Rapport de stage

Analyse d'un algorithme d'alignement multilingue

Romain VERSAEVEL, L3 Informatique Fondamentale, ENS de Lyon

Encadré par M. François YVON, directeur du LIMSI/CNRS

9 juillet 2014

Résumé

Ce rapport rend compte de mon stage de Licence 3 réalisé au LIMSI/CNRS, durant lequel j'ai étudié l'algorithme d'alignement multilingue *anymalign*.

Après une présentation du domaine de recherche, le traitement automatique des langues parlées, et plus particulièrement la traduction automatique, je propose des résultats pratiques et théoriques qui valident l'algorithme *anymalign* et en montrent certaines limites.

Table des matières

1 Introduction

J'ai suivi dans le cadre de ma formation, en Licence 3 d'Informatique à l'ENS de Lyon un stage de recherche d'une durée de six semaines, du 2 juin au 11 juillet 2014. Ce stage s'est déroulé au LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur), laboratoire CNRS situé sur le campus de l'université Paris-Sud, à Orsay, dans le groupe TLP (Traitement des Langues parlées, ou Spoken Language Processing Group). J'étais encadré par M. François Yvon, directeur du LIMSI.

Le sujet de ce stage était d'analyser l'algorithme d'alignement multilingue *anymalign*, conçu et implémenté par en , et utilisé par le LIMSI pour diverses applications.

Ce rapport est divisé en quatre parties. Dans la première, je présente le contexte dans lequel s'inscrit mon travail, le domaine de la traduction automatique. Dans la deuxième, je présente l'algorithme que j'ai étudié et l'analyse qui en avait déjà été réalisée. Dans les troisième et quatrième, je présente les résultats de ma propre analyse, empirique (confrontation avec des mesures d'association) et théorique.

2 Contexte, rappels

Cette section présente le domaine de recherche dans lequel s'inscrit mon stage : le traitement automatique des langues parlées, puis plus particulièrement la traduction automatique. Elle est essentiellement bibliographique.

2.1 Traitement des langues parlées

Le traitement automatique des langues parlées, en anglais *spoken language processing* (ou encore linguistique automatique, *computational linguistics*) est une discipline à l'intersection de l'informatique, de la linguistique, des sciences cognitives. Son objet est l'étude et l'utilisation des langues naturelles avec des outils informatiques.

Ses applications sont nombreuses. On peut citer le traitement du signal pour la simulation et la reconnaissance vocale, les interfaces homme/machine, l'écriture automatique , et la traduction automatique.

2.2 Traduction automatique

La *traduction automatique*, en anglais *machine translation*, a pour objet la traduction de textes d'une langue naturelle vers une autre, par l'intermédiaire d'algorithmes et d'ordinateurs (dans intervention humaine — on parle sinon de *traduction assistée par ordinateur*). Si l'on peut chercher les origines de cette discipline dans les langues universelles imaginées par Descartes et Leibniz au XVII^{e} siècle , par le truchement desquelles on pourrait passer de n'importe quelle langue à n'importe quelle autre, c'est en 1947 que le mathématicien américain Warren Weaver propose pour la première fois d'utiliser les ordinateurs pour réaliser des travaux de traduction à l'UNESCO. La discipline a depuis fait des progrès considérables. Ses motivations, développées par John Hutchins dans sont nombreuses. Il évoque la nécessité dans certaines professions de consulter des documents rédigés dans des langues très diverses, la simplification de la transmission des savoirs, des applications militaires, mais aussi des raisons plus idéologiques : la traduction permet la communication et donc la paix. Notons que la traduction automatique ne fait pas concurrence aux traducteurs humains, qui s'adressent à un public différent. Sa vocation n'est pas de réaliser des traductions littéraires ou des traductions parfaites, mais de rendre accessible, rapidement et à moindre coût, une très grande quantité de documents à des locuteurs de toutes les langues.

