RECITAL 2008, Avignon, 9–13 juin 2008

Transducteurs à fenêtre glissante pour l'induction lexicale

Yves Scherrer
LATL, Université de Genève
Rue de Candolle 5
1211 Genève 4, Suisse
yves.scherrer@lettres.unige.ch

Résumé. Nous appliquons différents modèles de similarité graphique à la tâche de l'induction de lexiques bilingues entre un dialecte de Suisse allemande et l'allemand standard. Nous comparons des transducteurs stochastiques utilisant des fenêtres glissantes de 1 à 3 caractères, entraînés à l'aide de l'algorithme de maximisation de l'espérance avec des corpus d'entraînement de tailles différentes. Si les transducteurs à unigrammes donnent des résultats satisfaisants avec des corpus très petits, nous montrons que les transducteurs à bigrammes les dépassent à partir de 750 paires de mots d'entraînement. En général, les modèles entraînés nous ont permis d'améliorer la F-mesure de 7% à 15% par rapport à la distance de Levenshtein.

Abstract. We apply different models of graphemic similarity to the task of bilingual lexicon induction between a Swiss German dialect and Standard German. We compare stochastic transducers using sliding windows from 1 to 3 letters, trained with the Expectation-Maximisation algorithm on training corpora of different sizes. While the unigram transducers provide good results with very small corpora, we show that bigram transducers outperform them with corpora of 750 word pairs or more. Overall, the trained models show between 7% and 15% F-measure improvement over Levenshtein distance.

Mots-clés : Induction lexicale, transducteurs stochastiques, langues apparentées.

Keywords: Lexicon induction, stochastic transducers, cognate languages.

1 Introduction

Les ressources lexicales constituent une partie essentielle de tout système de traitement automatique des langues. Comme la construction manuelle de telles ressources est fastidieuse et gourmande en temps, l'induction automatique de ressources lexicales est une alternative particulièrement attractive. Dans cet article, nous discuterons différentes approches pour l'induction d'un dictionnaire bilingue entre un dialecte suisse allemand et la variété standard de l'allemand.

Le choix particulier de cette paire de langues a des conséquences importantes sur la méthodologie. En Suisse allemande, les dialectes et la variété standard forment une diglossie médiale : les dialectes sont utilisés à l'oral, tandis que l'allemand standard est surtout utilisé à l'écrit. A cause de cette distribution complémentaire, il est difficile de trouver des corpus parallèles, et généralement des textes écrits en dialecte. Pourtant, l'attitude positive de la population envers le dialecte et son utilisation généralisée en font un candidat attractif pour le traitement automa-

tique. Ces contraintes relatives à la disponibilité des données placent notre recherche dans le contexte du traitement de langues peu dotées.

En même temps, les dialectes alémaniques sont étroitement apparentés à l'allemand standard. Cette parenté réduit la complexité des relations lexicales à induire. Nos travaux s'insèrent donc dans le courant de recherche du traitement de langues apparentées. Nous soutenons que l'avantage de la parenté étroite est à même de lever quelques restrictions imposées par la rareté des ressources. Plus précisément, nous faisons l'hypothèse que dans le cas de deux langues apparentées, l'utilisation de techniques d'apprentissage automatique est possible même si peu de ressources existent pour l'une d'entre elles.

Nous concevons un dictionnaire bilingue essentiellement comme une liste de paires de mots.[1] L'induction de paires de mots se fonde sur un critère de similarité. Théoriquement, cette similarité est d'ordre sémantique : deux mots sont associés dans un dictionnaire s'ils renvoient au même concept. Or, il est difficile d'extraire directement les significations des mots à partir de données textuelles brutes. En général, on recourt à des critères de similarité plus simples et plus opérationnels, mais néanmoins corrélés avec la similarité sémantique. L'approche classique se base sur l'alignement de mots dans un corpus parallèle (Brown *et al.*, 1993) : deux mots sont considérés similaires s'ils ont une probabilité d'alignement suffisante. Une autre approche (Rapp, 1999) s'affranchit de corpus parallèles ; selon celle-ci, deux mots sont similaires s'ils apparaissent dans des contextes lexicaux similaires.

