Statistical Machine Translation - the problem of unknown words

João Silva

Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais, 1 – 1049-001 Lisboa, Portugal
joao.v.silva@ist.utl.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. We always take advantage of the information that can be gathered by the words surrounding an unknown word, by the use of context. Our approach is tested in the translation of corpora from Portuguese to English (and vice-versa).

Keywords: statistical machine translation; unknown words; analogies; cognates; context

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*<sub>EN</sub>, *comer*<sub>PT</sub>), (*eating*<sub>EN</sub>, *comendo*<sub>PT</sub>) and (*read*<sub>EN</sub>, * Ler*<sub>PT</sub>) 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.

Various other ways to find translations of unknown words exist. For instance, using the *context* of each word (Fung and Yee, 1998), that is, the words that precede and succeed the unknown word. As an example, the word “driving” might be close to the word “car”, just as the word “condizir” is close to the word “carro”. Using these characteristics we can assume that two words with similar contexts are related somehow.

In this project we attempt to mix different methods that allow us to translate unknown words. If we use each method’s advantages to compensate for another method’s flaws, we can improve the results of the system as a whole and offer better translations for the unknown words.

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 translating all known words, we would obtain: *actions : action = gimmicks : ?*, reaching that *gimmick* is a possible translation of the word *futilité*. 
(Fung and Yee, 1998) show that it is possible to determine which word is more likely to be a translation of another word by comparing the contexts of each of the words in two corpora. The context is assumed to be the closest words around the selected unknown word. The authors try to achieve translations between English and Chinese and notice that the context for the word flu in Chinese very similar to the context of the word flu in English than to the context of the word Africa in English. This shows that context is a good indication of what words are possible translations.

3. Finding unknown words’ possible translations

The system is divided into three main modules: the first is responsible for cognate detection, the second for finding analogies and the third for comparing contexts. At the end, all will return a set of words that represent possible translations of a given unknown word and the system uses these results to rank the suggested translations.

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 1.

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 Pastry, 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 position 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] / [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 “$Ss$”, where $S$ represent the lack of characters to remove in this case, and (posição, posições) is associated with the rule “ão\(\)ões”.

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 2 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 2: Example of a Trie with the words “do”, “did”, “big” and “dig”](image)

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, compra) 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

\(^1\)http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
\(^2\)http://www.archives.gov/research/census/soundex.html
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 3. These nodes contain the information of the characters to remove and a list of possible characters to insert.

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 3. 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.

3.3. Context

The other completely new module we have implemented is the one that is based on context. It works similarly as to what is explained in (Fung and Yee, 1998). The general idea surrounding the use of context is that, if two words are related, there is a good probability that the words before and after each word are the same too. We can use this knowledge to find translations of words in parallel or comparable corpora.

As an example, consider the word "fisherman". Not knowing the meaning of this word, we can look at its context. The word is normally inserted in sentences such as "the fisherman caught the fish" or "the fish brought in by a fisherman". By using these sentences we can extract the context list for the word "fisherman". In this list we include the two closest words to the left and to the right of the unknown word, taking into consideration that only adjectives, adverbs, names and verbs give us any contextual information of the word.

The list is normalized, to prevent that words that appear many times in the corpus are considered translation of all other words. For example, the word "is" appears in numerous sentences and, therefore, has a much bigger number of entries in the context list than a word such as "fisherman". If the context list of is had one entry of each word: "caught", "fish" and "brought", it could be considered similar to the word "pescador". However, if instead we normalize each entry, each word has a much lower score and is less likely to be associated to "pescador". This way we guarantee that a words’ high frequency will not turn into this same word being considered a valid translation numerous times.

The context list of "fisherman" would then be as follows:

- "caught" - 25%
- "fish" - 50%
• “brought” - 25%

Considering the Portuguese equivalent of “pescador”, inserted in sentences such as “o pescador apanhou o peixe” and “o peixe trazido por um pescador”, the context list is:

• “apenhou” - 25%
• “peixe” - 50%
• “trazido” - 25%

Each of the unknown words will have its context list compared to the context lists of all words in the opposite language. By looking at each word in each list and its respective percentage, we can calculate the similarity between two context lists. If we compare these two context lists using the lexicon, we can see that “apenhou” is the translation of “caught”, “peixe” is the translation of “fish” and “trazido” is the translation of “brought”. So, by looking at the two context lists, we see they are exactly the same and thus these two words are likely to be a translation of each other.

3.4. Merging the Modules

The final decision on what is the best translation to a given unknown word will be taken by the Translations Analyzer. This module will see the output of all 3 methods used in the system and, taking into consideration each result, output a ranked list of the most probable translations for the unknown word.