La traduction est un exercice considéré difficile depuis longtemps. « Traduttore, traditore » disent les italiens : « Traduire, c'est trahir ». Les ordinateurs n'ont pas des prétentions si élevées que celle de transmettre fidèlement la pensée de l'auteur original ; ils doivent cependant relever toutes sortes de défis. Ainsi l'idée naïve d'une traduction mot à mot à l'aide d'un simple dictionnaire bilingue produit-elle de très mauvais résultats, parce qu'un mot dans une langue peut se traduire par plusieurs dans une autre (*Rindfleischetikettierungsüberwachungsaufgavenübertragungsgesetz* est ainsi l'équivalent allemand de *loi sur le transfert des obligations de surveillance de l'étiquetage de la viande bovine* en français, soit 1 mot contre 15), parce que la polysémie ou l'homonymie ne sont pas prises en compte (faut-il traduire *avocat* par *avocado* ou *lawyer* ? *cut* par *couper*, par *coupé*, par *coupions* ?), non plus que les idiomes (*Das ist nicht mein Bier* doit être traduit par *Ce ne sont pas mes oignons* et non littéralement par *Ce n'est pas ma bière*), etc.

Bon nombre d'outils de traduction automatique ont été développés, les plus connus étant Google translate et SYSTRAN (utilisé par Yahoo et BabelFish). Les approches sont nombreuses, elles peuvent s'appuyer ou non sur des outils d'analyse linguistique, être unilatérale ou bi-

latérales, statistiques ou non. . .

Les algorithmes qui nous intéressent plus particulièrement ici ne sont pas des algorithmes de traduction pure mais des algorithmes d'*alignement*, qui génèrent des données pour les premiers en analysant des corpus édités en plusieurs langues. On cite souvent l'exemple de Champollion qui apprit à déchiffrer les hiéroglyphes grâce à la pierre de Rosette. Les algorithmes d'alignement utilisent ainsi des corpus multilingues — dont il existe grâce à Internet de grandes quantités — pour en extraire des relations de traduction entre des paires de mot. Le plus utilisé est Giza++.

2.3 Quelques définitions et notations

Avant d'étudier plus avant *anymalign*, il convient de poser quelques définitions.

Dans un texte en langue naturelle, on appelle *segment* (*phrase* en anglais) un ensemble de mots. On appellera en outre n-gramme un segment de n mots consécutifs.

On appelle *corpus multilingue* un ensemble de textes en plusieurs langues. On appelle *corpus parallèle* un corpus multilingue dont les textes sont traduction les uns des autres. Enfin, un corpus parallèle est *aligné* lorsque sont déterminées des relations (de traduction) entre segments de ses textes.

anymalign considère des corpus parallèles de taille arbitraire alignés au niveau des phrases, et construit un dictionnaire d'alignements de segments plus petits. On limitera ici notre analyse à des corpus de deux seulement textes, qu'on appellera *texte source* et *texte cible* .

3 Présentation d'*anymalign*

3.1 L'algorithme

3.1.1 (bis)

La fonction auxiliaire $ajouter$ incorpore le mot w dans la table $Profils$, initialement vide, de telle sorte que :

– si $Profils$ contient déjà $w' \subset w$ de même profil que w, alors w' est écrasé par w ;
– si plusieurs segments d'un même texte ont le même profil, ils sont finalement tous rejetés.

Algorithme 1 : anymalign

Entrées : Corpus parallèle $(C_1, \ldots, C_n)$.

Sorties : Table d'alignements Alignements.

Début

Pour i *de* 1 *à* I **faire**
S ← sous-corpus de taille |S|
Pour j *de* 1 *à* J **faire**
Pour tout j-*gram* w *dans* S **faire**
Calculer son profil (vecteur de présence)
`ajouter` (w, Profils)
Pour tout *profil dans* Profils **faire**
Pour tout *groupe de mots* $(w_{n_1}, \ldots, w_{n_k})$ **faire**
Alignements$[(w_{n_1}, \ldots, w_{n_k})]$ ++
retourner Alignements

Fin

3.1.2 Exemple

3.1.3 Remarques

Tel que présenté ici, l'algorithme termine après I itérations ; en réalité, dans l'implémentation qui en a été faite, de nombreuses conditions d'interruption différentes peuvent être utilisées : après un certain temps, lorsque le nombre d'alignements par secondes devient inférieur à une certain borne, ou encore par une interruption manuelle de l'utilisateur.

anymalign est comme on le voit un algorithme statistique, dont l'exécution ne dépend pas de la taille du corpus ; il utilise les seuls textes parallèles (et aucune donnée linguistique) ; c'est en outre un algorithme complètement symétrique : tous les corpus C_i jouent le même rôle. Pour cette raison, on peut se contenter pour l'analyse de deux corpus C_1 et C_2, mais il s'agit évidemment d'une qualité importante de l'algorithme.