Les propriétés particulières de notre paire de langues nous ont amené à considérer un autre type de similarité : la similarité graphique. Pour des langues étroitement apparentées, une grande partie du lexique général est constitué de paires lexicales apparentées (*cognate word pairs*) : les mots qui ont la même signification ont également une forme phonétique et graphique similaire[2].

La similarité graphique a l'avantage de fournir de bons résultats avec peu de ressources. S'il est possible de l'utiliser sans données d'entraînement, nous montrerons que l'apprentissage automatique avec un corpus de taille modeste améliore nettement les résultats. Nous nous appuyons sur des recherches récentes dans le domaine de la traduction entre phonèmes et graphèmes afin de tenir compte du contexte des lettres lors du calcul de la similarité. Nous nous focalisons sur les correspondances de mots simples à mots simples.

Après une discussion des travaux récents dans ce domaine, nous présenterons notre architecture d'induction lexicale ainsi que les différents modèles implémentés (section 3). Ensuite, nous décrirons les données utilisées pour l'entraînement et l'évaluation (section 4), suivi par la présentation et la discussion des résultats obtenus (section 5).

2 Travaux connexes

Les mesures de similarité phonétique et graphique sont utilisées extensivement dans le cadre du traitement de la parole, afin de transformer des séquences de phonèmes en séquences de

[1] Étant donné la parenté de nos langues, des informations morphologiques et syntaxiques, disponibles pour le côté allemand standard, pourront être projetées sur le côté dialecte sans modifications majeures.

[2] Evidemment, l'existence de conventions orthographiques pour les langues en question et la nature de ces dernières peuvent influencer les résultats de cette méthode. Bien qu'il n'existe pas de conventions orthographiques obligatoires pour les dialectes suisse allemands, des questions pratiques nous ont amené à utiliser des données écrites (c'est-à-dire non transcites phonétiquement). Cependant, l'orthographe utilisée est très proche de la prononciation.

lettres et inversement. Dans ce cadre, (Ristad & Yianilos, 1998) introduisent des méthodes d'apprentissage automatique : ils entraînent un transducteur stochastique sans mémoire (à un état) en utilisant l'algorithme de maximisation de l'espérance (EM). Cet algorithme itératif permet d'estimer les probabilités des transitions du transducteur stochastique à partir d'un corpus d'entraînement contenant des paires de mots corrects.

Le modèle de Ristad & Yianilos a été repris pour l'induction de dictionnaires bilingues entre langues apparentées (Mann & Yarowsky, 2001). L'idée de Mann & Yarowsky est d'étendre un lexique bilingue existant à une langue apparentée. Par exemple, un lexique anglais-espagnol peut servir de base pour un lexique anglais-portugais, l'espagnol jouant le rôle de pivot. Les mesures de similarité graphique (appelés *cognate models*) sont utilisées pour apparier les mots espagnols et les mots portugais. Les auteurs de cette étude font la distinction entre des *mesures statiques*, qui sont assez génériques pour être appliquées à toute paire de langues sans entraînement préalable, et des *mesures adaptives*, qui sont adaptées à une paire de langues précise. En particulier, un transducteur stochastique entraîné à l'aide de EM comme mesure adaptive ainsi que la distance de Levenshtein comme mesure statique. La distance de Levenshtein entre deux chaînes de caractères est définie comme le nombre minimal d'opérations d'édition (insertion, effacement ou substitution d'un caractère) nécessaires pour transformer une chaîne dans l'autre.

La distance de Levenshtein et les méthodes basées sur les transducteurs sans mémoire ne prennent pas en compte le contexte : un seul symbole de l'entrée est comparé avec un seul symbole de la sortie à la fois. Cette approche s'est avérée insuffisante dans le cadre de la conversion entre phonèmes et lettres : *ph* doit être converti en *[f]*, tandis que *x* doit être converti en *[ks]*. Une solution au premier problème est l'utilisation d'une fenêtre glissante (Jansche, 2001) : on regarde plusieurs caractères dans l'entrée pour en générer un seul à la sortie. Une autre technique, plus sophistiquée, consiste à adapter l'algorithme d'apprentissage pour entraîner des correspondances plusieurs-à-plusieurs (Jiampojamarn *et al.*, 2007). Cette technique nécessite un prétraitement du mot source ; il faut le couper en morceaux d'une à plusieurs lettres afin de déterminer les types de correspondances à utiliser. Nous avons montré dans (Scherrer, 2007) qu'un transducteur basé sur des règles dépendantes du contexte, implémentées manuellement, obtient de meilleures performances que le transducteur sans mémoire entraîné avec EM.

