Modèle de document pour TALN 2014

Untel Trucmuche[1, 2] Unetelle Machinchose[1, 3]
(1) LPL, AMU, CNRS, 5 avenue Pasteur, 13100 Aix-en-Provence
(2) LIF, AMU, CNRS, 163 avenue de Luminy, 13288 Marseille Cedex 9
(3) Lab, adresse, CP Ville, Pays
utrucmuche@lpl-aix.fr, umachinchose@adresse-academique.fr

Résumé. Ici, un résumé en français (max. 150 mots).

Abstract. The DBnary project aims at providing high quality Lexical Linked Data extracted from different Wiktionary language editions. Data from 10 different languages is currently extracted for a total of over 3.16M translation links that connect lexical entries from the 10 extracted languages, to entries in more than one thousand languages. In Wiktionary, glosses are often associated with translations to help users understand to what sense they refer to, wether through a textual definition or a target sense number. In this article we aim at the extraction of as much of this information as possible and then the disambiguation of the corresponding translations for all languages available. We use an adaptation of various textual and semantic similarity techniques based on partial or fuzzy gloss overlaps to disambiguate the translation relations (To account for the lack of normalization, e.g. lemmatization and PoS tagging) and then extract some of the sense number information present to build a gold standard so as to evaluate our disambiguation as well as tune and optimize the parameters of the similarity measures. We obtain F-measures of the order of 80% (on par with similar work on English only), across the three languages where we could generate a gold standard (French, Portuguese, Finnish) and show that most of the disambiguation errors are due to inconsistencies in Wiktionary itself that cannot be detected at the generation of DBnary (shifted sense numbers, inconsistent glosses, etc.).

Mots-clés : Ici une liste de mots-clés en français.

Keywords: Wiktionary, Linked Open Data, Multilingual Resources.

1 Introduction

Wiktionary est un ressource lexico-sémantique construite collaborativement sous l'égide de la Fondation Wikimedia (qui héberge également la célèbre initiative Wikipedia). C'est actuellement la ressource collaborative de données lexicales la plus grande. Les pages Wiktionary décrivent habituellement des entrées lexicales en donnant leut catégorie grammaticale, un ensemble de définitions, des exemples, des relation lexico-sémantiques ainsi que des traductions dans plus de mille langues cible.

Le projet DBNary (Sérasset, 2012) a pour objectif de fournir des données liées lexicales de haute qualité extraites des éditions en différentes langues de Wiktionary. DBNary permet actuellement d'extraire des données issues de 10 éditions et regroupe $3,16$M de liens de traduction qui mettent en relation les entrées lexicales des 10 langues extraites vers des entrées dans plus de mille langues. Ces chiffres sont en augmentation constante, sachant que le jeu de données DBNary et extrait dès que Wikimedia met à disposition de nouvelles vidanges **? ? ?** des données (environs tous les 10 à 15 jours pour chaque langue).

La source de ces liens de traduction sont des *entrées lexicales*. Le but de ce travail est d'attacher ces traductions au sens de mot correct correspondant et d'ainsi augmenter la valeur et la qualité de DBNary. Des travaux similaires ont été menés (principalement dans le jeu de données Uby), mais sont limités à l'anglais et l'allemand. Dans cet article, nous avons travaillés des éditions dans 9 langues, avec lesquelles nous avons du faire face avec les habitudes diverses des différentes communautés Wiktionary qui s'exprimaient par différentes propriétés linguistiques mises en avant.

Après une revue des travaux similaires, nous présentons la structure de DBNary. Ensuite, après avoir montré comment nous construisons l'étalon-or endogène que nous utilisons pour évaluer notre travail, nous détaillons les méthodes employées pour atteindre notre but. Enfin, nous évaluons notre méthode est interprétons les résultats.

2 Travaux Similaires

2.1 Extraction de données depuis les éditions de langue Wiktionary

Depuis sa création en 2002, Wiktionary à connu une augmentation régulière en taille (à la fois par un travail collaboratif ainsi que l'insertion automatique de données lexicales libres précédemment disponible). L'engouement pour Wiktionary en tant que source pour la production de données lexicales pour des applications du TAL c'est développé rapidement , et des études comme par exemple (Zesch *et al.*, 2008b) où (Navarro *et al.*, 2009) démontrent la richesse et la puissance de ces ressources.

Depuis, les travaux se sont surtout concentrés sur l'extraction systématique de données provenant de Wiktionary. Ls plus part comme ressource spécifique à un projet donné, et qui ne constituent ainsi qu'une capture de Wiktionary figée dans le temps. Sachant que toutes les éditions de Wiktionary évoluent régulièrement (et indépendamment) vis-à-vis de comment sont représentées leur données, de tels travaux ne peuvent pas fournir un accès durable aux données de Wiktionary.

Certains des travaux cependant dont également maintenus et permettent un accès au cours du temps. L'un de plus mature de ces travaux est l'API *JWKTL* (Zesch *et al.*, 2008a) qui donne accès aiux éditions Anglaises, Allemandes et Russes. C'est cette dernière qui est utilisée dans le projet UBY (Gurevych *et al.*, 2012), qui met à disposition une version LMF de ces éditions.

Il faut également faire mention du projet wikokit (Krizhanovsky, 2010), qui donne accès aux éditions Anglaises et Allemandes, et qui à été utilisée par *JWKTL*.

(Hellmann *et al.*, 2013) présentent un autre tentative, sous l'égide du projet dbpedia (Lehmann *et al.*, 2014), dont l'objectif est en particulier de fournir un accès aux données de Wiktionary en tant que données liées ouvertes. La raison principale que rend cette approche intéressante, est l'aspect collaboratif, utilisé pour créer les patrons d'extraction, ce qui correspond à l'approche générale du projet dbpedia. Ce projet donne accès aux éditions de Wiktionary Anglais, Française, Russe et Allemande.ditions.

