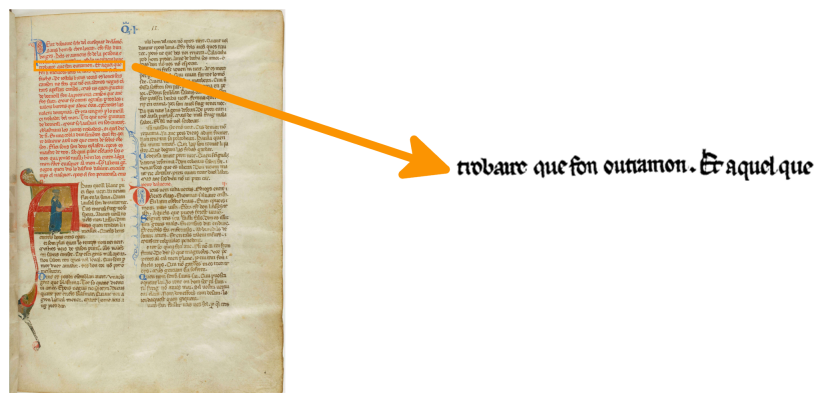


FIGURE 2 – Extractions des lignes binarisées d'un manuscrit (chans. I, BnF fr. 854)

rique de façon à générer un fichier image par ligne de texte, le philologue est tout d'abord invité à transcrire une sélection pertinente (d'un point de vue graphique ou typographique) du texte (fig. 2 et 3).

Cette transcription se fait selon les règles choisies par l'opérateur. Elle peut être de type "diplomatique". Nous préférons les termes d'allographétique et graphématique, suivant la terminologie proposée par Stutzmann 2011. Dans le premier cas chaque allographe est encodé par son propre caractère, tandis que dans le second cas, seul le graphème est représenté par un caractère informatique. Un exemple courant : tous les allographes de «s» peuvent soit être rendus par un caractère qu'on leur consacre (ſ,...), soit par un seul caractère (s).

Tous les signes nécessaires à cette transcription sont déclarés dans un fichier dédié (ici, chars.py). Ainsi, les seules limites à une approche allographétique seront celles de la modélisation réalisée par le chercheur (quels sont les allographes repérables? Combien de variantes en recense-t-on?) et de l'existence d'une convention pour la représentation de ces signes (idéalement au sein d'Unicode, ou bien d'une initiative collective comme la *Medieval Unicode Font Initiative*, Haugen 2014). Quoiqu'on ait choisi, et si l'on ne veut obtenir plus aisément des résultats, il est préférable de transcrire ce qui figure effectivement sur la source : c'est-à-dire notamment les abréviations (et non leur résolution), la ponctuation et la segmentation originale².

L'étape suivante consiste en la transcription des lignes (ou la correction de lignes prédites par le modèle précédent), permettant de fournir des données d'entraînement en volume suffisant (fig. 3).

Une fois ces données initiales apprêtées par un humain, la machine prend le relais et l'entraînement peut ainsi commencer. Lors de chaque itération de celui-ci, la machine tente d'aligner le texte de la vérité de terrain avec l'image (algorithme CTC) et de prédire ce texte. Le réseau de neurones est ensuite mis à jour en fonction des résultats obtenus. Cet

2. Demander à l'intelligence artificielle de résoudre d'elle-même les abréviations n'est pas impensable, mais bien plus difficile, et demande beaucoup plus de données (voir Bluche et al. 2017). Les scripts de résolution d'abréviation, en revanche, sont beaucoup plus facilement appliqués en post-traitement, ce qui permet, en outre, de conserver dans un même fichier l'état observé et l'état interprété.



FIGURE 3 – Production de données de vérité de terrain pour un imprimé ancien (gauche, *Doctrinal*, 1504) et un ms. (droite, transcr. allographétique du chans. I)

apprentissage se poursuit alors durant plusieurs dizaines ou centaines de milliers d’itérations, jusqu’à obtenir un résultat satisfaisant. La quantité de données disponibles est très importante si on veut retarder ce que l’on nomme l’*overtraining*, c’est-à-dire le moment où le réseau « apprend par cœur » les données d’entraînement, et commence à régresser sur les données de test (cf. fig. 4; après l’itération 120 000, l’erreur – non portée sur le graphique – sur les données de test recommence à augmenter).

À partir de ce moment, se met en place un va-et-vient entre l’humain et la machine. Les différents modèles enregistrés par le programme sont évalués. C’est-à-dire qu’ils sont appliqués, en aveugle, sur des données sélectionnées de façon aléatoire et déjà transcrites mais non vues lors de l’entraînement, de façon à ce que l’on puisse compter les erreurs.