Les méthodes d'induction lexicale proposées ici s'appliquent aux configurations linguistiques dans lesquelles la majorité des paires lexicales se ressemblent graphiquement. Si cette condition est vérifiée pour le lexique général de langues apparentées, elle l'est aussi pour des domaines lexicaux spécifiques, indépendamment du degré de parenté des langues. Dans cette optique, (Claveau & Zweigenbaum, 2005) entraînent un transducteur (non stochastique) pour inférer des traductions françaises de termes biomédicaux anglais. (Claveau, 2007) étend cette technique à d'autres paires de langues (par exemple, anglais-russe) et introduit un modèle de la langue cible pour filtrer les candidats lexicaux proposés.

3 Modèles d'induction lexicale

3.1 Les deux étapes de l'induction lexicale

En traduction automatique statistique, il est usuel de partager la problématique en deux tâches distinctes. La traduction d'une phrase doit en effet satisfaire deux conditions principales. Premièrement, son contenu doit rester fidèle à la phrase source, et deuxièmement, sa forme doit être

conforme à la grammaire de la langue cible. La première condition est garantie par le *modèle de traduction*, la deuxième par le *modèle de langue*.

Nous reprenons cette architecture pour l'induction du lexique. Dans une première étape, nous proposons des chaînes de caractères qui restent similaires au mot source. C'est ici que les différentes mesures de similarité graphique interviennent : elles génèrent une suite de chaînes de caractères, ordonnées par leur taux de similarité graphique par rapport au mot source. Dans une seconde étape, nous devons garantir que les chaînes de caractères ainsi générées soient conformes à la langue cible. Pour cela, nous utilisons une liste de mots de l'allemand standard comme filtre : seules les chaînes de caractères qui sont des mots allemands sont retenues. On peut donc considérer ce filtre lexical comme un modèle de langue binaire. La figure 1 illustre cette architecture à l'aide d'un exemple.

3.2 Distance de Levenshtein

La distance de Levenshtein entre deux chaînes de caractères est définie comme le nombre minimal d'opérations d'édition nécessaires pour transformer une chaîne dans l'autre.[3] Il y a trois types d'opérations d'édition : l'insertion d'un caractère, la substitution d'un caractère par un autre, et l'effacement d'un caractère. La distance de Levenshtein opère sur des caractères isolés sans prendre en compte les caractères précédents et suivants. Ainsi, elle peut être implémentée dans un transducteur sans mémoire (à un état). Par ailleurs, cette mesure de distance est statique ; elle est identique pour toutes les paires de langues. Nous l'utiliserons comme modèle de référence pour nos expériences.

3.3 Transducteurs stochastiques entraînés avec EM

Un transducteur implémentant la distance de Levenshtein possède deux classes de transitions : les transitions d'édition avec un coût unitaire, et les transitions d'identité (le même caractère en entrée et en sortie) avec un coût de 0. Pour des applications linguistiques, cette classification binaire est souvent insuffisante. Par exemple, lorsqu'on traduit des mots suisse allemands en allemand standard, l'insertion de *n* ou de *e* est beaucoup plus fréquente que celle de *m* ou de *i*. De même, un *a* reste plus souvent identique qu'un *ü*. Afin de pouvoir prédire de tels phénomènes spécifiques, il nous faut *primo* un type de transducteur plus souple, permettant d'associer des poids différents à chaque transition, et *secundo* un mécanisme d'apprentissage automatique pour déterminer ces poids. Suivant (Ristad & Yianilos, 1998), nous utilisons un transducteur stochastique pour satisfaire la première exigence, et l'algorithme EM pour satisfaire la seconde.[4]