Cet article et les travaux effectués ici le sont dans le cadre du projet DBnary (Sérasset, 2012), dont l'objectif est similaire à celui de (Hellmann *et al.*, 2013). Plus précisément, notre objectif est de fournir une base de donnée lexicale au format LEMON, qui structure les données comme dans des lexiques traditionnels. En effet, nous extrayons des données issues de Wiktionary, en nous restreignant toutefois aux donnée "natives" de chaque édition. Par exemple, nous extrayons les données en Français de l'édition Française, mais ignorons les données en Français contenues dans d'autre éditions. À notre connaissance, DBNary est actuellement l'extracteur pour Wiktionary le plus avancé, avec sont support actif courant de 12 langues. C'est également le seul projet qui donne accès à tous l'historique des données extraites.

2.2 Désambiguïsation des sources des liens de traduction

En ce qui concerne le rattachement des liens de traduction aux sens de mots les plus adéquats (déambiguïsation des liens de traduction), les travaux les plus similaires sont ceux de (Meyer & Gurevych, 2012b), dont les objectifs correspondent aux nôtres. Cependant leurs travaux ne portent que sur les éditions Anglaises et Allemandes. De plus, l'étalon-or utilisé pour évaluer leur méthode fut créé manuellement et est d'une taille significativement plus petite que l'étalon-or endogène que nous avons ici extrait de la ressource elle-même. Leur travail utilise également une stratégie de repli (vers le sens le plus fréquent), quand leur heuristiques à base de mesures de similarité ainsi que basées sur la structure de la ressource échouent. Les autres heuristiques qu'ils utilisèrent, impliquent également une analyse plus fine des définissions et gloses afin de notamment faire une distinction entre les étiquettes linguistiques (domaine, registre, titre, etc.).

Dans le travail ici présent, nous atteignons des résultats similaires sur les langues où nous avons la possibilité d'évaluer la désambiguïsation avec un étalon-or endogène, malgré le fait que nous n'aillons uniquement utilisés des mesures de similarité de chaines et de mots, et ce même dans les langues avec des propriétés moins communes (par exemple la nature agglutinante du Finnois.)

2.3 Mesures de similarité

Notre méthode est basée sur l'application de mesures d'intersection de gloses et de leurs extensions avec des idées provenant des mesures de similarité textuelles hybrides, qui calculent une correspondance de sous-séquences à la fois au niveau du caractère et du mot. Dans les travaux cités ci-dessus, (Meyer & Gurevych, 2012b), une mesure de similarité à base de traits est utilisée (intersection de gloses), alors que dans leurs travaux antérieurs (Meyer & Gurevych, 2010), ils ont utilisés une mesure de similarité textuelle basée sur des espaces vectoriels générés à partir de corpus (Analyse Sémantique Explicite).

Dans notre travail, nous proposons l'utilisation d'une mesure de similarité simple, où nous remplaçons la correspondance de mots exacte du calcul d'intersection par une mesure de distance de chaine approchée, et en nous plaçant dans le contexte plus général de la mesure de similarité qu'est l'indice de Tversky (qui peut être vu comme une généralisation de Lesk, du coefficient de Dice, des indices de Jaccard et Tatimono, etc.)

L'idée de *cardinalité molle*("soft-cardinality") proposée par (Jimenez *et al.*, 2010, 2012) est très similaire, dans le sens où elle exploite l'index de Tversky comme base et la conjugue avec une mesure de similarité textuelle. C'est à dire, au lieu d'incrémenter la cardinalité de l'intersection de 0 où 1, elle est incrémentée par la valeur retournée par une mesure de similarité de chaine pour chaque paire de mots considérée lors du calcul de la cardinalité de l'intersection.

Leur mesure est basée sur la notion de q-grammes, générés empiriquement (caractère-grammes correspondant à des sous-chaines) avec une pondération à base de contenue d'information mutuel ponctuel (pointwise mutual information). Dans note cas, construire ce type de modèles pour 12 langues nécessiterais de nombreux efforts, et avec l'extension future à plus de langues (voire toutes), cela deviendrait une tâche pratiquement impossible.

De ce fait, nous avons choisis une mesure de distance de chaine simple pour le calcul de la correspondance partielle de chaines. Cependant, il y a de nombreuses mesures disponibles et il advient de choisir celle qui est la plus appropriée pour notre tâche (voir Section 5). Qui plus est, il existe également des mesures dites de "Niveau 2" qui combinent déjà de différentes manières des mesures de similarité de chaines avec des mesure d'intersection de termes. Ainsi il faudra évaluer la méthode que nous proposons, avec certaines de ces mesures de "Niveau 2" existantes afin d'estimer le viabilité. Toutes ces mesures ont fait l' objet d'une évaluation et d'une comparaison extensive entre elles dans le contexte d'une tâche de correspondance de noms (Cohen *et al.*, 2003).

3 Le jeu de données DBNary

DBnary est un jeu de données liées ouvertes extraites depuis 12 éditions de langues Wiktionary (Anglais, Finnois, Français, Allemand, Grec, Italien, Japonais, Portugais, Russe, Turque, Espagnol, Bulgare). La ressource est disponible en ligne à l'adresse `http://kaiko.getalp.org/about-DBnary`. Au moment de l'écriture, DBnary contiens plus de 35 millions de triples. Ce nombre à constamment augmenté au long de l'évolution du jeu de données au rythme des évolutions des données originales de Wiktionary. En effet, DBnary est automatiquement mis-à-jour dès que Wikimedia met à disposition une nouvelle version des "vidanges" Wiktionary, c'est-à-dire à peu près tous les 10 à 15 jours.

