# Obtaining SMT dictionaries for related languages

**Miguel Rios, Serge Sharoff**
Centre for Translation Studies
University of Leeds
m.riosgaona,s.sharoff@leeds.ac.uk

## Abstract

This study explores methods for developing Machine Translation dictionaries on the basis of word frequency lists coming from comparable corpora. We investigate (1) various methods to measure the similarity of cognates between related languages, (2) detection and removal of noisy cognate translations using SVM ranking. We show preliminary results on several Romance and Slavonic languages.

## 1 Introduction

Cognates are words having similarities in their spelling and meaning in two languages, either because the two languages are typologically related, e.g., *maladie* vs *malattia* ('disease'), or because they were both borrowed from the same source (*informatique* vs *informatica*). The advantage of their use in Statistical Machine Translation (SMT) is that the procedure can be based on comparable corpora, i.e., similar corpora which are not translations of each other (Sharoff et al., 2013). Given that there are more sources of comparable corpora in comparison to parallel ones, the lexicon obtained from them is likely to be richer and more variable.

Detection of cognates is a well-known task, which has been explored for a range of languages using different methods. The two main approaches applied to detection of the cognates are the generative and discriminative paradigms. The first one is based on detection of the edit distance between potential candidate pairs. The distance can be a simple Levenshtein distance, or a distance measure with the scores learned from an existing parallel set (Tiedemann, 1999; Mann and Yarowsky, 2001). The discriminative paradigm uses standard approaches to machine learning, which are based on (1) extracting features, e.g., character n-grams, and (2) learning to predict the transformations of the source word needed to (Jiampojamarn et al., 2010; Frunza and Inkpen, 2009). Given that SMT is usually based on a full-form lexicon, one of the possible issues in generation of cognates concerns the similarity of words in their root form vs the similarity in endings. For example, the Ukrainian wordform ближнього 'near$_{gen}$' is cognate to Russian ближнего, the root is identical, while the ending is considerably different (ього vs его). Regular differences in the endings, which are shared across a large number of words, can be learned separately from the regular differences in the roots.

One also needs to take into account the false friends among cognates. For example, *diseñar* means 'to design' in Spanish vs *desenhar* in Portuguese means 'to draw'. There are also often cases of partial cognates, when the words share the meaning in some contexts, but not in others, e.g., жена in Russian means 'wife', while its Bulgarian cognate жена has two meanings: 'wife' and 'woman'. Yet another complexity concerns a frequency mismatch. Two cognates might differ in their frequency. For example, *dibujo* in Spanish ('a drawing', rank 1779 in the Wikipedia frequency list) corresponds to a relatively rare cognate word *debuxo* in Portuguese (rank 104,514 in Wikipedia), while another Portuguese word *desenho* is more commonly used in this sense (rank 884 in the Portuguese Wikipedia). For MT tasks we need translations that are equally appropriate in the source and target language, therefore cognates useful for a high-quality dictionary for SMT need to have roughly the same frequency in comparable corpora and they need to be used in similar contexts.

This study investigates the settings for extracting cognates for related languages in Romance and Slavonic language families for the task of reducing the number of unknown words for SMT. This in-



*Proceedings of the Eighth Workshop on Building and Using Comparable Corpora*, pages 68–73,
Beijing, China, July 30, 2015. 

cludes the effects of having constraints for the cognates to be similar in their roots and in the endings, to occur in distributionally similar contexts and to have similar frequency.

## 2 Methodology

The methodology for producing the list of cognates is based on the following steps: 1) Produce several lists of cognates using a family of distance measures, discussed in Section 2.1 from comparable corpora, 2) Prune the candidate lists by ranking items, this is done using a Machine Learning (ML) algorithm trained over parallel corpora for detecting the outliers, discussed in Section 2.2;

The initial frequency lists for alignment are based Wikipedia dumps for the following languages: **Romance** (French, Italian, Spanish, Portuguese) and **Slavonic** (Bulgarian, Russian, Ukrainian), where the target languages are Spanish and Russian[1].