Dans un transducteur stochastique, toutes les transitions représentent des probabilités. La probabilité de transduction d'une paire de mots donnée est la somme des probabilités de tous les chemins qui la génèrent. L'algorithme EM sert à trouver les probabilités de transition de sorte à

[3]Dans cet article, nous utilisons parfois le terme *similarité*, parfois le terme *distance*. Comme les valeurs de similarité ou de distance servent seulement à ordonner les candidats générés, le rapport exact entre ces deux notions ne nous semble pas important : dans ce travail, il nous suffit de pouvoir comparer des listes ordonnées par similarité décroissante avec des listes ordonnées par distance croissante.

[4]Pour notre paire de langues, le nombre de transpositions de caractères est relativement restreint ; on peut donc envisager de créer un transducteur à la main, sans utiliser un algorithme d'apprentissage. (Scherrer, 2007) présente une telle approche.

Mot d'entrée	**Première étape Génération de candidats**		**Deuxième étape Filtrage des candidats**	
vermuetet	vermuetet	29.87	**vermutet**	32.19
	vermutet	32.19	vermutete	36.08
	vermuett	32.19	vermute	36.65
	vrmuetet	32.19	vermuten	37.69
	vermaetet	32.68	vermutetet	39.23
	vermuetit	33.41	vermottet	39.41
	vermuitet	33.41	vermuteten	39.72
	virmuetet	33.41	vermutest	40.57
	vermuetent	33.51		
	vermunetet	33.51		
	vnermuetet	33.51		
	vermuetetn	33.51		
	nvermuetet	33.51		
	...	(10000 candidats)		

FIG. 1 – Sortie du modèle d'induction lexicale pour le mot dialectal *vermuetet 'supposé'*. Ce mot doit être associé au mot allemand standard *vermutet* (en gras). La colonne du milieu montre les chaînes de caractères générées à l'aide du modèle de similarité graphique. Les chiffres correspondent à des logarithmes négatifs de probabilités, et proviennent du transducteur stochastique à unigrammes (cf. section 3.3). La colonne de droite montre les candidats ayant passé la deuxième étape, c'est-à-dire ceux qui sont effectivement des mots allemands.

ce qu'elles maximisent la vraisemblance de générer les paires de mots vues pendant l'entraînement. Cet objectif peut être atteint itérativement en utilisant une liste de paires de mots corrects. Le transducteur est initialisé avec des probabilités uniformes. En traduisant les paires de mots de la liste d'entraînement, il compte toutes les transitions utilisées dans ce processus. Ensuite, les probabilités des transitions sont réestimées selon la fréquence d'utilisation des transitions comptées auparavant. Ces nouvelles probabilités sont ensuite utilisées dans l'itération suivante.

3.4 Transducteurs à fenêtre glissante

Le transducteur stochastique présenté ci-dessus ne tient pas compte du contexte graphique des caractères. La figure 1 illustre bien cette propriété : le modèle a appris que l'élimination de *e* est peu coûteuse, mais cette élimination obtient la même probabilité dans toutes les positions. On génère ainsi beaucoup de candidats inutiles, car éliminés dans la seconde étape. (Jansche, 2001) a présenté une solution à ce problème. Au lieu de fournir au transducteur un caractère à la fois, il lui fournit également le caractère précédent et le caractère suivant. A partir d'un trigramme d'entrée, le transducteur doit prédire un seul caractère, celui du milieu. Ce transducteur possède donc une fenêtre glissante de longueur 3. La figure 2 illustre son fonctionnement.[5]

Afin de pouvoir conserver l'algorithme d'entraînement simple du transducteur sans mémoire, nous considérons chaque trigramme comme un symbole primitif. Néanmoins, ce choix augmente considérablement le nombre de transitions à entraîner : avec un alphabet de *n* caractères,

[5]Les symboles spéciaux @ et $ sont insérés au début et à la fin du mot afin d'obtenir un nombre suffisant de trigrammes.