Le meilleur modèle est alors retenu et appliqué à une autre sélection du texte. L’opérateur peut alors, très rapidement, corriger cette sélection et ainsi créer plus de données vérifiées qui permettront une nouvelle série d’entraînements. Lorsque le dernier modèle entraîné est jugé suffisamment performant, on l’applique à l’ensemble de l’ouvrage et on récupère l’ensemble des données. Dans le cas d’ouvrages particulièrement lisibles et pour lesquels on peut prétendre arriver à plus de 99% de réussite, il est préférable de se satisfaire du texte créé par l’OCR. Il y a de grandes chances pour que certaines menues erreurs du transcripteur humain aient été rectifiées par la machine. En revanche, dans le cas d’ouvrages plus complexes ou dégradés, il est préférable de recourir aux données machine pour compléter celles de l’humain.

Pour un imprimé ancien, on peut raisonnablement atteindre des taux d’erreur de l’ordre de 1%³, ce qui permet immédiatement l’application de méthodes de recherche reposant sur la quantité de données textuelles non vérifiées. Dans le cas du manuscrit, une relecture reste impérative. Néanmoins, le temps nécessaire, par rapport à une transcription brute, est très fortement diminué dès lors que l’on brise la barrière des 10% d’erreur, et à plus forte raison encore, de 5%. Si le premier seuil n’est pas hors de portée avec quelques centaines seulement de lignes d’entraînement, le second demande une quantité bien plus importante

3. Durant nos essais sur des imprimés occitans du XVI^e siècle, seul le modèle des *Salmes* de Salette (1587) est resté à un taux d’échec supérieur à 2 (2,091%), dû sûrement à la mise en page complexe de l’ouvrage. Nous sommes parvenus, pour la traduction de Gerson (1556) à 0,797% d’échec.

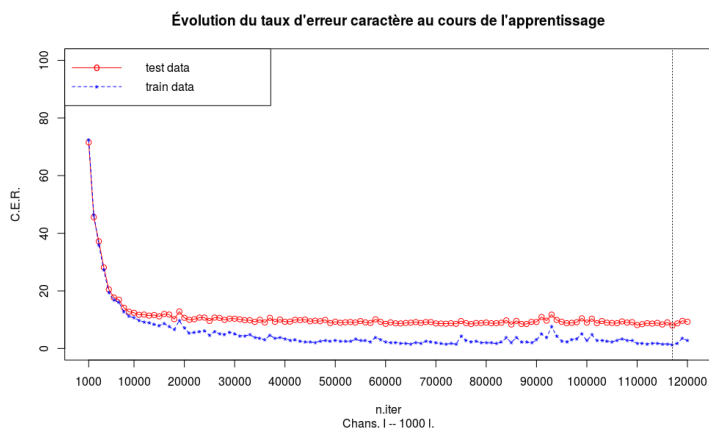


FIGURE 4 – Courbes d’apprentissage du second entraînement sur les données du chansonnier *I*, avec 1 000 lignes transcrites. Le meilleur modèle, à la fois sur les données de test et d’entraînement, apparaît à l’itération 117 000

	Modèle 1	Modèle 2
vérité de terrain (l.)	1000	1000
vér. de terr. artificielle (l.)	0	9000
test (l.)	100	100
n. itér.	117k	295k
CER entr..	1.3	3.5*
CER test	8	6.1
sans espaces	7.4	5.7

TABLE 1 – Résultats des meilleurs modèles pour les entraînements sur le chans. *I*.

*. données artificielles exclues dans le calcul du CER entr.

de données.

Comme la constitution des données de terrain (transcription, relecture) est ce qui est le plus coûteux en temps et compétences, diverses solutions peuvent être employées pour en augmenter artificiellement le nombre, soit en appliquant diverses formes de transformations aux images des lignes, soit en créant des pseudo-pages de manuscrits à partir de l’extraction de la forme des lettres. Ce travail peut être réalisé en quelques minutes grâce à un logiciel tel que DocCreator (Journet et al. 2017 ; LABRI et al. 2018). Si l’on retient la première solution, on pourra appliquer une variété de déformations aux images originelles, y compris des déformations tridimensionnelles (fig. 5).

Dans le cas du chansonnier *I*, 1000 lignes d’entraînement suffisent à obtenir un premier modèle tout à fait exploitable, tandis que l’ajout de lignes artificielles permet une amélioration supplémentaire de presque 2% de CER (table 1).

Que reste-t-il alors comme erreurs ? Les confusions les plus courantes que nous ayons pu observer pour les manuscrits romans ne sont pas surprenantes (table 2 pour le chansonnier *I*). Les erreurs les plus fréquentes concernent ainsi les caractères “combinatoires” –

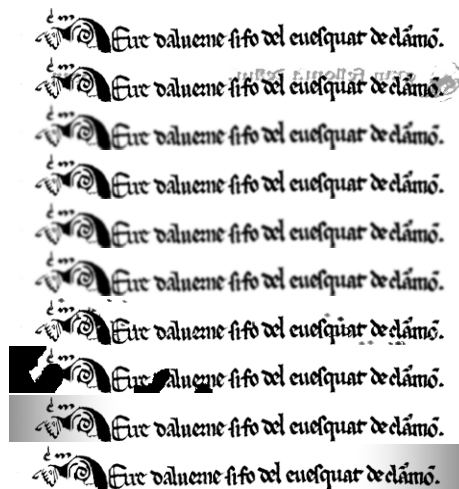


FIGURE 5 – Déformations appliquées, avec DocCreator, aux images de vérité de terrain du chansonnier *I*. De haut en bas : ligne originale, simulation de l'apparition du verso en transparence, quatre formes de flou, dégradation des caractères, simulation de trous et simulation de l'ombre de la reliure à gauche et à droite. Des simulations de déformation 3-D (page froissée...) sont possibles également.

