Dealing with unknown words in statistical machine translation

João Silva[†⋆], Luísa Coheur[†⋆], Ângela Costa[⋆⋆], Isabel Trancoso[†⋆]

[†]Instituto Superior Técnico, Technical University of Lisbon
[⋆] Centro de Linguística da Universidade Nova de Lisboa
[⋆]Spoken Language Systems Laboratory - L[2]F/INESC-ID
R. Alves Redol, 9 - 2º– 1000-029 Lisboa, Portugal
joao.v.silva@ist.utl.pt, luisa.coheur@l2f.inesc-id.pt, angela.costa@l2f.inesc-id.pt, isabel.trancoso@l2f.inesc-id.pt

Abstract

In Statistical Machine Translation, words that were not seen during training are unknown words, that is, words that the system will not know how to translate. In this paper we contribute to this research problem by profiting from orthographic cues given by words. Thus, we report a study of the impact of word distance metrics in cognates' detection and, in addition, on the possibility of obtaining possible translations of unknown words through Logical Analogy. Our approach is tested in the translation of corpora from Portuguese to English (and vice-versa).

Keywords: statistical machine translation; analogies; cognates

1. Introduction

Statistical Machine Translation systems base their performance in the possibility of finding high frequent patterns of co-occurrences of words. Therefore, unfrequent words have a higher probability of being incorrectly translated. However, in many European Languages, many words have a similar surface form or, at least, sound similar. Moreover, sometimes there are affixes that allow the direct translation of these words (Koehn and Knight, 2002). For instance, in English, the suffix *tion* corresponds to the suffix *ção* in Portuguese, as showed, for instance, in the pair (*intuition*, *intuição*). By the same token, prefixes *hyper* in English and *hiper* in Portuguese match for the same words quite often, as stated by the pair (*hyperactive*, *hiperactivo*). Words that can be translated with one of these strategies are usually **cognates**, that is, words that share the same root, or as it is said by linguists, have a common etymological origin.

Nevertheless, the percentage of cognates between two languages can be low, and other ways to find the translation of unknown words need to be envisaged. Just as there are regular affixes that allow the translation of words between languages, there are also certain analogies between words that can help the translation process of unknown words. As an example, the gerund in English can be obtained by adding *ing* to the end of a verb, as seen in the pairs (*eat*, *eating*) or (*read*, *reading*). Thus, if we manage to translate some of these elements to another language, *e.g.* Portuguese, we can infer the translation of the remaining elements. For instance, if we know the translations (eat_{EN}, $comer_{PT}$), ($eating_{EN}$, $comendo_{PT}$) and ($read_{EN}$, ler_{PT}) we can infer the translation of *reading* as *lendo*. **Logical Analogy** (Langlais and Patry, 2007) is the technique that allows us to establish those inferences and that we explore in this paper, where a framework that provides possible translations of unknown words, by mixing cognate detection and Logical Analogy, is presented. Although English and Portuguese are the target languages, the framework can be adapted to other languages.

The paper is organized as follows: in Section 2. we present the related work that inspired this framework, in Section 3. we describe the framework, in Section 4. we evaluate it and, finally, in Section 5. we point to future work and present some conclusions.

2. Related Work

Several methods to find the correct translation to an unknown word have been proposed in the literature. Here we detach the two strategies that influence our work: the ones that target cognates' detection (Koehn and Knight, 2002; Mann and Yarowsky, 2001; Kondrak et al., 2003; Simard et al., 1992), and the ones that explore Logical Analogies (Langlais and Patry, 2007; Arora et al., 1999).

Usually, cognates are detected by two main methods: the first is based on hand-crafted rules describing how the spelling of a given word should change when it is translated into another language; the second method uses similarity measures between strings, in order to detect cognates. In (Koehn and Knight, 2002), a list of English-German cognates is created by applying well-established mapping rules like the substitution of the letters *k* or *z* in German words by *c* in English. On the other hand, the work described in (Mann and Yarowsky, 2001) uses edit distance for cognate extraction. Both methods are compatible, once the latter can work with words to which the mentioned transliteration transformations were applied.