3.2 Qualités et défauts

(L'analyse déjà réalisée)

4 Mesures d'association

Dans cette section, après avoir introduit la notion de *mesure d'association* en statistiques, je compare les résultats empiriques fournis par *anymalign* avec différentes mesures d'association classiques.

4.1 Topo sur les mesures d'association

En statistiques, on appelle *mesure d'association* une relation entre plusieurs variables aléatoires non-indépendantes. Le terme est proche de celui de *corrélation* même si ce dernier a une définition plus contrainte, impliquant notamment d'un *facteur de corrélation*.

En théorie de l'information, il s'agit le plus souvent de comparer deux variables aléatoires X et Y. Notons e et f les événements $X \in E$ et $Y \in F$. Les mesures d'association entre X et Y utiliseront généralement les valeurs de la table de contingence ($\mathfrak{F}(e)$ désigne la fréquence absolue de l'événement e dans un effectif de taille totale N) :

$a := \mathfrak{F}(e \wedge f)$	b := $\mathfrak{F}(e \wedge \overline{f})$	$\mathfrak{F}(e)$
$c := \mathfrak{F}(\overline{e} \wedge f)$	$d := \mathfrak{F}(\overline{e} \wedge \overline{e})$	$\mathfrak{F}(\overline{e})$
$\mathfrak{F}(f)$	$\mathfrak{F}(\overline{f})$	N

Ici, X et Y seront des mots ; E et F désigneront les phrases alignées d'un corpus parallèle de taille N. On mesurera donc l'association de deux mots à travers la probabilité qu'ils apparaissent dans les mêmes phrases (a), le premier mais pas le deuxième (b), etc.

Les mesures utilisées sont listées dans le tableau suivant (les noms sont donnés en anglais car certains n'ont pas d'équivalent français) :

Nom	Formule
Likelihood ratio ou G^2	$2\left(a\log\left(\frac{aN}{(a+b)(a+c)}\right)+b\log\left(\frac{bN}{(a+b)(b+d)}\right)+c\log\left(\frac{cN}{(a+c)(c+d)}\right)+d\log\left(\frac{dN}{(b+d)(}\right.\right.$
Pointwise mutual information (PMI)	$\log_2(\frac{aN}{(a+b)(a+c)})$
Φ^2 de Church & Gale	$\frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)}$
Coefficient Q de Yule	$\frac{ad-bc}{ad+bc}$
Coefficient ω de Yule	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$
Jaccard	$\frac{a}{a+b+c}$
Normalized expectation	$\frac{2a}{2a+b+c}$
Salience	$\log\left(\frac{aN}{(a+b)(a+c)}\right)\cdot\log(a)$
t test	$\frac{a-\frac{(a+b)(a+c)}{N}}{\sqrt{a(1-\frac{a}{N})}}$
z score	$\frac{a-\frac{(a+b)(a+c)}{N}}{\sqrt{\frac{(a+b)(a+c)}{N}(1-\frac{(a+b)(a+c)}{N^2})}}$
Braun-Blanquet	$\frac{a}{\max(a+b,a+c)}$
Simpson	$\frac{a}{\min(a+b,a+c)}$
Laplace	$\frac{a+1}{\min(b,c)+a+2}$

On constate une grande diversité dans ces mesures, permise par la définition assez laxiste. Le principe est que la mesure est d'autant plus élevée que les les variables aléatoires sont dépendantes suivant un certain schéma. Notre objectif est de montrer que les paires de mots fréquemment alignées par *anymalign* ont une association forte. Il s'agit donc de comparer deux mesures d'association, puisque *anymalign* en réalise aussi une, quoique plus compliquée que les précédentes. En effet, le rapport du nombre de fois qu'un couple de mots est aligné par le nombre d'itérations converge vers la probabilité que ce couple ait le même profil dans un sous-corpus tiré aléatoirement.

4.2 Comparaison avec *anymalign*

Dans cette partie, on compare les résultats d'*anymalign* avec les mesures d'association listées précédemment.

4.2.1 *anymalign*

Ces résultats ont été obtenus en traitant un corpus français-anglais de 1000 lignes extrait d'Europarl . L'exécution d'*anymalign* a été interrompue après de 2 200 000 itérations ; le nombre de

nouveaux alignements par seconde était alors inférieur à 1 ; un peu plus de 45 000 paires de mots ont ainsi été alignées.