accents et tildes variés –, ainsi que l'espacement. On sait que l'opposition présence/absence d'une espace, propre à l'imprimé, ne correspond pas à la réalité du manuscrit et pose des difficultés au transcritteur qui voudrait reproduire la pratique médiévale. Ces difficultés réapparaissent aussi pour un réseau de neurones. D'autres confusions peuvent s'expliquer par des proximités de forme, entre *e* et *o* par exemple.

2 Structuration de documents textuels

Une fois les données acquises, se pose la question de leur représentation par un modèle, qui est nécessairement une simplification de la réalité, ou pour mieux dire une sélection de faits observables sur le document source, que nous retenons parce que nous voulons les analyser ou parce qu'ils sont d'un intérêt suffisamment général pour justifier le surcroît de travail (Sperberg-McQueen 2009 ; Camps 2016, p. cxv-cc).

Les sources présentent ainsi une grande diversité de faits, de différents ordres :

- matériel (cahiers, feuillets, dégâts touchant au support,...) et relatif à la mise en texte et en page (colonnes, lignes,...) ;
- logique (paragraphes, strophes, vers...);
- graphique (segmentation, allographie, ponctuation,...) ;
- métrique, linguistique, etc.

La première étape de la structuration des données est ainsi la conception d'un modèle abstrait, exprimant les concepts retenus et leurs relations, qui peut ensuite être retranscrit par un formalisme technique donné. Dans ce contexte, les recommandations de la *Text*

freq.	pred.	GT
21	–	[diacr.]
12	[espace]	–
9	–	[espace]
6	[diacr.]	–
5	o	e
4	–	i
4	e	o

TABLE 2 – Matrice des confusions les plus fréquentes obtenues par le meilleur modèle sur les données de test (chansonnier *I*). La colonne de gauche donne ce qui a été prédit par le modèle, et celle de droite la vérité de terrain. Le symbole ‘–’ signale une absence de caractère.

encoding initiative (TEI Consortium 2007–) fournissent un cadre dans lequel peut se placer l’implémentation du modèle.

La conformité à la TEI n’est toutefois pas suffisante en soi : il reste au chercheur à concevoir son modèle comme un sous-ensemble plus spécifique de celle-ci, et documenté selon l’usage qu’il en fait. Pour que ce modèle soit utile, il ne doit pas simplement viser à la satisfaction intellectuelle de son créateur, mais doit demeurer manipulable, permettre l’exploitation scientifique, la publication et la réutilisation des données, ainsi que leur pérennisation et, idéalement, l’échange aisé entre différents projets. D’un point de vue plus général, il est souhaitable de se conformer aux principes de la science ouverte et des « FAIR data », données « trouvables, accessibles, interopérables et réutilisables » (Wilkinson et al. 2016).

Une fois le modèle de données établi, il convient de structurer les documents d’une manière conforme à celui-ci (ce que facilite, en cas d’utilisation de technologies XML, l’utilisation d’un schéma dans un langage tel que RelaxNG). Cette étape de structuration peut être sujette à différentes automatisations. Il est ainsi possible *a minima* de recourir à des scripts dans un langage de programmation, tel que Python ou xslt par exemple, pour une première phase d’encodage sommaire en tirant profit des informations fournies par l’OCR ou déterminables par des motifs et expressions régulières (segmentation en feuillets, lignes, strophes, vers, mots, ...).

Si un encodage automatique de ce type ne permet pas d’aller très loin, des pistes existent pour permettre un encodage fin de documents par la machine, grâce à des méthodes d’intelligence artificielle. Des outils comme Grobid (Romary et al. 2008-2018) permettent déjà de traiter de cette manière des documents très structurés, tels que des publications scientifiques ou des dictionnaires, à partir d’un échantillon encodé de données d’entraînement.

Dans le cas le plus général, pour ce qui est des éditions de manuscrits ou imprimés anciens, une phase d’intervention humaine sur les fichiers s’avère nécessaire, quoique coûteuse et chronophage. Il reste néanmoins possible d’enrichir systématiquement les données transcrites par des méthodes computationnelles, qu’il s’agisse de la reconnaissance des entités nommées ou de la lemmatisation.

3 Enrichir les données transcrites : l'exemple de la lemmatisation

Une infinité d'enrichissements peuvent être accomplis sur les données issues de la transcription, certains plus courants que d'autres : reconnaissance des entités nommées, annotation du système graphique, des morphèmes, du mètre, des figures de style, etc. Parmi ces enrichissements, nous nous attarderons sur un cas très fréquent, et utile à de nombreuses formes d'exploitation des données, à savoir la lemmatisation et l'annotation morpho-syntaxique.