DBNary est structuré en suivant le modèle de l'Ontologie LEMON pour la représentation de données lexicales liées (McCrae *et al.*, 2011). Le tableau 1 donne une idée du nombre d'éléments lexicaux telles que définies dans l'ontologie LEMON dans les différents éditions de langues.

Les éléments dans DBnary qui n'ont pu être représentés en LEMON, on été définis dans une ontologie sur mesure construite sur la base des classes et relations LEMON, notablement les relations lexico-semantiques ainsi que ce que nous appelons des `Vocables`, les entrées de haut-niveau dans Wiktionary qui correspondent aux pages Wiktionary pour des mots spécifiques et qui contiennent plusieurs entrées lexicales (`LexicalEntry`) catégorisées en deux niveaux :

1. Distinction de mots homonymes selon l'origine étymologique (par exemple : `mode [la mode actuelle]`) contre `mode [le mode de fonctionnement]`.

2. Pour chaque origine étymologique différente, distinction selon la catégorie grammaticale (par exemple `rouge#Adj [la voiture rouge]` contre `rouge#Nom [le rouge au front]`)

Langue	Entrées	LexicalSense	Traductions	Gloses	Texte	Nbr. de sens	Texte + Nbr. de sens
Anglais	544, 338	438, 669	1, 317, 545	1, 288, 667	1, 288, 667	515	515
Bulgare	13888	10104	10104	0	0		
Espagnol	114951	66931	2786	64145	0		
Finnois	49, 620	58, 172	121, 278	120, 728	120, 329	115, 949	115, 550
Français	291, 365	379, 224	504, 061	136, 319	135, 612	28, 821	28, 114
Allemand	205, 977	100, 433	388, 630	388, 553	3, 101	385, 452	0
Grec Moderne	242, 349	108, 283	56, 638	8, 368	8, 368	12	12
Italien	33, 705	47, 102	62, 546	0	0	0	0
Japonais	24, 804	28, 763	85, 606	22, 322	20, 686	4, 148	2, 512
Portugais	45, 109	81, 023	267, 048	74, 901	72, 339	71, 734	69, 172
Russe	129, 555	106, 374	360, 016	151, 100	150, 985	115	0
Turque	64, 678	91, 071	66, 290	53, 348	585	52, 901	138

TABLE 1 – Nombre d'éléments dans le jeu de données DBnary actuel avec des détails sur le nombre d'entrées et de sens de mot ainsi que le nombre de traduction. Le tableau détaille également le nombre de gloses rattachées à des traduction, et de manière plus précise le nombre de gloses textuelles, le nombre de gloses qui contiennent un numéro de sens et enfin le nombre de gloses qui contiennent à la fois une description textuelle et un numéro de sens.

3.1 Relations de traduction

DBnary utilise une représentation ad-hoc pour les relations de traduction, sachant que le modèle LEMON ne propose pas de vocabulaire pour représenter une telle information. `Translation` est une ressource RDF qui rassemble toute l'information se rapportant aux relations de traduction. Par exemple, l'une des traductions de l'entrée lexicale de *frog* est représentée comme suit : [1] :

```
eng:__tr_fra_1_frog__Noun__1
      a       dbnary:Translation ;
      dbnary:gloss "amphibian"@en ;
      dbnary:isTranslationOf
              eng:frog__Noun__1 ;
      dbnary:targetLanguage
              lexvo:fra ;
      dbnary:usage "f" ;
      dbnary:writtenForm "grenouille"@fr .
```

Les propriétés de cette ressource pointent vers une source de type `LexicalEntry`, la langue de la cible (représentée comme une entrée `lexvo.org` (de Melo & Weikum, 2008)), la forme de surface de la cible, et 'éventuellement une glose et des notes d'usage. Les notes d'usage donnent des informations sur la cible de la traduction (habituellement le genre où la transcription de la cible).

The gloss gives disambiguation information about the source of the translation. In the example given, it states that the given translation is valid for the word sense of *frog* that may be described by the hint "*amphibian*". Some of these glosses are textual and summarize or reprise the definition or part thereof of one or more specific sense to which the translation specifically applies to.

As an example, the English `LexicalEntry` *frog* contains 8 word senses, defined as follows :

1. A small tailless amphibian of the order Anura that typically hops
2. The part of a violin bow (or that of other similar string instruments such as the viola, cello and contrabass) located at the end held by the player, to which the horsehair is attached
3. (Cockney rhyming slang) Road. Shorter, more common form of frog and toad
4. The depression in the upper face of a pressed or handmade clay brick
5. An organ on the bottom of a horse's hoof that assists in the circulation of blood
6. The part of a railway switch or turnout where the running-rails cross (from the resemblance to the frog in a horse's hoof)
7. An oblong cloak button, covered with netted thread, and fastening into a loop instead of a button hole.

1. Dans cet article ainsi que pour les jeux de données DBNary nous utilisons la syntaxe RDF Turtle.

8. The loop of the scabbard of a bayonet or sword.

Translations of this entry are devided in 4 groups corresponding to the glosses "*amphibian*", "*end of a string instrument's bow*", "*organ in a horse's foot*" and "*part of a railway*".

Additionally among the glosses, some may contain sense numbers, indicated by users in an ad-hoc way (may or may not be present, if the latter is true, no standard format is systematically followed or enforced). However, the presence of disambiguation information is very irregular and varies greatly between languages, both in terms of wiki structure and representation.

In the current state of the Wiktionary extraction process, we extract translation and when possible the associated glosses. However up to now, we did not exploit the information contained in the glosses to enrich and disambiguate the source senses of translation relations.

As mentioned above, the information contained in translation glosses and their format is very variable between languages, both quantitatively and qualitatively.