@ve	ver	erm	rmu	mue	uet	ete	tet	et$
↓	↓	↓	↓	↓	↓	↓	↓	↓
v	e	r	m	u	ɛ	t	e	t

FIG. 2 – Le meilleur alignement pour le mot *vermuetet* avec le modèle à fenêtre glissante de trigrammes. L'élimination du *e* est conditionnée par le contexte *u_t*.

@g	gu	ue	et	t$
↓	↓	↓	↓	↓
g	u	ɛ	t	ɛ

@h	hu	uu	us	s$
↓	↓	↓	↓	↓
ɛ	h	a	u	s

FIG. 3 – A gauche, le meilleur alignement pour la paire *guet – gut 'bon'* avec un transducteur à bigrammes régressif. Ce type est particulièrement bien adapté aux diphtongues dont le deuxième élément est modifié, comme *ue* → *uɛ*. A droite, le meilleur alignement pour la paire *huus – Haus 'maison'* avec un transducteur à bigrammes progressif. Ce type est bien adapté aux diphtongues dont le premier élément est modifié, comme *uu* → *au*.

nous obtenons n^2 transitions pour le transducteur traditionnel (à unigrammes), mais n^4 transitions pour le transducteur à trigrammes. Étant donné les limitations de nos données d'entraînement, nous avons développé une solution intermédiaire, utilisant des fenêtres glissante de longueur 2 (bigrammes). Techniquement, il existe deux variantes de transducteurs à bigrammes : une variante régressive, générant un caractère en fonction des caractères courant et précédent, et une variante progressive, générant un caractère en fonction des caractères courant et suivant. La figure 3 en donne des exemples. Nous avons choisi de combiner les deux variantes pour nos expériences. Deux transducteurs sont entraînés à l'aide du même corpus, mais avec un biais initial favorisant soit les transitions de type $AB \rightarrow B$ pour la variante régressive, soit les transitions de type $AB \rightarrow A$ pour la variante progressive. Dans la phase d'évaluation, nous utilisons l'union des résultats des deux transducteurs.[6]

4 Données et entraînement

Comme évoqué dans l'introduction, il est difficile d'obtenir des données écrites en dialecte suisse allemand. Afin d'éviter les difficultés posées par le manque de règles orthographiques et par le style très familier de la plupart des textes, nous avons choisi un livre de littérature en dialecte bernois.

Les différences linguistiques entre l'allemand standard et le dialecte bernois concernent en grande partie les voyelles. Selon le contexte, des monophtongues allemands peuvent correspondre à des diphtongues bernois, ou l'inverse. Certains *i* allemands peuvent devenir *ü* en bernois, et les *e* finaux sont soit élidés, soit transformés en *i*. Au niveau des consonnes, les phénomènes les plus fréquents sont l'effacement du *n* final en bernois et la vocalisation du *l* préconsonantique devenant *u*. Nous espérons également capter certains phénomènes morphologiques simples comme l'alternance du suffixe diminutif (*-chen* en allemand standard, *-li* en dialecte bernois).

[6] L'union a donné des meilleurs résultats que l'intersection, aussi bien au niveau de la précision que du rappel.

De notre livre bernois, nous en avons extrait les mots sous forme d'une liste. Nous n'avons pas fait d'analyse morphologique de ces mots ; plusieurs formes fléchies du même lexème peuvent donc apparaître dans la liste. Seules les variantes morpho-phonologiques (phénomènes de *sandhi*) ont été éliminées. En plus, le corpus contenait quelques citations en langues étrangères et en allemand standard. Ces mots ont également été exclus. Les 4731 mots restants ont été traduits en allemand standard par l'auteur. Le contexte du mot dans le texte original aidait à résoudre les ambiguïtés de traduction éventuelles. La moitié de cette liste de paires de mots a été réservée pour l'entraînement des modèles, l'autre moitié pour l'évaluation.[7]

Dans la partie réservée à l'entraînement, nous avons sélectionné uniquement des paires dont la distance de Levenshtein était inférieure à 3, afin d'éviter le bruit causé par des paires non apparentées. Par exemple, nos modèles ne permettent pas de trouver la correspondance entre le mot bernois *himugüegeli 'coccinelle'* et le mot allemand standard *Marienkäferchen*.[8] Lorsque le corpus d'entraînement est relativement petit, il est crucial qu'il ne contienne aussi peu de bruit que possible. Cet élagage a réduit la taille du corpus d'entraînement de 2365 paires à 1500 paires. De plus, nous avons effectué nos expériences avec des sous-ensembles de ce corpus contenant 300 et 750 paires de mots. Les modèles ont été entraînés avec EM en 50 itérations.