Cet enrichissement des données textuelles par des annotations d'ordre linguistique peut également connaître certaines automatisations. Si ces procédés sont bien connus des linguistes et déjà bien fonctionnels pour les langues modernes, ils posent encore des difficultés pour des langues comme l'occitan médiéval, dues en partie à l'absence de corpus lemmatisés de taille conséquente.

3.1 Principe général

Les lemmatiseurs fondés sur des lexiques de forme sont d'un emploi courant pour les langues contemporaines (par ex. Schmid 1994), mais ceux-ci s'avèrent inadaptés pour des états anciens de langue présentant une forte variation graphique et de nombreuses formes en hapax, et rarement capables de désambiguïser des formes homographes. La très grande variation graphique réduit sensiblement la pertinence des lemmatiseurs fondés sur des lexiques de formes, sachant que des substitutions graphiques nouvelles peuvent apparaître dans chaque nouveau jeu de données soumis au lemmatiseur.

En revanche, la nouvelle génération de lemmatiseurs, qui tire profit des avancées de l'apprentissage machine, est plus prometteuse pour les langues médiévales ou riches en variation. Ceux-ci se fondent souvent sur des corpus annotés – soit des occurrences prises en contexte –, dont ils cherchent à modéliser l'espace sémantique et la variation graphique (Müller et al. 2015 ; Kestemont, Pauw et al. 2016). Ces lemmatiseurs sont ainsi capables de réaliser la tâche centrale à la lemmatisation d'états anciens de langue : l'identification correcte du lemme d'une forme inconnue (i.e. jamais rencontrée durant l'entraînement). Ils peuvent aussi décider correctement du lemme d'une forme qui possède plusieurs homographes.

3.2 Collecte et alignement des données

Pour entraîner ces lemmatiseurs, il faut donc disposer de corpora de données annotées, en quantité et qualité suffisantes. Ce besoin se heurte d'emblée à une première difficulté, celle de l'interopérabilité des jeux de données, en termes de formats, mais aussi en termes de référentiels, particulièrement les listes de lemmes de référence et les jeux d'étiquette. Des barrières d'usage ou juridiques surgissent également rapidement, car l'accès aux données annotées ou la possibilité de les réutiliser n'est pas systématique. Tandis que certaines données peuvent être emprisonnées dans des systèmes propriétaires, d'autres ne sont simplement pas accessibles pour des raisons techniques ou parce que la démarche de les doter d'un statut et d'une licence claire et explicite n'a pas été accomplie. Sur un plan pratique, se pose en outre la question de la quantité de données nécessaires. Il est en effet délicat

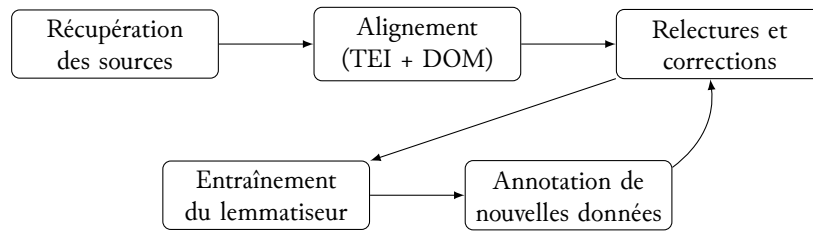


FIGURE 6 – Étapes de l'entraînement d'un lemmatiseur

de parvenir à des résultats dotés d'un peu de généralité sans un corpus dont le nombre d'occurrences n'est pas un multiple de 10^5 , ou, de préférence, 10^6 – à titre de comparaison, la totalité de la COM1 (Ricketts, Akehurst et al. 2001) contient environ $7,3 \times 10^5$ occurrences. Dans les lignes qui suivent, nous décrirons comment nous avons constitué un corpus d'entraînement primaire (annoté), ainsi qu'un corpus secondaire (non annoté) pouvant être utilisé plus marginalement pour ajuster les représentations sémantiques des mots.

Dans le cadre de notre démarche d'entraînement d'un lemmatiseur pour l'occitan médiéval et prémoderne, la première étape a été de tenter de collecter des données d'entraînement, héritées de divers projets. Nous avons ainsi utilisé les données fournies, sous licence libre, par l'édition des *Comptes des consuls de Montferrand* (Lodge 2006), ainsi que deux corpora qui nous ont été fournis par leurs auteurs (Field 2016 ; Scrivner et al. 2016)⁴. Il nous a ensuite fallu procéder selon les étapes présentées en fig. 6.

Comme ces données étaient dans des formats différents, l'étape suivante est de les aligner sur un format commun, avec un même jeu de lemmes de référence.