Indeed, as shown in Table 1 some language like Italian, contain no gloss altogether, others, like English attaches textual glosses to translations almost systematically, but with no sense numbers. Others still, like German hardly contain textual glosses but give sense numbers to translations. In other cases such as for Finnish, French and Portuguese, many translations have an attached (textual) gloss with associated sense numbers.

In order to evaluate our method, we decided to use these mixed glosses that both contain a textual hint and a sense number to create a endogenous gold standard.

3.1.1 Creation of a gold standard

It is often the case that among translation glosses that are available and that do contain textual information or sense numbers, there are many false positives and variability that result from the variety of structures employed in Wiktionary as well as artefacts resulting from the extraction process. Before we can proceed further, it is paramount to filter this information so as to keep only the relevant parts.

More concretely two steps must be followed if we are to successfully extract the information we need :
– Remove empty glosses, or glosses containing irrelevant textual content that often correspond to TO DO notes in various forms (e.g. *translations to be checked*)
– Extract sense numbers from the glosses when available, using language dependent templates (e.g. "*textual gloss (1)*" or "*1. textual gloss*")

When sufficient glosses contained both a textual hint and sense numbers, we removed the sense numbers [2] from the gloss and used them to create a gold standard in trec_eval format.

Now that we have successfully extracted as much of the information contained in translation glosses, we can move on to the disambiguation process itself. While, the steps above are indeed language specific, what follows was designed to be as generic and computationally efficient as possible, as we are required to periodically perform the disambiguation, whenever a new version of DBnary is extracted from the latest Wiktionary dumps.

4 Attaching Translations to Word Senses

4.1 Formalization of translation disambiguation

Let T be the set of all translation relations, L the set of all `LexicalEntry` in a given language edition of DBnary. Let $T_i \in T : Gloss(T_i)$ be a function that return the gloss of any translation $T_i \in T$ and let $Source(T_i) = L_{T_i}$ be a function that returns a reference to the source `LexicalEntry`, L_{T_i} of a translation T_i. Let $Senses(L_i) = S_{L_i}$ be the set of all the senses associated with `LexicalEntry` L_i. Let $S^k_{L_i}$ be the k-th sense contained in S_{L_i} and let $Def(S^k_{L_i})$ be a function that returns the textual definition of a sense $S^k_{L_i}$. Finally let $Sim(A, B)$ be a function that returns a semantic similarity or relatedness score between A and B, where A, B are a pair of textual definitions or textual glosses.

2. Translation are sometimes valid for several source word senses

Then, we can express the disambiguation process as :

$$\forall T_i \in T, S = Senses(Source(T_i)):$$

$$Source^*(T_i) \leftarrow \underset{S^k \in S}{\operatorname{argmax}}\{Score(Gloss(T_i), Def(S^k))\}$$

This corresponds exactly to a standard semantic similarity maximisation and yields one disambiguated source sense per translation, however in many case a translation corresponds to one or more senses. The solution adopted by (Meyer & Gurevych, 2012a) is to use a threshold k for their gloss overlap, however in our case, we want to be able to plug-in several different measures so as to find the most suitable one, as such just choosing a fixed arbitrary value for k is not readily and option. Thus, we need to add one more constraint : that the values returned by our similarity function need to be normalized between 0 and 1.

Here instead of taking a threshold k, we set a window δ around the best score in which the senses are accepted as a disambiguation of a given translation. We hypothesise that a relative threshold dependant on the maximal score, will set a precedent and be more representative of the possible range of values. Of course, setting a fixed threshold has for effect of not assigning any senses if all the scores are low, thus increasing precision at the cost of a lower recall. While in a general setting it is better to remove answers that are more likely to be mistakes as detecting errors *a posteriori* is difficult. However in the context of the experiment, we prefer to keep such low or null scores as we will then be able with the help of the gold standard pin-point errors more precisely for the sake of our analysis.

We can express this formally by modifying the $argmax$ function as such :

$$\forall T_i \in T, S = Senses(Source(T_i)):$$

$$M_S = \max_{S_k \in S}(Score((Gloss(T_i), Def(S^k))),$$

$$\underset{S_i \in S}{\overset{\delta}{\operatorname{argmax}}}\{Score(Gloss(T_i), Def(S^k))\} =$$

$$\{S^k \in S | M_S > Score((Gloss(T_i), Def(S^k)) > M_S - \delta\}$$

4.2 Similarity Measure

In order to disambiguate the translation, we need to be able to compute some form of semantic similarity measure. Given that the only information available in the translations is the gloss that summarises the definition of the corresponding sense, we need a measure to capture the similarity by comparing the translation glosses and the sense definitions. The Lesk (Lesk, 1986) measure is a standard semantic similarity measure, specifically suited for such tasks as it computes a similarity based on the number of exact overlapping words between definitions. The Lesk similarity however, has several important issues that need to be addressed when its use is mandated :

– If the sizes of the glosses or definitions is not the same, the Lesk measure will always favor longer definitions.
– The size and the appropriateness of the words contained in the definitions is important, as one key word to the meaning of the definition is missing (or the presence of a synonym for that matter) can lead to an incorrectly low similarity.
– The Lesk overlap is not in itself normalized, and the normalization process requires some though depending of the distinct problems at hand.

The issues of normalization and of the unequal length of definitions are actually related, as one of the practical ways of doing so is to, for example, divide by the length of the shortest definition as a normalization. Moreover, one can only notice the similarity between the Lesk measure and other overlap coefficients such as the dice coefficient, the Jaccard or Tatimono indices. In fact, all of these measures are special forms of the Tversky index, which stems from the works on similarity of cognitive psychologist A. Tversky (Tversky, 1977).