### 2.1 Cognate detection

We extract possible lists of cognates from comparable corpora by using a family of similarity measures:

**L** direct matching between the languages using Levenshtein distance (Levenshtein, 1966); $L(w_s, w_t) = 1 - ed(w_s, w_t)$

**L-R** Levenshtein distance with weights computed separately for the roots and for the endings; $LR(r_s, r_t, e_s, e_t) = \frac{\alpha \times ed(r_s, r_t) + \beta \times ed(e_s, e_t)}{\alpha + \beta}$

**L-C** Levenshtein distance over word with similar number of starting characters (i.e. prefix);

$$LC(c_s, c_t) = \begin{cases} 1 - ed(c_s, c_t), & \text{same prefix} \\ 0, & \text{otherwise} \end{cases}$$

where $ed(.,.)$ is the normalised Levenshtein distance in characters between the source word $w_s$ and the target word $w_t$. The $r_s$ and $r_t$ are the stems produced by the Snowball stemmer[2] . Since the Snowball stemmer does not support Ukrainian and Bulgarian, we used the Russian model for making the stem/ending split. $e_s, e_t$ are the characters at the end of a word form given a stem and $c_s, c_t$ are the first $n$ characters of a word. In this work, we set the weights $\alpha = 0.6$ and $\beta = 0.4$ giving more importance to the roots. We set a higher weight to roots on the **L-R**, which is language dependent, and compare to the **L-C** metric, which is language independent. We transform the Levenshtein distances into similarity metrics by subtracting the normalised distance score from one.

The produced lists contain for each source word the possible n-best target words accordingly to the maximum scores with one of the previous measures. The n-best list allows possible cognate translations to a given source word that share a part of the surface form. Different from (Mann and Yarowsky, 2001), we produce n-best cognate lists scored by edit distance instead of 1-best. An important problem when comparing comparable corpora is the way of representing the search space, where an exhaustive method compares all the combinations of source and target words (Mann and Yarowsky, 2001). We constraint the search space by comparing each source word against the target words that belong to a frequency window around the frequency of the source word. This constraint only applies for the **L** and **L-R** metrics. We use Wikipedia dumps for the source and target side processed in the form frequency lists. We order the target side list into bins of similar frequency and for the source side we filter words that appear only once. We use the window approach given that the frequency between the corpora under study can not be directly comparable. During testing we use a wide window of $\pm 200$ bins to minimise the loss of good candidate translations. The second search space constraint heuristic is the **L-C** metric. This metric only compares source words with the target words upto a given $n$ prefix. For $c_s, c_t$ in **L-C** , we use the first four characters to compare groups of words as suggested in (Kondrak et al., 2003).

### 2.2 Cognate Ranking

Given that the n-best lists contain noise, we aim to prune them by an ML ranking model. However, there is a lack of resources to train a classification model for cognates (i.e. cognate vs. false friend), as mentioned in (Fišer and Ljubešić, 2013). Available data that can be used to judge the cognate lists are the alignment pairs extracted from parallel data. We decide to use a ranking model to avoid data imbalance present in classification and to use the probability scores of the alignment pairs as

[1] For the Slavonic family we only use languages based on the Cyrillic alphabet to avoid the character set problems.

[2] http://snowball.tartarus.org/

ranks, as opposed to the classification model used by (Irvine and Callison-Burch, 2013). Moreover, we also use a popular domain adaptation technique (Daumé et al., 2010) given that we have access to different domains of parallel training data that might be compatible with our comparable corpora.

The training data are the alignments between pairs of words where we rank them accordingly to their correspondent alignment probability from the output of GIZA++ (Och and Ney, 2003). We then use a heuristic to prune training data in order to simulate cognate words. Pairs of words scored below the Levenshtein similarity threshold of 0.5 are not considered as cognates given that they are likely to have a different surface form.