Les données des *Comptes...* se présentaient déjà sous un format TEI, en partie seulement segmenté et lemmatisé, via des liens avec les entrées du glossaire (fig. 7). Nous avons donc procédé, par un script `xslt`, à une segmentation complète, et avons aligné les entrées du glossaire avec le jeu de lemmes de référence que nous avons retenu, à savoir celui du *Dictionnaire de l'occitan médiéval* (Stempel et al. 1996–), comme présenté en fig. 8. Pour ce faire, un script a interrogé systématiquement les entrées du DOM à partir de l'entrée du glossaire, et proposé une ou plusieurs correspondances possibles, qui ont été vérifiées par l'humain et intégrées aux données (ligne 3 sur la figure). Dans les cas où aucun lemme du DOM ne correspondait à l'occurrence, nous avons créé et documenté de nouveaux lemmes, en tentant de suivre les mêmes principes de création que le DOM.

Une fois l'alignement fait, nous avons réinjecté l'information de lemme dans les données segmentées (fig. 9).

Nous avons procédé d'une manière similaire pour le *Flamenca* lemmatisé fourni par Scrivner et al. 2016. À l'issue de ce travail, une ultime phase de reprise d'ensemble, pour une correction et harmonisation générale, des données s'est avérée nécessaire. Pour ce faire, l'utilisation d'un tableur peut suffire, mais des logiciels plus perfectionnés existent, qui permettent la vérification de l'intégrité des lemmes et étiquettes utilisées, tout en facilitant

4. À la date d'écriture de cet article, nous n'avons pas encore eu recours au corpus fourni par Field 2016, n'ayant encore pu compléter de manière systématique les informations qui y figuraient et les aligner sur le référentiel commun utilisé.

```

1 <item xml:id="CC6.278">Item
2   IIII s. qu <w lemmaRef="#gloss_a116_11">ac</w>
3   D. Chapus <w lemmaRef="#gloss_d57">dos</w> II jorns
4   <w lemmaRef="#gloss_d40">desus</w>
5   <w lemmaRef="#gloss_d49_9">dit</w>
6   <w lemmaRef="#gloss_q11">que</w>
7   <w lemmaRef="#gloss_a47_4">anet</w>
8   <w lemmaRef="#gloss_e9">en</w> Garsias
9   <w lemmaRef="#gloss_p30">per</w>
10  <w lemmaRef="#gloss_m73">mostrar</w>
11  las <w lemmaRef="#gloss_g16">gens</w> e los
12 ostals.</item>

```

FIGURE 7 – État original des fichiers sources de Lodge 2006

```

1 <entry n="116" xml:id="gloss_a116">
2   <form>aver</form>
3   <form source="#DOM" type="lmlv" cert="high">aver</form>
4   <gramGrp>
5     <pos>verbe</pos>
6     <subc>transitif</subc>
7     <mood>infinitif</mood>
8   </gramGrp>
9   <re xml:id="gloss_a116_1">
10    <form>avem</form>
11    <gramGrp>
12      <pos>verbe</pos>
13      <mood>indicatif</mood>
14      <tns>présent</tns>
15      <per>4</per>
16    </gramGrp>
17  </re>
18  <!-- Suite de l'entrée -->
19 </entry>

```

FIGURE 8 – Aligement des entrées du glossaire de Lodge 2006 avec les lemmes du DOM (Stempel et al. 1996–)

```

1 <item xml:id="CC6.278">
2 <w lemma="item" xml:id="w_028267">Item</w>
3 <w lemma="@num@" xml:id="w_028268">IIII</w>
4 <w lemma="sol" xml:id="w_028269">s</w>
5 <pc>.</pc>
6 <w lemma="que" xml:id="w_028270">qu</w>
7 <w lemma="aver" xml:id="w_028271" lemmaRef="#gloss_a116_11">ac</w>
8 <w lemma="D" xml:id="w_028272">D</w>
9 <pc>.</pc>
10 <w lemma="Chapus" xml:id="w_028273">Chapus</w>
11 <w lemma="da+lo2" xml:id="w_028274" lemmaRef="#gloss_d57">dos</w>
12 <w lemma="@num@" xml:id="w_028275">II</w>
13 <w lemma="jorn" xml:id="w_028276">jorns</w>
14 <w lemma="desus" xml:id="w_028277" lemmaRef="#gloss_d40">desus</w>
15 <w lemma="dire" xml:id="w_028278" lemmaRef="#gloss_d49_9">dit</w>
16 <w lemma="que" xml:id="w_028279" lemmaRef="#gloss_q11">que</w>
17 <w lemma="anar" xml:id="w_028280" lemmaRef="#gloss_a47_4">anet</w>
18 <w lemma="en" xml:id="w_028281" lemmaRef="#gloss_e9">en</w>
19 <w lemma="Garsias" xml:id="w_028282">Garsias</w>
20 <w lemma="per" xml:id="w_028283" lemmaRef="#gloss_p30">per</w>
21 <w lemma="mostrar" xml:id="w_028284" lemmaRef="#gloss_m73">mostrar</w>
22 <w lemma="lo2" xml:id="w_028285">las</w>
23 <w lemma="gen" xml:id="w_028286" lemmaRef="#gloss_g16">gens</w>
24 <w lemma="e" xml:id="w_028287">e</w>
25 <w lemma="lo2" xml:id="w_028288">los</w>
26 <w lemma="ostal" xml:id="w_028289">ostals</w>
27 <pc>.</pc>
28 </item>

```

FIGURE 9 – État final des fichiers des *Comptes...*

la correction par lots des cas similaires (par ex. Clérice 2017).