The Tversky index can be defined as follows. Let $s_1 \in Senses(L_1)$ and $s_2 \in Senses(L_2)$ be the senses of two lexical entries L_1 and L_2. Let $d_i = Def(s_i)$ be the definition of s_i, represented as a set of words. The similarity between the senses $Score(s_1, s_2)$ can be expressed as

$$Score(s_1, s_2) = \frac{|d_1 \cap d_2|}{|d_1 \cap d_2| + \alpha|d_1 - d_2| + \beta|d_2 - d_1|}$$

The measure can further be generalized following (Pirrò & Euzenat, 2010) by replacing the cardinality function by any function F. Depending on the values of α and β, the Tversky index takes the particular form of other similar indexes. For $(\alpha = \beta = 0.5)$ for example it is equivalent to the dice coefficient, and for $(\alpha = \beta = 1)$ to the Tatimono index. More generally, the values of α and β express how much emphasis one wants to attribute to the commonality or differences of one or the other set.

The Tversky index in itself is not a metric in the mathematical sense, as it is neither symmetric nor respects the triangular inequality, however, a symmetric variant has been proposed by (Jimenez *et al.*, 2010) for such cases where the symmetry property is important or required.

When using the Tversky index or its symmetric version, careful attention must be given to the weights that one might use depending on the application.

In this paper, there is no indication that there is any need for a symmetric version and preliminary experiments have shown better performance with the non-symmetric version.

4.2.1 Multilingual Setting & Partial overlaps

When working on a single language such as English and French, we have at our disposal tools such as a lemmatizer or a stemmer that may help to retrieve a canonical representation of the terms. Thus, we can hope to maximize the overlap and reduce the usual sparsity of glosses or sense definitions. For agglutinative languages like German or Finnish, highly inflective language (for example in the Bangla language, common stems are often composed of a single character, which makes stemming difficult to exploit) or languages with no clear segmentation, the preprocessing steps are paramount in order to make overlap based measures viable. If one is working on a single language, even if stemmers and lemmatizers do not exist, it is not impossible to build such a tool.

However, in the context of this work we are currently dealing with 10 languages (and potentially in the future with all the languages present in Wiktionary) and thus, in order to propose a truly general method, we cannot expect the prerequisite presence of such tools.

How then, can we manage to compute overlaps effectively ? When computing Lesk, if two words overlap, the score is increased by 1 and of two words do not overlap, the overlap does not change. What if we had a way to count meaningful partial overlaps between words and instead of adding 1, we may add whatever values between 0 and 1 represents the amount of overlap.

The simplest approach to the problem is to use some form of partial string matching metric to compute partial overlaps, a seemingly trivial and classical approach that can, however, greatly improve the result as we shall endeavour to elicit and as has already been extensively shown by (Jimenez *et al.*, 2012).

As mentioned in the Related Work section, there are many approximate string matching measures as reviewed by (Cohen *et al.*, 2003). We integrate these measures in the Teversky index by setting the F function that replaces the set cardinality function appropriately (a simplified version of soft cardinality) :

$$A \text{ , a set : } F(A) = (\sum_{A_i, A_j \in A} sim(A_i, A_j))^{-1}$$

In our case, sim will be an string distance measure.

4.2.2 Longest Common Substring Constraints

With this similarity measure, we are mainly interested in capturing word that have common stems, without the need for a stemmer, as such we do not for example want to consider the overlap of prefixes or suffices, as they do not carry the main semantic information of the word. If two words only match by a common suffix that happens to be used very often in that particular language, we will have a non-zero overlap, but we would have captures no sematic information whatsoever. Thus, in this word we put a threshold of a longest common subsequence of three characters.

5 Experiments

As was mentioned previously, we have been able to extract gold standards from the sense numbered textual glosses of translations in certain language editions of Wiktionary. Then we stripped all sense number information from the glosses, so we could disambiguate those same translation and then evaluate the results on the previously generated gold standard.

We will first describe how we generated the Gold standard and the tools and measures used for the evaluation. We will then proceed onto the empirical selection of the best parameters for our Tversky index as well as the most appropriate string distance measure to use for the fuzzy or soft cardinality. Then, we will compare the results of the optimal Tversky index with other Level 2 similarity measures.

5.1 Evaluation

Let us first describe the Gold Standard generation process, then proceed on to describing how we represented the Gold Standard in Trec_eval format, a scorer program from the query answering Trec_Eval campaign. Finally we will described the evaluation measures we will use.

5.2 Gold Standard

Only certain languages meet the requirements for the generation of a sufficiently large Gold Standard. To be more specific, we could only chose among languages where :

1. There are textual glosses (for the overlap measures)
2. There are numbers in said glosses indicating the right sense number
3. The above are available in a sufficient quantity (at least a few thousand)

Four languages could potentially meet the criteria (see the last column of Table 1) : French, Portuguese, Finnish and Japanese, however we could only manage the time to extract gold standards for French, Portuguese and Finnish.

5.2.1 Trec_eval, scoring as a query answering task

A query answering task is more generally a multiple-labelling problem, which is exactly equivalent to what we are producing when we use the threshold δ. Here, we can consider that each translation number is the query identifier and that each sense URI is a document identifier. We answer the "translation" queries by providing one or more senses and an associated weight.

Thus, we can generate the gold standard and the results in the Trec_eval format, the very complete scorer for and information retrieval evaluation campaign of the same name.

5.2.2 Measures

We will use the standard set matching metrics used in Information Retrival and Word Sense Disambiguation, namely Recall, Precision and F_1 measure. Where, $P = \frac{|\{Relevant\} \cap \{Disambiguated\}|}{|\{Disambiguated\}|}$, $R = \frac{|\{Relevant\} \cap \{Disambiguated\}|}{|\{Relevant\}|}$, and $F_1 = \frac{2 \cdot P \cdot R}{P+R}$, the harmonic mean of R and P. However, for the first step consisting in the estimation of the optimal parameters, we will only provide the $F_1 score$, as we are interested in maximising both recall and precision in an equal fashion.