After the system’s evaluation (described in depth in Section 4.), we came to the conclusion that the context module does not output viable results. This fact, along with the added time it takes for this module to calculate a score, led us to decide to cut the context module from the final system.

Since the cognate evaluation is based on words that exist in the lexicon, many of the valid analogy words would not have a cognate score associated to them which would end up lowering their final score. For instance, the word “assistants” is a valid translation of the word “assistentes” and the analogy module does suggest it as a translation. However, since the word “assistants” does not exist in the lexicon we used for testing, the corresponding score of the cognate module is 0. Due to this, we have decided to run the cognate model not only over all the words in the lexicon, but also over all the words suggested by the analogy module. Since the words “assistants” and “assistentes” are written in a similar way, the cognate module will attribute a high score to this pair, which will end up raising its overall score. The final architecture of the system is shown in Figure 4.

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 website3, 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).

4.2. Evaluating the Cognate’s detection module

Concerning the analogy module, the evaluation was made using Europarl4 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 Translate5 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. We ended up using 10 Portuguese and 4 English words as a development set, to tune the factors that calculate scores, therefore the test set consists of 34 Portuguese and 5 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 to translate a TedTalk, a set of questions and a touristic magazine, from English to Portuguese.

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

<table>
<thead>
<tr>
<th>Measure</th>
<th>Precision</th>
<th>Recall</th>
<th>FMeasure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCSM</td>
<td>70.7%</td>
<td>63.1%</td>
<td>66.7%</td>
</tr>
<tr>
<td>Levenshtein</td>
<td>59.7%</td>
<td>68.5%</td>
<td>63.8%</td>
</tr>
<tr>
<td>Soundex</td>
<td>37.6%</td>
<td>74.6%</td>
<td>50.0%</td>
</tr>
<tr>
<td>LCSR</td>
<td>30.0%</td>
<td>86.2%</td>
<td>44.5%</td>
</tr>
</tbody>
</table>

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

Concerning the analogy module, the evaluation was made using Europarl4 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 Translate5 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. We ended up using 10 Portuguese and 4 English words as a development set, to tune the factors that calculate scores, therefore the test set consists of 34 Portuguese and 5 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 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.

3http://www.euronews.net/
4http://www.europarl.europa.eu
5http://translate.google.com
4.3. Evaluating the Analogy module

As explained in the Evaluation Setup, we have extracted 34 Portuguese and 5 English unknown words. These words were the target of the analogy module. Results show that 20 out of the 34 Portuguese unknown words (58.8%) have a valid translation as the top scored word returned by the analogy module. 3 words (8.8%) also have a valid translation somewhere among the translations returned by the analogy module, and 8 unknown words (23.5%) were left without even a possible translation. The translation of the remaining 3 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.

<table>
<thead>
<tr>
<th>Unknown Word</th>
<th>Best Scored Translation(s)</th>
<th>Score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aprazar-me-ia</td>
<td>understand</td>
<td>93%</td>
</tr>
<tr>
<td>compreenderiam</td>
<td>understand</td>
<td>93%</td>
</tr>
<tr>
<td>desejar-vos</td>
<td>wishe</td>
<td>100%</td>
</tr>
<tr>
<td>disponibilizada</td>
<td>available</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>provided</td>
<td>26%</td>
</tr>
<tr>
<td>divisas</td>
<td>debt</td>
<td>62%</td>
</tr>
<tr>
<td></td>
<td>debts</td>
<td>32%</td>
</tr>
<tr>
<td>interrupcão</td>
<td>interruption</td>
<td>93%</td>
</tr>
<tr>
<td>perturba</td>
<td>disturbs</td>
<td>29%</td>
</tr>
<tr>
<td></td>
<td>derailes</td>
<td>12%</td>
</tr>
<tr>
<td></td>
<td>disturbs</td>
<td>11%</td>
</tr>
<tr>
<td></td>
<td>derailes</td>
<td>11%</td>
</tr>
<tr>
<td>reduzirem</td>
<td>reduce</td>
<td>61%</td>
</tr>
<tr>
<td></td>
<td>reduc</td>
<td>11%</td>
</tr>
</tbody>
</table>

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

Looking at the far fewer examples with English unknown words, there are 3 out of 4 (60.0%) 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 0.60.

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.