Nous disposons ainsi d'un premier (petit) corpus de 54600 occurrences, mais les lemmatiseurs de nouvelle génération, comme Pandora ou Lemming, fournissent aussi parfois la possibilité d'utiliser un corpus secondaire, non annoté, mais fournissant une grande quantité d'occurrences. Ce corpus secondaire peut être utilisé pour améliorer la modélisation de l'espace sémantique de la langue, en proposant une représentation des formes en fonction de leur contexte (à gauche et à droite) s'appuyant sur une quantité plus importante (et donc plus fiable) de données. Pour cela, Pandora a recours à la technologie de « *word embeddings* » (ou, plongement de mots, en français) fournie par l'algorithme « skipgram » de *w2vec* (Mikolov et al. 2013). Lemming utilise l'algorithme « Marlin » (Martin et al. 1998).

Pour ce corpus secondaire, nous avons extrait et utilisé les textes du premier volume des COM (Ricketts, Akehurst et al. 2001), ce qui a nécessité de les convertir en TEI et de les segmenter en mots.

3.3 Entraînement d'un modèle de lemmatisation

Nous avons retenu l'utilisation du lemmatiseur Pandora (Kestemont 2016), sans pouvoir utiliser en parallèle Lemming (Müller et al. 2015), car, au cours des opérations d'alignement du corpus, nous n'avons pour l'instant traité que les lemmes et pas les étiquettes morpho-syntaxiques (pour lesquelles un jeu consensuel d'étiquettes n'existe pas à notre connaissance pour l'occitan ancien). Lemming prenant en compte de pair morpho-syntaxe et lemmes, son utilisation devenait, pour le moment, impossible, tandis que Pandora les traite séparément. À l'avenir, il sera utile de compléter le corpus sur ce point pour permettre la comparaison des résultats de ces deux outils.

Pandora est un lemmatiseur en cours de développement, en Python, utilisant des bibliothèques d'intelligence artificielle telles que Keras et TensorFlow. Son code est libre et ouvert. Il a été conçu spécifiquement pour traiter les particularités des états médiévaux de langue, et notamment leur forte variabilité graphique. Concrètement, Pandora utilise des plongements de mots et de caractères pour créer une représentation, par des vecteurs, des mots, de leurs caractères et de leur contexte à gauche et à droite. Cette représentation est ensuite fournie à un réseau de neurone convolutif ou récurrent (de type LSTM) qui apprend sur les données du corpus et réalise la tâche de classification (Kestemont, Pauw et al. 2016 ; Kestemont et De Gussem 2017).

Nous avons ainsi pu entraîner des modèles de lemmatisation avec Pandora. Dans un premier temps, les données des COM ont servi pour réaliser la représentation sémantique initiale des mots, en 100 dimensions avec l'algorithme SKIPGRAM. Cette représentation peut être visualisée par réduction en deux dimensions (fig. 10), et donne à voir des proximités sémantiques. Celles-ci rapprochent des graphies différentes du même lemme, mais aussi des mots de sens voisin (i.e. employés dans des contextes similaires). Ces proximités peuvent aussi être interrogées directement et affichées avec un coefficient de similarité. Ainsi, les mots les plus proches de *domna* sont *dompna* (0,97), *dona* (0,87), *bona* (0,86), *amia* (0,85), etc. Ceux les plus proches de *joy* sont *gaug* (0,92), *deport* (0,88), *ioy* (0,85), *esperansa* (0,85), *pessamen* (0,84), etc.

L'entraînement peut ensuite commencer sur les données annotées. De celles-ci, 80% sont utilisées pour l'entraînement, 10% pour des ajustements, et 10% pour tester la per-

	Entr.	Dev.	Test
toutes formes	0,99	0,86	0,88
formes connues	0,99	0,92	0,93
formes inconnues	NA	0,35	0,37

TABLE 3 – Précision des résultats obtenus par le modèle entraîné avec Pandora après 100 époques d’apprentissage, sur les données d’entraînement, d’ajustement et de test

formance des modèles. Durant chaque époque de l’entraînement, l’ensemble du corpus d’entraînement est soumis au réseau de neurones convolutif, qui apprend et tente de prédire les lemmes. Après un nombre donné d’époques (ici, 100), l’entraînement est arrêté. Les résultats obtenus par le meilleur modèle sont présentés en table 3. Si la performance sur les formes connues est bonne, elle demeure très nettement en deçà de ce qu’il est possible d’atteindre sur les formes inconnues avec une quantité supérieure de données d’entraînement.