The analogy strategy is described, for instance, in (Langlais and Patry, 2007). Here, it is used proportional analogy to find translation of unknown words, denoted as $[A : B = C : D]$, which reads "A is to B as C is to D". As an example, to translate the French word *futilité*, we could build the following analogy: $[activités : activité = futilités : futilité]$. Then by the translion of all known words, we would obtain: $[actions : action = gimmicks :?]$, reaching that *gimmick* is a possible translation of the word *futilité*.

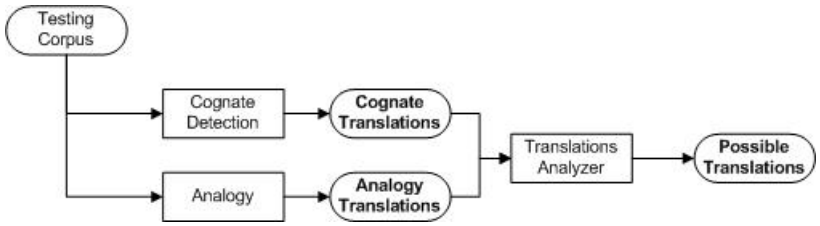

Figure 1: System Architecture.

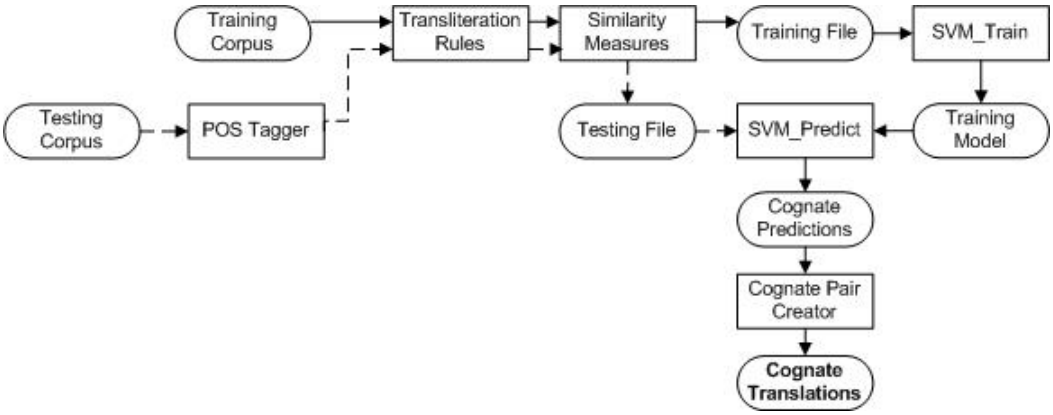

Figure 2: Cognate Detection System Architecture.

3. Finding unknown words' possible translations

The system is divided into two main modules: the first is responsible for cognate detection and the second for finding analogies. At the end both will return a set of words that represent possible translations of a given unknown word, as shown in Figure 1.

3.1. Cognates' detection module

In general terms, the cognates' detecting module works as follows: a training file where pairs of words are manually tagged as cognates or not cognates is given as input. Then, after having applied a set of transliteration rules and calculated several distances between each word in the pairs, the resulting file is given to a Support Vector Machine (SVM) in order to train a model. Using this model the system can then predict which words in two files (testing file) are cognates of each other. A POS Tagger (our framework uses TreeTagger[1]) is used to discard pairs of words that do not belong to the same category. This module is depicted in Figure 2.

Considering the transliteration rules, these are used in a similar way to which it is done in (Mulloni and Pekar, 2006), where orthographic cues are used. These rules determine the substitution of certain prefixes, suffixes or substrings in the middle of words.

Regarding similarity measures, several measures such as Soundex[2] and the Levenshtein Distance (Levenshtein, 1966) are implemented in order to calculate the distance between words.

After the POS Tagger classification of each word, only words with the same category are cognates' candidates. This will guarantee that if, for instance, a name and a verb are considered to be possible cognates, this pair is filtered.