Le lexique allemand standard, utilisé dans la seconde étape, comporte 202 000 formes. Les informations morphologiques et syntaxiques présentes dans le lexique n'ont pas été utilisées.

Le corpus de test contient 2366 paires de mots, dont 407 (17,2%) sont identiques en dialecte et en allemand standard. 565 mots du corpus (23,9%) ne se retrouvent pas dans le lexique allemand. Même si ces mots sont correctement induits par le modèle de similarité, ils sont éliminés en seconde étape. Il s'agit avant tout de noms composés, dont certains ont été formés *ad hoc* dans le texte littéraire. En plus, quelques mots du dialecte bernois correspondent à deux mots en allemand standard (par exemple *ir – in der 'dans la'*). Pour des raisons de complexité computationnelle, nos modèles ne trouvent pas de telles correspondances.

5 Résultats et discussion

Ci-dessus, nous avons présenté notre architecture d'induction lexicale à deux étapes. La première étape prend le mot source et génère 10 000 candidats.[9] La seconde étape valide les candidats qui se trouvent dans le lexique de la langue cible. En général, entre 0 et 20 candidats sont ainsi validés par mot source. Les coûts ou probabilités associés aux candidats lors de la première étape permettent de les ordonner (cf. figure 1).

Dans un premier temps, nous nous intéressons aux candidats situés en tête de liste. La partie gauche du tableau 1 montre combien de fois le mot allemand correct se trouve en tête de la liste des candidats. Le rappel est calculé comme suit :

$$rappel = \frac{nombre\ de\ paires\ correctes\ en\ première\ position}{nombre\ de\ paires\ dans\ le\ corpus\ d'évaluation}$$

[7] Les dictionnaires bilingues suisse allemand – allemand standard disponibles sur Internet se limitent en général aux paires de mots non apparentées. Ils constituent donc un complément intéressant à notre approche, mais ne peuvent pas servir de point de départ pour l'entraînement de nos modèles.

[8] Le seuil de 3 est arbitraire ; nous l'avons repris de (Mann & Yarowsky, 2001).

[9] Cette valeur est arbitraire. Dans (Scherrer, 2007), nous avons généré seulement 500 candidats. Il se trouvait alors que pour beaucoup de mots longs, aucun de ces candidats n'était validé dans la seconde étape. Les résultats rapportés ici ne sont donc pas directement comparables.

	Tête de liste				Toutes positions	
	N	Précision	Rappel	F-mesure	N	Rappel
Levenshtein	725	22,8	30,6	26,1	932	39,4
Unigrammes 300	840	45,8	35,5	40,0	1384	58,5
Unigrammes 750	859	44,8	36,3	40,1	1441	60,9
Unigrammes 1500	864	45,3	36,5	40,5	1446	61,1
Bigrammes 300	805	32,4	34,0	33,2	1088	46,0
Bigrammes 750	890	39,8	37,6	38,7	1239	52,4
Bigrammes 1500	930	44,4	39,3	41,7	1309	55,3
Trigrammes 1500	394	22,3	16,7	19,0	492	20,8

TAB. 1 – Paires lexicales correctement induites après la seconde étape. La partie gauche du tableau montre les résultats pour les paires correctes induites en tête de liste. La partie droite montre les résultats pour les paires correctes induites toutes positions de liste confondues. Les chiffres dans la première colonne se réfèrent à la taille du corpus d'entraînement. Étant donné les résultats du modèle Trigrammes 1500, nous avons renoncé à indiquer les chiffres des deux autres modèles à trigrammes. *N* se refère au nombre absolu de paires induites, les autres chiffres représentent des pourcentages.