4 Des perspectives d’exploitation renouvelées

Si les traitements présentés ici sont d’ores-et-déjà fonctionnels, et permettent d’obtenir des résultats intéressants en production de corpus, la clef de leur amélioration pour l’avenir consiste en le nécessaire accroissement de la quantité de données disponibles et utilisables. Une quantité supérieure de données d’entraînement signifie en effet l’obtention de modèles plus performants qui permettent en retour un accroissement plus rapide de la quantité de données exploitables.

Pour cela, il faut que les données soient interoperables et documentées, mais aussi rendues disponibles en ligne, sous une licence libre, permettant la réutilisation, et sous une forme citable et pérenne. Ce n’est qu’à ce prix que l’on permettra un progrès des outils, méthodes et connaissances sur les textes occitans médiévaux et pré-modernes.

Remerciements

Les auteurs remercient Marine Mazars et Lucence Ing pour leur collaboration philologique et technique, ainsi que Thomas Fields et Olga Scrivner pour la communication de leurs données. Cet article s’inscrit dans une démarche relevant des projets LAKME (financement de l’Université Paris Sciences & Lettres) et CoRLiG (ministère de la culture, DGLFLF).

Références

ARTSIMOVICH, Joseph, Nate CRAUN et al. (2016). *ScanTailor*. Version 0.9.12. URL : <http://scantailor.org/>.

- BLUCHE, Théodore, Sebastien HAMEL, Christopher KERMORVANT, Joan PUIGSERVER, Dominique STUTZMANN, Alejandro H TOSELLI et Enrique VIDAL (2017). « Preparatory KWS Experiments for Large-Scale Indexing of a Vast Medieval Manuscript Collection in the HIMANIS Project ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 311–316.
- BREUEL, Thomas M (2014). *Ocropy : Python-based tools for document analysis and OCR*. URL : <https://github.com/tmbdev/ocropy> (visité le 29/10/2016).
- BREUEL, Thomas M, Adnan UL-HASAN, Mayce Ali AL-AZAWI et Faisal SHAFAIT (2013). « High-performance OCR for printed English and Fraktur using LSTM networks ». In : *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, p. 683–687.
- CAMPS, Jean-Baptiste (2016). « La Chanson d’Otinel : édition complète du corpus manuscrit et prolégomènes à l’édition critique ». Thèse de doct. Université Paris-Sorbonne-Paris IV.
- CARRASCO, Raphaël, Vincent CHALLET, Gilda CAÏTI-RUSSO, Stéphane DURAND, Marc CONESA et Yves MAUSEN, éd. (2014–). *Le « Petit Thalamus » de Montpellier : édition critique numérique du manuscrit AA9 des Archives municipales : page d’accueil*. URL : <http://thalamus.huma-num.fr/> (visité le 28/05/2018).
- CLÉRICE, Thibault (2017). *pandora-postcorrect-app*. Avec la coll. de Jean-Baptiste CAMPS. original-date : 2017-09-25T08:26:41Z. URL : <https://github.com/hipster-philology/pandora-postcorrect-app> (visité le 27/10/2017).
- FIELD, Thomas T., éd. (2016). *Corpus linguistique de l’ancien occitan gascon*. Baltimore. URL : <http://mllidev.umbc.edu/gascon/French/description/index.html> (visité le 28/05/2018).
- UL-HASAN, Adnan (2016). « Generic Text Recognition using Long Short-Term Memory Networks ». URL : <https://kluedo.uni-kl.de/frontdoor/index/index/docId/4353> (visité le 21/06/2016).
- HAUGEN, Odd Einar (2014). *Medieval Unicode Font Initiative*. URL : <http://folk.uib.no/hnooh/mufi/> (visité le 08/05/2015).
- JOURNET, Nicholas, Muriel VISANI, Boris MANSENCAL, Kieu VAN-CUONG et Antoine BILLY (2017). « DocCreator : A New Software for Creating Synthetic Ground-Truthed Document Images ». In : *Journal of imaging* 3.4, p. 62.
- KESTEMONT, Mike (2016). *Pandora : A Tagger-Lemmatizer for Latin*. Version en développement. Anvers. URL : <https://github.com/mikekestemont/pandora> (visité le 27/10/2016).
- KESTEMONT, Mike et Jeroen DE GUSSEM (2017). « Integrated Sequence Tagging for Medieval Latin Using Deep Representation Learning ». In : *Journal of Data Mining & Digital Humanities*. URL : <https://arxiv.org/abs/1603.01597>.
- KESTEMONT, Mike, Guy de PAUW, Renske van NIE et Walter DAELEMANS (2016). « Lemmatization for variation-rich languages using deep learning ». In : *Digital Scholarship in the Humanities* fqw034. ISSN : 2055-7671, 2055-768X. DOI : 10.1093/llc/fqw034. URL : <http://dsh.oxfordjournals.org/content/early/2016/08/26/llc.fqw034> (visité le 23/10/2016).
- LABRI et L3I (2018). *DocCreator : DIAR software for synthetic document image and ground-truth generation, with various degradation models for data augmentation*. original-date :