3.2. Analogy module

Here, we follow the work presented in (Langlais and Patry, 2007). This module is built on a monolingual text where analogies between words in the same language are captured, and on a bilingual lexicon that allows to establish translations between words in the source and target languages. With this information, analogies can be inferred. For instance, consider that the word *conditions* is an unknown word. In order to translate it by analogy, we need to have a set of analogies between words of the same language. For instance, we need to know that $[position : positions = condition : conditions]$ – which reads as "*position is to condition as condition is to conditions*". By the same token, we need translations of some of these words. For instance, we need to know the following translations: ($position_{EN}$, $posição_{PT}$), ($positions_{EN}$, $posições_{PT}$) and ($condition_{EN}$, $condição_{PT}$). With this information we can build the same analogy for Portuguese: [posição : posições = condição : ?] and, in this way, we can infer that the translation of *conditions* is *condições*.

In an off-line process, the system learns a set of rules, which represent the prefixes or suffixes that can be used in order to establish analogy relations. Rules are written in the form "$[remove]\backslash[insert]$", where remove and insert represent the characters that need to be removed and inserted into a word to transform it into the other word. For example: (*position*, *positions*) is associated with the rule "$\backslash$s", where $ represent the lack of characters to remove in this case, and (*posição*, *posições*) is associated with the rule "ão$\backslash$ões".

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
[2] http://www.archives.gov/research/census/soundex.html

The next step is to store these rules and their respective word pairs in such a way that it is easy to check what rules can be applied to a new word. We use Tries (Fredkin, 1960) to store all the rules, since that allows strings with the same prefix to have a similar path and, therefore, makes the rule search faster. In Figure 3 we show an example of this, where the words "do", "did", "big" and "dig" are introduced in the same Trie. With this we can see that if two words start with the same sequence of characters, less nodes are needed to represent those two words in the Trie. This can be very helpful in our case since we need to store a large number of rules, many of which are associated with the same prefix or suffix to be removed.

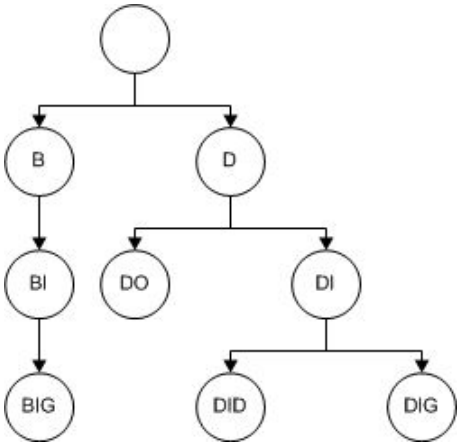

Figure 3: Example of a Trie with the words "do", "did", "big" and "dig".

Instead of inserting words, like in a normal Trie usage, we insert the sequence of characters that we want to remove. This way, each node represents a prefix or a suffix that can be removed from a certain word as stated by the equivalent rule. For instance, if we encounter the rules "|ar\o" ((*comprar*, *compro*)) and "|er\o" ((*correr*, *corro*)), we need to create the nodes "AR" and "ER". By adding these nodes we first need to create the node "R". we insert the last character because it is a suffix rule and, therefore, we start reading the suffix from the last character backwards. Since the rules exist both ways and "|ar\o" can also be the rule "|o\ar", if we switch the pair of words that created the rule, we also insert the node "O". This results in the Trie shown in Figure 4. These nodes contain the information of the characters to remove and a list of possible characters to insert.

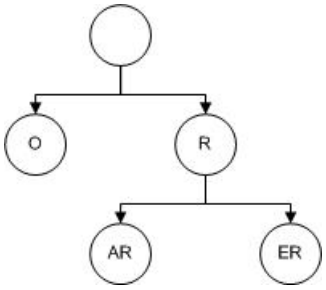

Figure 4: Trie with nodes O, R, AR and ER.

When we find an unknown word, the system goes down both to the prefix and the suffix Trie, finding all the nodes that can be reached by this word and that have characters that can be inserted in it. When it reaches a node with one or more rules associated, it creates analogies with other words associated to those rules. Each rule may also create one or more analogies, depending on the number of words associated with it.