Les paires les plus alignées correspondent aux mots-outils, très fréquents :

Français	Anglais	Nombre d'alignements
.	.	6185271
et	and	394766
je	I	330634
rapport	report	329316
Commission	Commission	286020
concurrence	competition	277991

Les alignements erronés les plus fréquents font eux aussi intervenir des mots-outils, présents dans une grande majorité des phrases ; le premier d'entre eux est (« . »,« the »), au 103^{e} rang avec 25031 alignements.

1451 mots sont alignés plus de 1000 fois, 8266 mots plus 100 fois, 28264 mots plus de 10 fois.

Le critère de comparaison qui a été retenu pour comparer *anymalign* aux autres mesures d'association est le nombre d'alignements produits.

4.2.2 Jaccard

L'une des mesures d'association produisant les meilleurs résultats est le Jaccard dont on rappelle l'expression : $Jacc(e,f) = \frac{a}{a+b+c} = \frac{\mathfrak{F}(e \wedge f)}{\mathfrak{F}(e \wedge f)+\mathfrak{F}(e \wedge \overline{f})+\mathfrak{F}(\overline{e} \wedge f)}$ ou, si l'on note E (resp. F) l'ensemble des phrases où apparaît e (resp. f) : $Jacc(e,f) = \frac{E \cap F}{E \cup F}$.

La figure 4.2.2 montre le nombre d'alignements réalisés par *anymalign* en fonction du Jaccard. On y observe une association importante : pour j fixé, les couples de mots de Jaccard proche de j les plus alignés le sont d'un nombre proche de $C \cdot j$ (C une constante, ici égale à , mais qui est de peu d'importance puisqu'elle dépend de la durée d'exécution de l'algorithme.) Ainsi, l'absence de points dans les coins supérieur gauche (les paires d'association faible sont peu alignées) et inférieur droit (les paires de forte association sont beaucoup alignées) valident l'algorithme.

En revanche, l'épaisseur de la courbe, en d'autres termes le grand nombre de points situés endessous de la courbe $y = C \cdot x$ montre une variation importante dans les alignements réalisés par *anymalign* pour des mots de même Jaccard

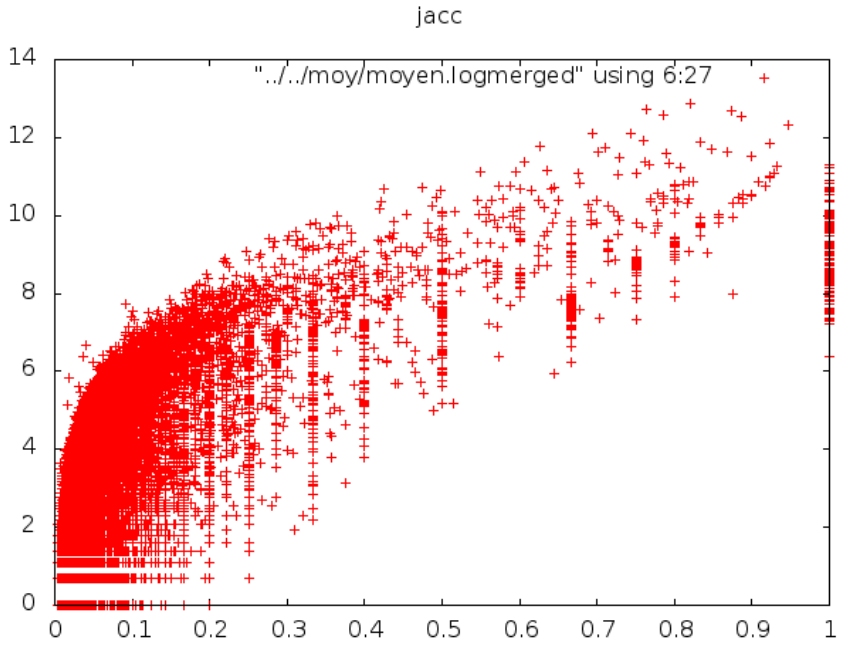

FIGURE 1 – LEGENDE

Ces résultats montrent une ressemblance importante entre les résultats d'*anymalign* et le Jaccard ; l'algorithme en donne une estimation On aurait cependant espéré une proximité encore supérieure : la motivation de ces travaux se trouve dans

4.2.3 PMI

L'une des mesures d'association produisant les meilleurs résultats est l'Information Mutuelle (IM) Jaccard dont on rappelle l'expression : $IM(e,f) = \log_2\left(\frac{aN}{(a+b)(a+c)}\right) = \log_2\left(\frac{\mathfrak{F}(e\wedge f)\cdot N}{\mathfrak{F}(e)\cdot\mathfrak{F}(f)}\right)$ (avec N la taille totale du corpus).