- 2017-10-20T10 :53 :54Z. Bordeaux, La Rochelle. URL : <https://github.com/DocCreator/DocCreator> (visité le 26/06/2018).
- LODGE, R. Anthony, éd. (2006). *Les comptes des consuls de Montferrand (1273–1319)*. Avec la coll. d'Olivier CANTEAUT, Frédéric GLORIEUX, Nicolas MAZZIOTTA, Anna GADRÉ et Jocelyn CHALICARNE. Éditions en ligne de l'École des chartes (Élec) 16. Paris. URL : <http://elec.enc.sorbonne.fr/montferrand/> (visité le 28/05/2018).
- MARTIN, Sven, Jörg LIERMANN et Hermann NEY (1998). « Algorithms for bigram and trigram word clustering1 ». In : *Speech communication* 24.1, p. 19–37.
- MIKOLOV, Tomas, Ilya SUTSKEVER, Kai CHEN, Greg S CORRADO et Jeff DEAN (2013). « Distributed representations of words and phrases and their compositionality ». In : *Advances in neural information processing systems*, p. 3111–3119.
- MÜLLER, Thomas, Ryan COTTERELL, Alexander FRASER et Hinrich SCHÜTZE (2015). « Joint lemmatization and morphological tagging with lemming ». In : *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2268–2274.
- NÉPOTE, Fanny, Nathalie DAUVOIS, Jean-François COUROUOU et Jean-Luc NARDONE, édés. (2006). *Bibliotheca tholosana - Accueil*. Toulouse. URL : <http://bibliotheca-tholosana.fr/> (visité le 28/05/2018).
- RICKETTS, Peter Thomas, F. R. P. AKEHURST, John HATHAWAY et Cornelis Henricus Maria van der HORST (2001). *Concordance de l'Occitan médiéval (COM) = The concordance of medieval Occitan*. Turnhout, Belgique : Brepols Publishers. 16+16 ; 62. ISBN : 978-2-503-51159-7.
- RICKETTS, Peter Thomas, Alan REED, F. R. P. AKEHURST, John HATHAWAY et Cornelis Henricus Maria van der HORST (2005). *Concordance de l'Occitan médiéval : COM 2 : les troubadours, les textes narratifs en vers = The concordance of medieval Occitan : the troubadours, narrative verse*. Turnhout, Belgique : Brepols Publishers. 135 p. ISBN : 978-2-503-51416-1.
- ROMARY, Laurent, Patrice LOPEZ et al. (2008-2018). *Grobid-information extraction from scientific publications*. URL : <https://github.com/kermitt2/grobid>.
- SCHMID, Helmut (1994). *TreeTagger : a part-of-speech tagger for many languages*. Stuttgart. URL : <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/> (visité le 29/10/2016).
- SCRIVNER, Olga, Michaël Paul MCGUIRE, Sandra KÜBLER, Barbara VANCE et E.D. BLODGETT, édés. (2016). *Old Occitan Corpus – Digital Collection*. Bloomington. URL : www.oldoccitancorpus.org (visité le 28/05/2018).
- SÉGUY, Jean (1971). « La relation entre la distance spatiale et la distance lexicale ». In : *Revue de linguistique romane* 35, p. 335–357.
- SPERBERG-McQUEEN, C Michael (2009). « How to teach your edition how to swim ». In : *Literary and linguistic computing* 24.1, p. 27–39.
- SPRINGMANN, Uwe et David KAUMANN (2015). *Ocrocis : a high accuracy OCR method to convert early printings into digital text*. URL : <http://cistern.cis.lmu.de/ocrocis/tutorial.pdf>.
- STEMPEL, Wolf-Dieter, Maria SELIG, Claudia KRAUS, Renate PETER et Monika TAUSEND (1996–). *Dictionnaire de l'occitan médiéval (DOM en ligne)*. Munich. URL : <http://www.dom-en-ligne.de/>.
- STUTZMANN, Dominique (2011). « Paléographie statistique pour décrire, identifier, dater. . . Normaliser pour coopérer et aller plus loin? » In : *Kodikologie und Paläographie*

im digitalen Zeitalter 2 / Codicology and Palaeography in the Digital Age 2. Sous la dir. de Fischer FRANZ, Christiane FRITZ et Georg VOGELER. Schriften des Instituts für Dokumentologie und Editorik, 3. Norderstedt, p. 247–277. URL : <https://halshs.archives-ouvertes.fr/halshs-00596970/>.

TEI CONSORTIUM (2007–). *TEI P5 : Guidelines for Electronic Text Encoding and Interchange*. URL : <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/> (visité le 09/05/2015).

WILKINSON, Mark D, Michel DUMONTIER, IJsbrand Jan AALBERSBERG, Gabrielle APPLETON, Myles AXTON, Arie BAAK, Niklas BLOMBERG, Jan-Willem BOITEN, Luiz Bonino da SILVA SANTOS, Philip E BOURNE et al. (2016). « The FAIR Guiding Principles for scientific data management and stewardship ». In : *Scientific data* 3.