We represent the training and test data with features extracted from different edit distance scores and distributional measures. The edit distances features are as follows: 1) Similarity measure **L** and 2) Number of times of each edit operation. Thus, the model assigns a different importance to each operation. The distributional feature is based on the cosine between the distributional vectors of a window of n words around the word currently under comparison. We train distributional similarity models with word2vec (Mikolov et al., 2013a) for the source and target side separately. We extract the continuous vector for each word in the window, concatenate it and then compute the cosine between the concatenated vectors of the source and the target. We suspect that the vectors will have similar behaviour between the source and the target given that they are trained under parallel Wikipedia articles. We develop two ML models: 1) Edit distance scores and 2) Edit distance scores and distributional similarity score.

We use SVMlight (Joachims, 1998) for the ranking model with the augmented features for domain adaptation. The domains are as follows: Wikipedia aligned titles, open source subtitles and proprietary subtitles, discussed in Section 3.1.

## 3 Results and Discussion

In this section we describe the data used to produce the n-best lists and train the cognate ranking models. We evaluate the ranking models with heldout data from each training domain. We also provide manual evaluation over the ranked n-best lists for error analysis.

### 3.1 Data

The n-best lists to detect cognates were extracted from the respective Wikipedias by using the method described in Section 2.1. The training data for the ranking model consists of different types of parallel corpora. The parallel corpora are as follows: 1) **Wiki-titles** we use the inter language links to create a parallel corpus from the tittles of the Wikipedia articles, with about 500K aligned links (i.e. 'sentences') per language pair (about 200k for bg-ru), giving us about 200K training instances per language pair [3], 2) **Opensubs** is an open source corpus of subtitles built by the fan community, with 1M sentences, 6M tokens, 100K words, giving about 90K training instances (Tiedemann, 2012) and 3) **Zoo** is a proprietary corpus of subtitles produced by professional translators, with 100K sentences, 700K tokens, 40K words and giving about 20K training instances per language pair.

Our objective is to create MT dictionaries from the produced n-best lists and we use parallel data as a source of training to prune them. We are interested in the corpora of subtitles because the chosen domain of our SMT experiments is subtitling, while the proposed ranking method can be used in other application domains as well.

We consider Zoo and Opensubs as two different domains given that they were built by different types of translators and they differ in size and quality. The heldout data consists of 2K instances for each corpus.

We use Wikipedia documents and Opensusbs subtitles to train word2vec for the distributional similarity feature. We use the continuous bag-of-words algorithm for word2vec and set the parameters for training to 200 dimensions and a window of 8 words. The Wikipedia documents with an average number of 70K documents for each language, and Opensubs subtitles with 1M sentences for each language. In practice we only use the Wikipedia data given that for Opensubs the model is able to find relatively few vectors, for example a vector is found only for 20% of the words in the pt-es pair.

### 3.2 Evaluation of the Ranking Model

We define two ranking models as: model *E* for edit distance features and model *EC* for both edit

[3]The aligned links have been extracted with: https://github.com/clab/wikipedia-parallel-titles

| | Zoo error% | | Opensubs error% | | Wiki-titles error% | |
|---|---|---|---|---|---|---|
| Lang pairs | Model E | Model EC | Model E | Model EC | Model E | Model EC |
| Romance | | | | | | |
| pt-es | 53.31 | 53.72 | 54.81 | 48.31 | 12.22 | 9.87 |
| it-es | 56.00 | 42.86 | 63.95 | 63.03 | 8.44 | 11.23 |
| fr-es | 59.05 | 53.00 | 43.00 | 41.19 | 10.75 | 10.09 |
| Slavonic | | | | | | |
| uk-ru | 47.90 | 40.84 | 37.06 | 30.19 | 10.71 | 10.72 |
| bg-ru | 54.17 | 43.98 | 49.12 | 57.89 | 18.72 | 17.13 |

Table 1: Zero/one-error percentage results on heldout test parallel data for each training domain.

distance and distributional similarity features. We evaluate these models with heldout data from each domain used for training. Each test dataset is evaluated with Zero/one-error percentage, that is the fraction of perfectly correct rankings. We evaluate the models for the Romance and Slavonic families where the target languages are Spanish and Russian respectively.