5.3 Similarity Measure Tuning

There are several parameters to set in our Tversky index, however the first step is to find the most suitable string distance measure.

	French	Portuguese	Finnish
	F1	F1	F1
FTiJW	0.7853	0.8079	0.9479
FTiLcss	0.7778	0.7697	0.9495
FTiLs	**0.7861**	**0.8176**	**0.9536**
FTiME	0.7684	0.7683	0.9495
Ti	0.7088	0.7171	0.8806

TABLE 2 – Results comparing the performance in terms of F_1 score for French, Finnish and Portuguese (highest scores in bold).

5.3.1 Optimal String Distance Metric

The δ parameter influences performance independently of the similarity measure, so we can first operate with $\delta = 0$, which restricts us to a single disambiguation per translation. Furthermore, the weights of the Tvsersky index are applied downstream from the string edit distance, and thus does not influence the relative performance of the different string distance metrics combined to our Tversky index. In simple terms, the ratio of the tverski indices computed on different measures is constant, independently of α and β. Thus for this first experiment, we will set $\alpha = \beta = 0.5$, in other words the index becomes the dice coefficient.

As for the selection of the string similarity measures to compare, we take the best performing measures from (Cohen *et al.*, 2003), namely Jaro-Winkler, Monge-Elkan, Scaled Levenshtein Distance, to which we also add the longest common substring for reference. As a baseline measure, we will use the Tversky index with a standard overlap cardinality.

We give the following short notations for the measures : Tversky Index – Ts ; Jaro-Winkler – JW ; Monge-Elkan – ME ; Scaled Levenshtein – Ls ; Longest Common Substring – Lcss ; F – Fuzzy. For example standard Tversky index with classical cardinality shall be referred to as "Ti", while the fuzzy cardinality version with a Monge-Elkan string distance shall be referred to as "FTiME".

Table 2 presents the results for each string similarity measure and each of the languages (Fr, Fi, Pt).

As we can see, for all language, the best string similarity measure is clearly the scaled Levenstein measure as it systematically exhibits a score higher from $+1\%$ to $+1.96\%$.

5.3.2 Optimal α, β selection

Now that we have found the optimal string distance measure, we can look for the optimal ratio of α and β. Here, we will keep both values complementary, that is $\alpha = 1 - \beta$ so as to obtain more balances score that remain in the 0 to 1 range.

Given that translation glosses are short (often a single word), it is likely that the optimum is around $\alpha = 1 - \beta = 0.1$, given that what interests us is that the word in the translation gloss matches with less importance for the remaining words in the sense definition that do not match.

We chose, here, to evaluate the values of α and β in steps of 0.1. Figure 1 graphically shows the F_1 score for each pair of values of $alpha$ and $beta$ for all three languages. We can indeed confirm our hypothesis as the optimal value in all three cases is indeed $\alpha = 1 - \beta = 0.1$ with a difference between $+0.15\%$ to $+0.43\%$ with the second best scores.

5.3.3 Optimal δ selection

Now that we have fixed the best values of α and β we can search for the best value for δ. We make delta vary in steps of 0.05 between 0 and 0.3. The choice of the upper bound is based on the hypothesis that the optimal value is somewhere closer to 0, as a too large threshold essentially means that most of the senses for each translation might be considered as corresponding to the translation at hand and thus it should drastically reduce performance.

The δ heuristic affects the results of the disambiguation whether the measure is Tversky index or another Level 2 Textual similarity. Thus, in this experiment, we will also include Level 2 version of the three string distance measures that we used in the first experiment.

Figure 2 graphically presents the F_1 scores for each value of $delta$ and each language. The first apparent trend is that

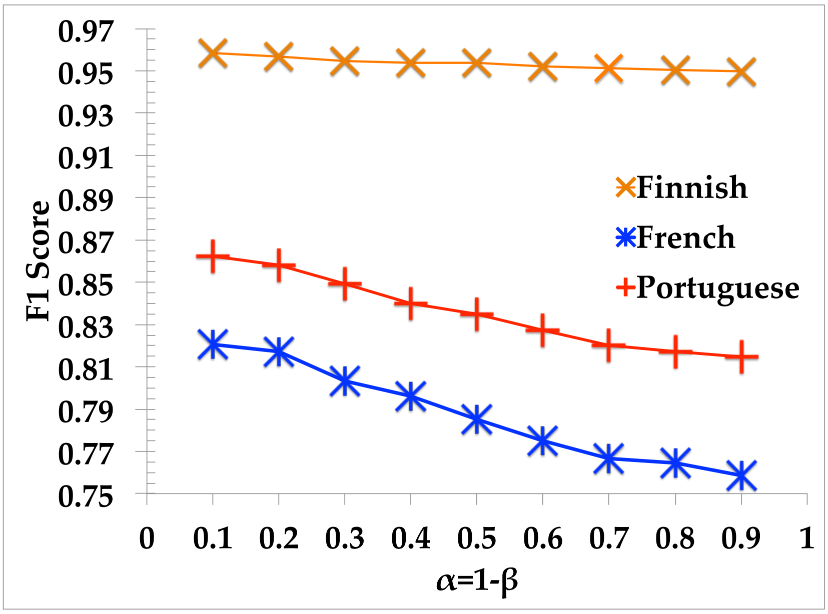

FIGURE 1 – F1 score for Finnish, French and Portuguese depending on the value of α and β.

	P	R	F1	MFS F1	Random
Portuguese	0.8572	0.8814	0.8651	0.2397	0.3103
Finnish	0.9642	0.9777	0.9687	0.7218	0.7962
French	0.8267	0.8313	0.8263	0.3542	0.3767

TABLE 3 – Final results with optimal measure and parameter values. Precision, Recall, F1 measure for all three languages compared against the MFS and Random Baselines.