Dans certains cas (surtout avec la distance de Levenshtein), plusieurs candidats se trouvent *ex aequo* en première position. Il est donc utile de calculer également la précision :

$$\textit{précision} = \frac{\textit{nombre de paires correctes en première position}}{\textit{nombre de paires en première position}}$$

La F-mesure est calculée de manière standard :

$$F = \frac{2 \cdot \textit{précision} \cdot \textit{rappel}}{\textit{précision} + \textit{rappel}}$$

La partie gauche du tableau 1 montre que les modèles adaptifs à unigrammes et à bigrammes donnent de meilleurs résultats que la distance de Levenshtein. En revanche, le modèle des trigrammes fournit des résultats décevants : la taille du corpus d'entraînement ne suffit visiblement pas pour entraîner correctement le nombre élevé de transitions de ce modèle. Si la taille du corpus d'entraînement semble avoir un impact très léger sur le modèle à unigrammes (augmentation de la F-mesure de $0,4\%$ entre le corpus à 300 paires et celui à 1500 paires), l'impact est plus prononcé pour le modèle à bigrammes (augmentation de la F-mesure de $8,5\%$). Ce résultat suggère qu'un corpus plus grand pourrait encore améliorer les performances du modèle à bigrammes, permettant d'obtenir des résultats bien meilleurs que les modèles à unigrammes.

Au lieu d'évaluer seulement les candidats apparaissant en première position de la liste, il peut également être intéressant de considérer la liste entière. Par exemple, une architecture étendue pourrait utiliser des heuristiques (fréquence des mots, contexte syntaxique des mots, ...) pour réordonner les candidats. Dans ce cas, l'ordre des candidats proposé par le modèle de similarité graphique aurait peu d'importance. On s'intéresserait donc avant tout à ce que le mot correct se trouve dans la liste, peu importe sa position. La partie droite du tableau 1 montre le nombre absolu des mots corrects apparaissant dans la liste, ainsi que le rappel.[10] Ces chiffres confirment les caractéristiques globales des modèles. En revanche, le modèle des bigrammes ne parvient

[10] Il ne nous semble pas pertinent de calculer la précision pour ce cas de figure.

pas à égaliser les résultats du modèle à unigrammes. La progression des chiffres selon la taille du corpus d'entraînement suggère cependant que les limites du modèle des bigrammes ne sont pas encore atteintes.

Les performances générales de l'induction lexicale par similarité graphique paraissent assez faibles. Sur un corpus de 2366 paires, moins de 900 paires peuvent être induites de manière fiable sans heuristiques supplémentaires. Mais comme nous l'avons déjà évoqué, il est impossible d'obtenir 100% de réussite avec l'architecture choisie. D'une part, 565 paires ne peuvent pas être induites parce qu'elles ne sont pas présentes dans le lexique cible. D'autre part, certaines paires de mots sont complètement différentes dans les deux variétés linguistiques, et il est illusoire de les induire avec des mesures de similarité graphique. Si on admet, avec (Mann & Yarowsky, 2001), que les paires avec une distance de Levenshtein inférieure à 3 sont faciles à induire, on obtient une borne de 1256 paires de mots présents dans le lexique cible et faciles à induire. Selon les chiffres se référant à toutes les positions de liste, les modèles à bigrammes atteignent cette borne, et les modèles à unigrammes la dépassent même. De plus, on constate que jusqu'à 70% des paires faciles à induire sont correctement induites en première position.

Nous avons expliqué (Scherrer, 2007) que les études précédentes (Mann & Yarowsky, 2001) obtiennent de bien meilleurs résultats grâce à une méthode d'évaluation moins sévère. Cette méthode consiste à trouver les paires étant donné une liste de 100 mots de la langue source et la liste (en désordre) des 100 mots correspondants de la langue cible. Ils obtiennent autour de 67% de réussite sur le vocabulaire espagnol-portugais complet. Avec la même méthode d'évaluation, nos modèles atteignent des chiffres entre 85% et 89% avec les unigrammes, et entre 71% et 81% avec les bigrammes. Sur le vocabulaire des mots apparentés (distance de Levenshtein inférieure à 3), Mann & Yarowsky obtiennent des chiffres de 92%. Dans cette tâche, les performances de nos modèles à unigrammes et à bigrammes se situent entre 91% et 98%.