La figure 4.2.2 montre le nombre d'alignements réalisés par *anymalign* en fonction du Jaccard. On y observe une association importante : pour j fixé, les couples de mots de Jaccard proche de j les plus alignés le sont d'un nombre proche de $C \cdot j$ (C une constante, ici égale à , mais qui est de peu d'importance puisqu'elle dépend de la durée d'exécution de l'algorithme.) Ainsi, l'absence de points dans les coins supérieur gauche (les paires d'association faible sont peu alignées) et inférieur droit (les paires de forte association sont beaucoup alignées) valident l'algorithme.

- Dépend trop de la fréquence (mots rares à droite, mot fréquents à gauche).

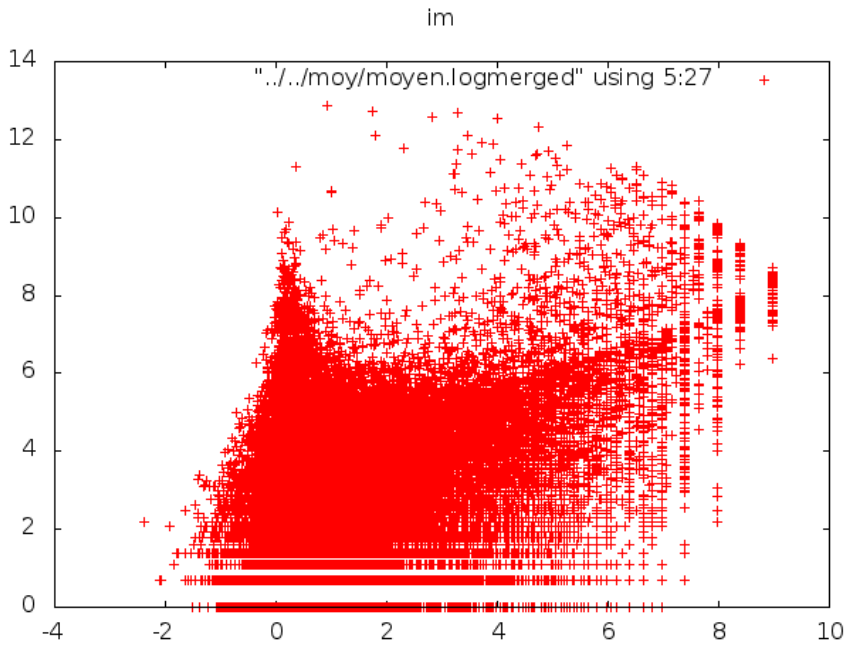

FIGURE 2 – LEGENDE

- A droite : log(N)=9,96 ; mots rares équivalents

- 0 et en-dessous : mots assez fréquents mais sans intérêt. Malheureusement, beaucoup d'alignements juste au-dessus de 0...

- Les bons alignements entre mots fréquents sont entre 1 et 4 ; les bons alignements moyen

- Bilan : sans invalider l'algorithme, le pic autour de 0 trahit la présence importante de "déchet"

4.2.4 Conclusion

5 Analyse théorique

Dans cette partie, on cherche à analyser *anymalign* à l'aide de quelques calculs de probabilités.

La notion qu'il m'a paru intéressant d'étudier est celle de profil, qui est intrinsèquement liée au fonctionnement d'*anymalign*.

On considère un sous-corpus quelconque S, composé de $|S|$ phrases parallèles dont on fera l'hypothèse simplificatrice qu'elles sont toutes de longueur m. Soient Π un profil (c'est-à-dire

un élément de $\{0;1\}^{|S|}$), et w un mot de fréquence φ.

Notons $\mathcal{P}_{w,\Pi}$ la probabilité que w ait le profil Π dans S . La longueur des phrases étant constante, elle ne dépend que de $|\Pi|$; c'est la probabilité que w apparaisse au moins une fois dans exactement $|\Pi|$ des $|S|$ phrases du sous-corpus. En faisant l'hypothèse que les mots sont tirés au hasard d'après leur seule fréquence, la probabilité que w n'apparaisse pas dans une phrase donnée est $(1-\varphi)^m$ et donc

$$\begin{aligned}\mathcal{P}_{w,\Pi} &= (1-\varphi)^{m(|S|-|\Pi|)} \cdot (1-(1-\varphi)^m)^{|\Pi|} \\ &= (1-\varphi)^{m|S|} \cdot \left[(1-\varphi)^{-m}-1\right]^{|\Pi|}\end{aligned}$$

Soient maintenant w et w' deux mots de fréquences respectives φ et ψ. Notons $\mathcal{A}_{w,w'}$ la probabilité que ces deux mots aient le même profil et $\check{\mathcal{A}}_{w,w'}$ la probabilité que ces deux mots aient le même profil *non-nul* dans S.