Level 2 measures are systemically performing much worse (by up to 30%) than our own similarity measure. Depending on the language different values of $delta$ are optimal, even though it is difficult to see a great difference. For French $\delta = 0.10$, for Finnish $\delta = 0.15$ and for Portuguese $\delta = 0.10$. In all three previous experiments, it became apparent, that the same string similarity measure, the same values for alpha and beta as well as the same value for delta were optimal, which leads us to believe that their optimality will be conserved across all languages. However, especially for the string similarity measure, it is reasonable to believe that for languages such a Chinese or Japanese that lack segmentation, the optimal choice for the string distance measure may be entirely different.

5.4 Final Disambigation Results

Now that we have found all the optimal parameters, we can actually present the final results combining all the optimal parameters. They are in fact the very results that were optimal in the previous experiment. We shall now take these results and place them in a separate table (Table 3) so as to make the comparison analysis easier. We will use the chance of random selection as well as the most frequent sense selection as baseline for this comparison.

The first thing one can notice is that there is a stark difference between the scores of Finnish, and the rest. Indeed, first of all the random baseline and most frequent sense baselines are an indication that the French and Portuguese DBNaries are highly polysemous, while Finnish contains a very large amount of monosemous entries, which artificially inflates the value of the score.

Another point of interest is that the random baseline is higher (up to 6.6%) than the most frequent sense baseline, which indicates that the first sense if often not the right sense to select to match the translation.

We can see that for all three languages we achieve a good performance compared to what is presented in the literature, most notably in the fact that most of the errors, can easily be identified as such just by looking at whether or not it produced any overlap.

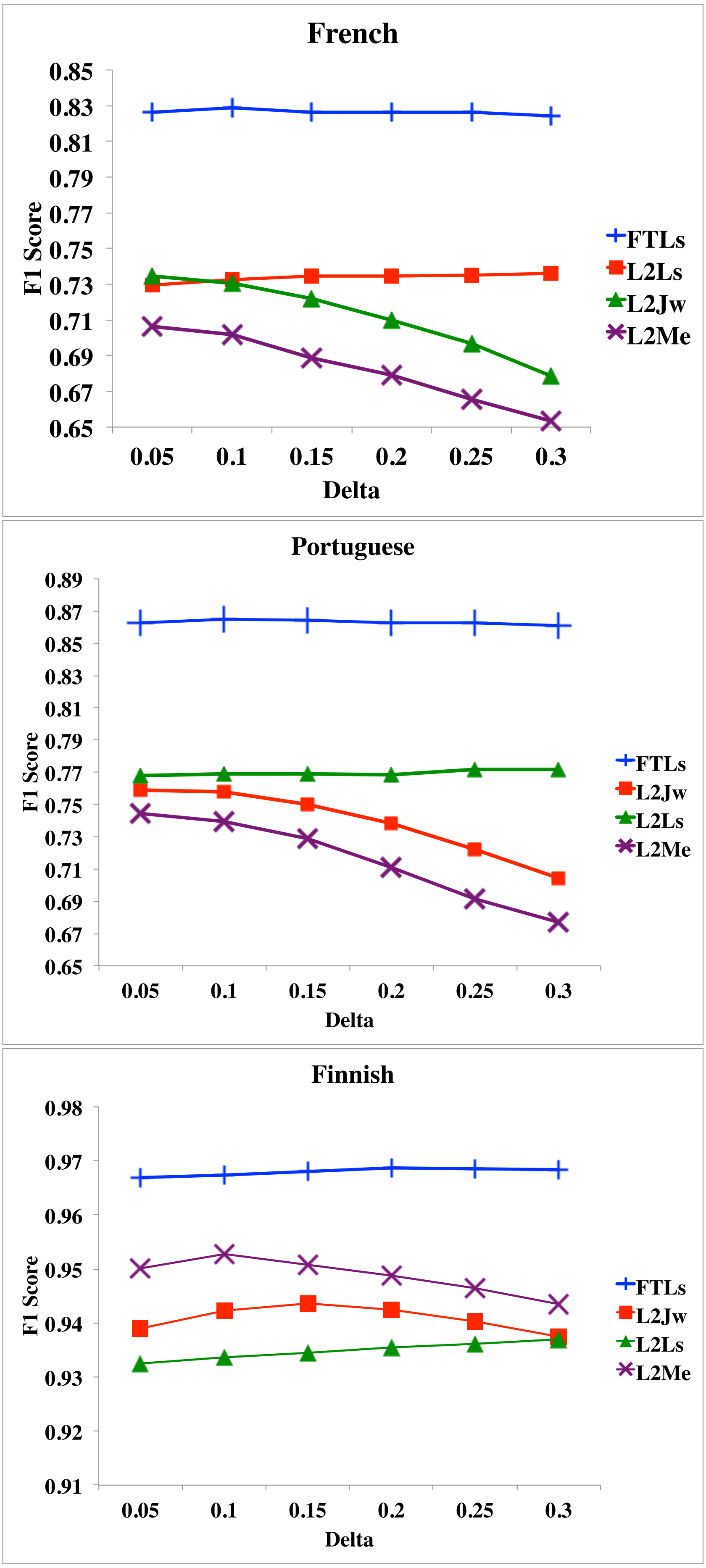

FIGURE 2 – Graphical representation of the F1 score against delta for our measure and other Level 2 Measures.