Table 1 shows the results of the ranking procedure. For the Romance family language pairs the model *EC* with context features consistently reduces the error compared to the solely use of edit distance metrics. The only exception is the it-es *EC* model with poor results for the domain of Wiki-titles. The models for the Slavonic family behave similarly to the Romance family, where the use of context features reduces the ranking error. The exception is the bg-ru model on the Opensubs domain.

A possible reason for the poor results on the it-es and bg-ru models is that the model often assigns a high similarity score to unrelated words. For example, in it-es, *mortes* 'deads' is treated as close to *categoria* 'category'. A possible solution is to map the vectors form the source side into the space of the target side via a learned transformation matrix (Mikolov et al., 2013b).

### 3.3 Preliminary Results on Comparable Corpora

After we extracted the n-best lists for the Romance family comparable corpora, we applied one of the ranking models on these lists and we manually evaluated over a sample of 50 words[4]. We set $n$ to 10 for the n-best lists. We use a frequency window of 200 for the n-best list search heuristic and the domain of the comparable corpora to Wiki-titles for the domain adaptation technique. The purpose of manual evaluation is to decide whether the ML setup is sensible on the objective task. Each list is evaluated by accuracy at 1 and accuracy at 10. We also show success and failure examples of the ranking and the n-best lists. Table 2 shows the preliminary results of the ML model *E* on a sample of Wikipedia dumps. The **L** and **L-R** lists consistently show poor results. A possible reason is the amount of errors given the first step to extract the n-best lists. For example, in pt-es, for the word *vivem* 'live' the 10-best list only contain one word with a similar meaning *viva* 'living' but it can be also translated as 'cheers'.

In the pt-es list for the word *representação* 'description' the correct translation *representación* is not among the 10-best in the **L** list. However, it is present in the 10-best for the **L-C** list and the ML model *EC* ranks it in the first place. The edit distance model *E* still makes mistakes like with the list **L-C**, the word *vivem* 'live' translates into *viven* 'living' and the correct translation is *vivir*. However, given a certain context/sense the previous translation can be correct. The ranking scores given by the SVM varies from each list version. For the **L-C** lists the scores are more uniform in increasing order and with a small variance. The **L** and **L-R** lists show the opposite behaviour.

We add the produced Wikipedia n-best lists with the **L** metric into a SMT training dataset for the pt-es pair. We use the Moses SMT toolkit (Koehn et al., 2007) to test the augmented datasets. We compare the augmented model with a baseline both trained by using the Zoo corpus of subtitles. We use a 1-best list consisting of 100K pairs. Te dataset used for pt-es baseline is: 80K training sentences, 1K sentences for tuning and 2K sen-

[4]The sample consists of words with a frequency between 2K and 5.

| | List L | | List L-R | | List L-C | |
|---|---|---|---|---|---|---|
| Lang Pairs | acc@1 | acc@10 | acc@1 | acc@10 | acc@1 | acc@10 |
| pt-es | 20 | 60 | 22 | 59 | 32 | 70 |
| it-es | 16 | 53 | 18 | 45 | 44 | 66 |
| fr-es | 10 | 48 | 12 | 51 | 29 | 59 |

Table 2: Accuracy at 1 and at 10 results of the ML model E over a sample of 50 words on Wikipedia dumps comparable corpora for the Romance family.

tences for testing. We use fast-align[5], KenLM[6] with a 5-gram language model and Moses with the standard feature set . The BLEU score for the baseline is 20.68 and 20.86 for the augmented version, where the +0.18 increase is not statistically significant. However, the augmented dataset improves the coverage of the model. The out of vocabulary (OOV) words decrease from: 1476 tokens (9.4%), 623 types (21.1%) to 896 tokens (5.7%) and 337 types (11.4%). For uk-ru the baseline training data is: 140K training sentences, 1K sentences for tuning and 2K sentences for testing. The uk-ru 1-best list consists of 100K. The BLEU results for the baseline is 28.72 and 29.56 for the augmented dataset with a difference in +0.93 which is not statistically significant[7]. The results for OOV are: 1202 tokens (8.1%), 756 types (21.6%) to 894 tokens (6.0%) and 545 types (15.6%).