On a alors sous *l'hypothèse que w et w' sont indépendants* :

$$\begin{aligned}\mathcal{A}_{w,w'} &= \sum_{\Pi\ profil} \mathcal{P}_{w,\Pi} \cdot \mathcal{P}_{w',\Pi} \\ &= \sum_{\Pi\ profil} (1-\varphi)^{m|S|} \cdot ((1-\varphi)^{-m}-1)^{\Pi} \cdot (1-\psi)^{m|S|} \cdot ((1-\psi)^{-m}-1)^{\Pi} \\ &= \sum_{\Pi\ profil} [(1-\varphi)(1-\psi)]^{m|S|} \cdot [((1-\varphi)^{-m}-1)((1-\psi)^{-m}-1)] \\ &= \sum_{k=0}^{|S|} \sum_{|\Pi|=k} [(1-\varphi)(1-\psi)]^{m|S|} \cdot [((1-\varphi)(1-\psi))^{-m}-(1-\varphi)^{-m}-(1-\psi)^{-m}+1]^{|\Pi|} \\ &= \sum_{k=0}^{|S|} \binom{|S|}{k} [(1-\varphi)(1-\psi)]^{m|S|} \cdot [((1-\varphi)(1-\psi))^{-m}-(1-\varphi)^{-m}-(1-\psi)^{-m}+1]^{k} \\ &= [(1-\varphi)(1-\psi)]^{m|S|} \cdot [((1-\varphi)(1-\psi))^{-m}-(1-\varphi)^{-m}-(1-\psi)^{-m}+2]^{|S|} \\ &= [1-(1-\varphi)^m-(1-\psi)^m+2(1-\varphi)^m(1-\psi)^m]^{|S|}\end{aligned}$$

et

$$
\begin{aligned}
\check{\mathcal{A}}_{w,w'} &= \sum_{\Pi \neq 0} \mathcal{P}_{w,\Pi} \cdot \mathcal{P}_{w',\Pi} \\
&= \sum_{k=1}^{|S|} \binom{|S|}{k} [(1-\varphi)(1-\psi)]^{m|S|} \cdot [((1-\varphi)(1-\psi))^{-m} - (1-\varphi)^{-m} - (1-\psi)^{-m} + 1]^k \\
&= [(1-\varphi)(1-\psi)]^{m|S|} \cdot [[((1-\varphi)(1-\psi))^{-m} - (1-\varphi)^{-m} - (1-\psi)^{-m} + 2]^{|S|} - 1] \\
&= [1-(1-\varphi)^m - (1-\psi)^m + 2(1-\varphi)^m(1-\psi)^m]^{|S|} - [(1-\varphi)(1-\psi)]^{m|S|}
\end{aligned}
$$

5.1 Alignement de e et f

Soient e et f deux mots de deux langues différentes (de fréquences respectives φ et ψ). On va supposer que toutes les phrases du corpus ont une longueur 20 et qu'*anymalign* sélectionne des sous-corpus de taille $|S| = 15$. L'espérance du nombre d'alignements de e et f en k étapes est $\mathbb{E}[Al(e,f)] \leq k\check{\mathcal{A}}_{e,f}$ (l'inégalité vient du fait qu'avoir le même profil non-nul est une condition nécessaire mais pas suffisante pour que e et f soient alignés : il faut aussi qu'ils soient les seuls dans leur langue respective à posséder ce profil).

Ainsi, si l'on suppose l'indépendance de e et f, que $k = 10^6$, et que $\varphi = \psi$ (ce qui maximise $\check{\mathcal{A}}_{e,f}$), on obtient les soit $\mathbb{E}[Al(e,f)] \leq 27210$.

φ	$\mathbb{E}[Al(e,f)] \leq \ldots$	*anymalign*

5.2 Alignement de n-grams

5.3 Autres pistes

6 Conclusion

7 Bibliographie

8 Annexes