6 Conclusion

Nos expériences ont montré que les mesures de similarité graphique peuvent faciliter l'induction lexicale lorsque les deux langues sont étroitement apparentées. En plus, nous avons montré que l'utilisation de méthodes d'apprentissage automatique permet d'améliorer nettement les performances par rapport à des modèles génériques comme la distance de Levenshtein. Le modèle à unigrammes fournit de bons résultats avec des corpus d'entraînement très petits. Le modèle utilisant une fenêtre glissante de bigrammes permet de faire des prédictions plus ciblées, augmentant ainsi le rappel. Cependant, ce modèle nécessite des corpus d'entraînement plus grands à cause du nombre plus élevé de transitions à entraîner. Nos expériences suggèrent que des corpus de plus de 1500 paires de mots pourraient améliorer davantage les performances du modèle à bigrammes. Des recherches futures devraient montrer si tel est le cas. En revanche, le corpus de 1500 paires s'est révélé clairement insuffisant pour entraîner un modèle à trigrammes. Il reste à voir si un corpus plus grand permettrait à ce modèle de dépasser les performances du modèle à bigrammes.

Nous avons constaté que le lexique de la langue cible, que nous utilisons comme modèle de langue simple, est insuffisant, car une grande partie des mots composés ne s'y trouve pas. Pour remédier à cette lacune, il pourrait être avantageux de ne pas utiliser ce lexique directement, mais plutôt de manière indirecte pour créer un modèle de langue à n-grammes de lettres, à l'instar de (Claveau, 2007).

Les résultats suggèrent également qu'il peut être intéressant d'inclure d'autres heuristiques afin de sélectionner la bonne traduction dans la liste des candidats. En particulier, l'architecture présentée ici ne tient compte ni de l'information contextuelle riche encodée dans les textes, ni des informations morphologiques et syntaxiques contenues dans le lexique allemand utilisé dans la deuxième étape. L'intégration de ces informations nous paraît prometteuse, d'autant plus que celles-ci sont facilement disponibles pour notre paire de langues. Les méthodes basées sur la similarité graphique peuvent donc être utilisées avec profit dans la tâche d'induction lexicale pour des langues apparentées sans pour autant exiger de grandes quantités de données d'entraînement.

Remerciements

Nous remercions Paola Merlo pour son soutien et ses commentaires précieux au cours de cette recherche. Nous aimerions aussi remercier Éric Wehrli pour sa permission d'utiliser le lexique allemand du projet *Fips*.

Références

BROWN P. F., PIETRA V. J. D., PIETRA S. A. D. & MERCER R. L. (1993). The mathematics of statistical machine translation : parameter estimation. *Computational Linguistics*, **19**(2), 263–311.

CLAVEAU V. (2007). Inférence de règles de réécriture pour la traduction de termes biomédicaux. In *Actes de TALN 2007*, p. 111–120, Toulouse, France.

CLAVEAU V. & ZWEIGENBAUM P. (2005). Traduction de termes biomédicaux par inférence de transducteurs. In *Actes de TALN 2005*, p. 253–262, Dourdan, France.

JANSCHE M. (2001). Re-engineering letter-to-sound rules. In *Proceedings of NAACL'01*, Pittsburgh, PA, USA.

JIAMPOJAMARN S., KONDRAK G. & SHERIF T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of NAACL'07*, p. 372–379, Rochester, NY, USA.

MANN G. S. & YAROWSKY D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL'01*, Pittsburgh, PA, USA.

RAPP R. (1999). Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of ACL'99*, p. 519–526, Maryland, USA.

RISTAD E. S. & YIANILOS P. N. (1998). Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(5), 522–532.

SCHERRER Y. (2007). Adaptive string distance measures for bilingual dialect lexicon induction. In *Proceedings of ACL'07, Student Research Workshop*, Prague, République Tchèque.