A possible reason for low improvement in terms of the BLEU scores is because MT evaluation metrics, such as BLEU, compare the MT output with a human reference. The human reference translations in our corpus have been done from English (e.g., En→Es), while the test translations come from a related language (En→Pt→Es), often resulting in different paraphrases of the same English source. While our OOV rate improved, the evaluation scores did not reflected this, because our MT output was still far from the reference even in cases it was otherwise acceptable.

## 4 Conclusions and future Work

We have presented work in progress for developing MT dictionaries extracted from comparable resources for related languages. The extraction heuristic show positive results on the n-best lists that group words with the same starting characters, because the used comparable corpora consist of related languages that share a similar orthography. However, the lists based on the frequency window heuristic show poor results to include the correct translations during the extraction step. Our ML models based on similarity metrics over parallel corpora show rankings similar to heldout data. However, we created our training data using simple heuristics that simulate cognate words (i.e. pairs of words with a small surface difference). The ML models are able to rank similar words on the top of the list and they give a reliable score to discriminate wrong translations. Preliminary results on the addition of the n-best lists into an SMT system show modest improvements compare to the baseline. However, the OOV rate shows improvements around 10% reduction on word types, because of the wide variety of lexical choices introduced by the MT dictionaries.

Future work involves the addition of unsupervised morphology features for the n-best list extraction, i.e. first step, given that the use of starting characters shows to be an effective heuristic to prune the search space and language independent. Finally, we will measure the contribution for all the produced cognate lists, where we can try different strategies to add the dictionaries into an SMT system (Irvine and Callison-Burch, 2014).

## Acknowledgments

The research was funded by Innovate UK and ZOO Digital Group plc.

## References

Hal Daumé, III, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 53–59, Stroudsburg, PA, USA.

Darja Fišer and Nikola Ljubešić. 2013. Best friends or just faking it? Corpus-based extraction of Slovene-

[5]https://github.com/clab/fast_align

[6]https://kheafield.com/code/kenlm/

[7]The p-value for the uk-ru pair is 0.06 we do not consider this result as statistically significant.

Croatian translation equivalents and false friends. *Slovenščina 2.0*, 1.

Oana Frunza and Diana Inkpen. 2009. Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques. *International Journal of Linguistics*, 1(1).

Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013)*, Atlanta, Georgia, June.

Ann Irvine and Chris Callison-Burch. 2014. Using comparable corpora to adapt mt models to new domains. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 437–444, June.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training framework. In *Proceedings of NAACL-2010*, Los Angeles, CA, June.

T. Joachims. 1998. Making large-scale svm learning practical. LS8-Report 24, Universität Dortmund, LS VIII-Report.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA.

Grzegorz Kondrak, Daniel Marcu, and Kevin Knight. 2003. Cognates can improve statistical translation models. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2*, pages 46–48, Stroudsburg, PA, USA.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL*, page 151–158, Pittsburgh, PA, June.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proc. Workshop at ICLR'13*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Serge Sharoff, Reinhard Rapp, and Pierre Zweigenbaum. 2013. Overviewing important aspects of the last twenty years of research in comparable corpora. In Serge Sharoff, Reinhard Rapp, Pierre Zweigenbaum, and Pascale Fung, editors, *BUCC: Building and Using Comparable Corpora*, pages 1–17. Springer.

Jörg Tiedemann. 1999. Automatic construction of weighted similarity measures. In *Proc. Empirical methods in Natural Language Processing and Very Large Corpora*, pages 213–219.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.