For instance, if the word "salto" is an unknown word and the system had the suffix Trie represented in Figure 4. This word matches the suffixes in the empty node, as well as node "O". There is no rule associated with the empty node, but there are two rules in node "O". By applying these two rules, we obtain two different words, "saltar" and "salter". Using the original pairs of words that created these rules we can generate two different analogies: [salto : saltar = compro : comprar] and [salto : salter = corro : correr]. Since the word "salter" does not exist, that analogy can immediately be discarded.

An analogy will have a resulting translation suggestion and an attributed score, which depends on the similarities between the words in the analogy and if the suggested word is a known word or not. For instance, the word "reduce" and the word "reduc" are both suggested translations to the Portuguese word "reduzirem", however, since "reduce" is a known word, its score is higher than the score of "reduc". Since various analogies can have the same suggested word as output, in the end we sum all the scores of those analogies to give a final score to the translation. With that, we can compare the scores of all possible translations and return an ordered list of the most probable translations.

4. Evaluation

4.1. Experimental setup

The evaluation of the cognate's detection module was made with 19 economic and 19 politic news extracted from the Euronews website[3], both in English and Portuguese. Cognates were manually extracted from these corpora (Carvalho, 2010) and 15 news of each domain were used for training (and the remaining 4 were used for testing).

Concerning the analogy module, the evaluation was made using Europarl [4] parallel texts. To create the bilingual lexicon we extracted all the unique words from 10000 sentences of Portuguese and English (Europarl) texts. To have a legitimate translation of all these words we have inserted them into Google Translate [5] and created a bilingual lexicon with almost 20000 entries from it.

For the testing phase we also used the following 100 sentences from Europarl (both in English and Portuguese). We then tried to match each word from this corpus with the elements of the bilingual lexicon. If the word could not be found there, this meant that it was an unknown word and that we should try to find a translation to it through analogy. With this process we obtained a total of 44 Portuguese and 9 English words.

As we also wanted to test the analogy module in a real life scenario, we gathered the words that were left untranslated from a statistical machine translation system applied

[3] http://www.euronews.net/

[4] http://www.europarl.europa.eu

[5] http://translate.google.com

Measure	Precision	Recall	FMeasure
LCSM	70.7%	63.1%	66.7%
Levenshtein	59.7%	68.5%	63.8%
Soundex	37.6%	74.6%	50.0%
LCSR	30.0%	86.2%	44.5%

Table 1: Individual evaluations of the top 5 similarity measures.

to translate a TedTalk, a set of questions and a touristic magazine, from english to portuguese.

4.2. Evaluating the Cognate's detection module

In a previous work (Carvalho, 2010), the author started by using 11 similarity measures. However, he then decided to evaluate how each would behave on their own. After the experiment, he found that *Soundex*, *Lcsm*, *Lcsr* and *Levenshtein* are the top 4 similarity measures, with their results shown in Table 1. So, these are the only ones used for classifying cognates.

The usage of the POS Tagger decreases the number of false positives found, which are the number of pairs that were considered cognates when in fact they are not. Using the POS Tagger, we can also see that the number of cognates missed has increased, this is due to the fact that the Tagger can make mistakes when attributing a category tag to a word in Portuguese different to what it attributes to the same word in English. Errors like this can later be corrected in the Analogy phase. In Table 2, we can see that the recall of the system has decreased after introducing the tagger, due to the number of cognates missed, but that is compensated with an increase in precision, from the decrease of false positives found. Combining these scores and calculating the *FMeasure* we can see an overall improvement of the system.

	Precision	Recall	FMeasure
Before Tags	71.7%	55.9%	62.8%
After Tags	76.5%	55.36%	64.3%

Table 2: Comparison between the statistics of cognates before and after the introduction of the POS Tagger.