5.5 Error analysis

We did not here perform a full fledged and systematic error analysis, but rather an informal manual sampling so as to have an idea of what the error can be and if there are ways to correct them by adapting the measures or the methodology. We looked at some of the error made by the disambiguation process and manually checked them so as to categorize them. We found three main categories :

1. No overlap between the gloss and sense definitions (Random choice by our algorithm), this happens when the translation gloss is a paraphrase of the sense definition or simply a metaphor for it.
2. The overlap is with the domain category label or the example glosses, which we do not currently extract. This is a particular case of the first type of error.
3. New senses have been introduced in Wiktionary and shifted sense numbers, which were not subsequently updated in the resource. Such errors cannot be detected during the extraction process.

We can in fact easily find all the errors due to the lack of overlap and correct the errors of type 2 by enriching the extraction process of DBnary. Thus we can single out errors that are due to inconsistencies in the resource and thus potentially use the disambiguation results to indicate to users where errors are located an need to be updated.

6 Conclusion

With our method, we were able to determine and optimal similarity measure for disambiguating translation in DBnary. Similar results across the three evaluation languages suggests that it is a general optimality that can be applied to all the languages currently present in DBnary, although for Asian Languages that have no segmentation, it is likely not the case.

Then, we compared the results and concluded that our method is viable for the task of disambiguating glossed translation relations, especially considering the low random baselines and first sense baselines compared to the top score of our disambiguation method.

For translation relations without glosses, the disambiguation process is more complex and is part of the Future Work that we plan on carrying out.

Remerciements (pas de numéro)

Paragraphe facultatif

Références

COHEN W. W., RAVIKUMAR P. & FIENBERG S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, p. 73–78.

DE MELO G. & WEIKUM G. (2008). Language as a Foundation of the {Semantic Web}. In C. BIZER & A. JOSHI, Eds., *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC)*, volume 401 of *CEUR WS*, Karlsruhe, Germany : CEUR.

GUREVYCH I., ECKLE-KOHLER J., HARTMANN S., MATUSCHEK M., MEYER C. M. & WIRTH C. (2012). Uby : A large-scale unified lexical-semantic resource based on LMF. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 580–590 : Association for Computational Linguistics.

HELLMANN S., BREKLE J. & AUER S. (2013). Leveraging the Crowdsourcing of Lexical Resources for Bootstrapping a Linguistic Data Cloud. *Semantic Technology*, p. 191—-206.

JIMENEZ S., BECERRA C. & GELBUKH A. (2012). Soft Cardinality : A Parameterized Similarity Function for Text Comparison. In **SEM 2012 : The First Joint Conference on Lexical and Computational Semantics – Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*.

JIMENEZ S., GONZALEZ F. & GELBUKH A. (2010). Text comparison using soft cardinality. In *Proceedings of the 17th International Conference on String Processing and Information Retrieval*, p. 297–302, Los Cabos, Mexico : Springer-Verlag.

KRIZHANOVSKY A. A. (2010). Transformation of Wiktionary entry structure into tables and relations in a relational database schema. *arXiv preprint arXiv :1011.1368*.

LEHMANN J., ISELE R., JAKOB M., JENTZSCH A., KONTOKOSTAS D., MENDES P. N., HELLMANN S., MORSEY M., VAN KLEEF P., AUER S. & BIZER C. (2014). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*.

LESK M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries : How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, p. 24–26, New York, NY, USA : ACM.

MCCRAE J., SPOHR D. & CIMIANO P. (2011). Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In G. ANTONIOU, M. GROBELNIK, E. P. B. SIMPERL, B. PARSIA, D. PLEXOUSAKIS, P. D. LEENHEER & J. Z. PAN, Eds., *ESWC (1)*, volume 6643 of *Lecture Notes in Computer Science*, p. 245–259 : Springer.

MEYER C. M. & GUREVYCH I. (2010). Worth its weight in gold or yet another resource – a comparative study of wiktionary, openthesaurus and germanet. In A. GELBUKH, Ed., *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics*, volume 6008 of *Lecture Notes in Computer Science*, p. 38–49. Berlin/Heidelberg : Springer.

MEYER C. M. & GUREVYCH I. (2012a). *Electronic Lexicography*, chapter Wiktionary : A new rival for expert-built lexicons ? Exploring the possibilities of collaborative lexicography, p. (to appear). Oxford University Press.

MEYER C. M. & GUREVYCH I. (2012b). To Exhibit is not to Loiter : A Multilingual, Sense-Disambiguated Wiktionary for Measuring Verb Similarity. In *Proceedings of COLING 2012*, p. 1763–1780, Mumbai, India : The COLING 2012 Organizing Committee.

NAVARRO E., SAJOUS F., GAUME B., PRÉVOT L., HSIEH S., KUO T. Y., MAGISTRY P. & HUANG C. R. (2009). Wiktionary and NLP : Improving synonymy networks. In I. GUREVYCH & T. ZESCH, Eds., *Proceedings of the 2009 Workshop on The People's Web Meets NLP : Collaboratively Constructed Semantic Resources (People's Web)*, p. 19–27, Suntec, Singapore : Association for Computational Linguistics.

PIRRÒ G. & EUZENAT J. (2010). A Semantic Similarity Framework Exploiting Multiple Parts-of Speech. In R. MEERSMAN, T. S. DILLON & P. HERRERO, Eds., *OTM Conferences (2)*, volume 6427 of *Lecture Notes in Computer Science*, p. 1118–1125 : Springer.

SÉRASSET G. (2012). Dbnary : Wiktionary as a Lemon Based RDF Multilingual Lexical Resource. *Semantic Web Journal - Special issue on Multilingual Linked Open Data*.

TVERSKY A. (1977). Features of Similarity. *Psychological Review*, **84**(2), 327–352.

ZESCH T., MÜLLER C. & GUREVYCH I. (2008a). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

ZESCH T., MÜLLER C. & GUREVYCH I. (2008b). Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI*, Chicago, Illinois, USA.