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.

\[
MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}
\]
any correct translation amongst its results. So, in the end, only 11 words had at least one suggestion that was related to the unknown word. If we use Equation 1 to figure out a value for MRR, we reach a result of 0.28. However, considering there are many tied words, we can calculate the MRR with the lowest rank that word could achieve; for instance, a word ranked first but tied with 15 other words would have its lowest possible rank as 16. Using these ranks, the MRR would then be only 0.1, which is a very low result. If we use the same 100-line corpora, but translate the unknown words in English, the system only finds 12 words to translate; out of those, 1 has no suggested translation and 9 have no valid translations, leaving only 2 words with valid translations. The MRR is 0.11 and, when using the lowest possible ranks, it goes down to 0.05.

Since the number of words is limited, a word appears in average only one time in the whole text. This makes the context list not very accurate and, therefore, outputs a lot of possible translations that, in fact, only have one word in common with the unknown word.

We then evaluated the 10000-word corpora, translating from Portuguese to English. In the Portuguese to English evaluation, we see a decrease in the MRR, going down to 0.06 with the best rank the word can achieve and 0.02 if we calculate it with the worst rank. One possible explanation of this decrease is that the context is so big that hundreds of words are associated to that unknown word, but also many of the unknown words were verbs that do not have a specific context associated to them, such as “obterem” or “enti”, which makes it harder to find a translation through context.

Evaluating the same 10000-word corpora, translating from English to Portuguese, we obtain slightly better results, reaching an MRR score of 0.22 in the best case and 0.12 in the worst case scenario. These results are more in tune with what we expected would happen.

### 4.5. Evaluating the Merged System

The lexicon and the unknown words used to evaluate the system after merging, were the same that were used in the Analogy module evaluation, to guarantee that we could compare these results to the results of using of analogy by itself.

The context module was not included in the merged system due to its poor results, as shown in the previous section. Considering it needs extra data (the context lists) as well as extra uploads by the user (corpora with the unknown words so the context can be extracted), we felt that its results were not good enough to compensate making the system more complex.

The cognate module’s results by itself, outputs less accurate translation suggestions than when compared with the analogy module’s results. However, there are some valid translations that only the cognate module can recognize. For this reason, we have decided to merge both these modules, but give them different weights. After evaluating various possible weights, the one that maximized the MRR of the Portuguese unknown words was leaving the cognate’s weight at 1, but setting the analogy’s weight to 3, resulting in an MRR of 0.70.

The main words that cause an increase in the MRR, when compared to the results of the analogy module, are:

- “congratula-se”: the word has no translation with the analogy module, but using the merged module, it correctly guesses “congratulates” as the most likely translation;
- “pátio”, similarly: the system only finds the translation “patio” when using the cognate module;
- “rectificar”: even though “rectify” is only ranked as the second most likely translation, it is still an improvement compared to the no valid translations with the analogy module.

When testing with the real life scenario explained at the end of Section 4.1., we can also slightly improve the MRR. Instead of being 0.395, we reach 0.417 with the default values we have chosen. This is not such a clear increase as with the Portuguese evaluation, but the words “fiber”, “liters” and “mortuary” now have valid translations, while when using solely the analogy module, the system did not return any suggested translation.

### 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.

A third module was developed, using context to attempt to find a relation between two words. Two words that are similar, even in different languages, will likely be surrounded with other similar words. However, the results of this module were much lower than expected, reaching a best case scenario of 0.22 as the MRR. These poor results led us to decide not to include the context module in the final system.

We then merged the cognate and the analogy modules, using certain characteristics from both and giving the analogy module a bigger weight in the results of the system. We
managed to improve the results previously obtained by testing the systems individually. Using the same two scenarios to translate Portuguese and English unknown words, the MRR rises to 0.70 when translating the Portuguese words and to 0.417 when using the English unknown words extracted from an SMT.

The main improvement that could be done in future work has to do with the context module. One of the main reasons for the low results comes from the fact that the words that surround the unknown word can be words that appear too often. An extra step that could be made would be to use the frequency of a word in its score. For instance, verbs such as “is” or “does” are used more often than other verbs such as “fishes” or “drives”. This increase in frequency makes it harder to associate the verbs to a specific word since they appear in the context of most words. If we could decrease the score as the frequency increases, the results of the context module could increase.

Even though the system focuses on translations from English to Portuguese and vice-versa, it would be interesting to have a system that can be easily adapted to any other languages. To allow the system to be used by other languages we need to introduce different transliteration rules in the cognate module. If, instead of writing the transliteration rules, the system could calculate them before the cognate module, we could use the system with any pair of languages.

Regarding the analogy module, the method chosen to deal with words that have no suggested translation was to switch the last character of a word (in case it is a vowel), by the different vowels to see if a translation could be obtained. Another possible approach would be to attempt to use rules that were previously discarded for not having enough word pairs associated to them.

6. References


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