4.3. Evaluating the Analogy module

As explained in the Evaluation Setup, we have extracted 44 Portuguese and 9 English unknown words. These words were the target of the analogy module. Results show that 26 out of the 44 Portuguese unknown words (59.1%) have a valid translation as the top scored word returned by the analogy module. 4 words (9.1%) also have a valid translation somewhere among the translations returned by the analogy module, and 10 unknown words (22.7%) were left without even a possible translation. The translation of the remaining 4 words, although obtaining invalid translations, result in words that give a good idea of the meaning of the unknown word. Examples of words in all these cases can be seen in Table 3.

Unknown Word	Best Scored Translation(s)	Score (%)
aprazar-me-ia		
compreenderiam	**understand**	93%
desejar-vos	wishe	100%
disponibilizada	**available**	54%
	provided	26%
dívidas	debt	62%
	debts	32%
interrupção	**interruption**	93%
perturba	**disturbs**	29%
	derailes	12%
	disturbes	11%
	derails	11%
reduzirem	**reduce**	61%
	reduc	11%

Table 3: Analogy scores for 8 unknown words (words in bold represent valid translations).

Since the output of the module is a ranked list of translations, the best way to evaluate the results is by using the Mean Reciprocal Rank (Voorhees, 2008) (MRR). The MRR is mostly used in question answering, but by using Equation 1 on the ranked lists we obtained, we have an MRR of 0.63.

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \quad (1)$$

Looking at the far fewer examples with English unknown words, there are 5 out of 9 (55.6%) with a valid translation as its top scorer. This shows that, even though the test sample was much smaller than the Portuguese one, the resulting percentage of correct translations is still very similar, giving us a good perspective on the overall results of this module. Since the remaining 4 words either have no possible translations or no valid translation, the MRR is 0.56.

In what concerns the real scenario, 49 out of 101 words, had a correct translation in the top 1 and the MRR was 0.42.

5. Conclusions and Future work

SMT systems have to deal with unknown words, that is, words that were not seen during training. Thus, having a system that proposes translations to these unknown words can improve SMT systems' results.

One of the ways to find translations of unknown words is to find possible translations of these words in parallel corpora. If two words are considered to be cognates, there is a strong possibility that they are translations of each other. In the framework described in this paper, we used a set of similarity measures to determine if two words are cognates. However, this is not an easy task, since there are a number of false cognates and also because many words that are translation of each other are not cognates. The cognate detection using a POS Tagger, manages to correctly determine 55% of the total cognates that exist, however it also assumes as cognates many other words that are not. Thus, we have implemented a module that follows the Logical Analogy

paradigm in order to find possible translations of unknown words.

This module was able to find translation to 68% of the Portuguese unknown words found on the same website as the training, with an MRR of 0.63. When evaluating for English unknown words that were extracted from a different context, the results lowered, finding translation to 46% of these words and resulting in an MRR of 0.42.

In the future we plan to merge these two modules, attempting to take advantage of the characteristics of both approaches. Another improvement that can be made is adding the top scored translation of an unknown word to the bilingual lexicon, this word could then help to find the translation of other unknown words in the future.

6. References

K. Arora, M. Paul, and E. Sumita. 1999. The trec-8 question answering track report. In *in Proceedings of TREC-8*, pages 77–82.

L. Carvalho. 2010. Criação de léxicos bilingues para tradução automática estatística. Master's thesis, Instituto Superior Técnico.

E. Fredkin. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *In Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*, pages 9–16.

Grzegorz Kondrak, Daniel Marcu, and Keven Knight. 2003. Cognates can improve statistical translation models. In *In Proceedings of HLT-NAACL 2003*, pages 46–48.

P. Langlais and A. Patry. 2007. Translating unknown words by analogical learning. In *in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 877–886.

V. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *in Soviet Physics Doklady*, pages 707–710.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *NAACL*.

Andrea Mulloni and Viktor Pekar. 2006. Automatic detection of orthographic cues for cognate recognition.

Michel Simard, George F. Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *in Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81.

E. M. Voorhees. 2008. Translation of unknown words in phrase-based statistical machine translation for languages of rich morphology. In *in Proceedings of the First International Workshop on Spoken Language Technologies for Uner-Resourced Languages*, pages 70–75.