

Metodološki zvezki, Vol. 5, No. 1, 2008, 65-80

Parse Tree Based Machine Translation for Less-used Languages

Jernej Vičič¹ and Andrej Brodnik²

Abstract

The article describes a method that enhances translation performance of language pairs with a less used source language and a widely used target language. We propose a method that enables the use of parse tree based statistical translation algorithms for language pairs with a less used source language and a widely used target language.

Automatic part of speech (POS) tagging algorithms have become accurate to the extent of efficient use in many tasks. Most of these methods are quite easily implementable in most world languages.

The method is divided in two parts; the first part constructs alignments between POS tags of source sentences and induced parse trees of target language. The second part searches through trained data and selects the best candidates for target sentences, the translations.

The method was not fully implemented due to time constraints; the training part was implemented and incorporated into a functional translation system; the inclusion of a word alignment model into the translation part was not implemented.

The empirical evaluation addressing the quality of trained data was carried out on a full implementation of the presented training algorithms and the results confirm the employability of the method.

1 Introduction

Machine translation (MT) represents the usage of computers, any kind of usage, as tools for translating texts from a source natural language to a target natural language EAMT (2008). A contemporary survey of the machine translation field Sanchez-Martinez et al. (2007) divides the machine translation paradigm into two major subfields: Rule-Based Machine Translation (RBMT) and Corpus-Based machine translation (CBMT).

RBMT systems rely on a, possibly big number, of hand-crafted rules. These systems have been among the best performing machine translation systems in the

¹ University of Primorska; jernej.vicic@upr.si

² University of Primorska; andrej.brodnik@upr.si

past, but lately newer technologies, based on algorithms that extract translation knowledge from big corpora, are prevailing.

Statistical machine translation (SMT) as defined in Al-Onazian (1999) has become one of the most studied fields of CBMT. SMT is based on statistical models whose parameters are derived from the observation of bilingual parallel corpora. Statistical machine translation by parsing (SMTbyP), as described in Melamed (2004), represents a subfield of SMT where statistical models' parameters are derived from the analysis of syntactically annotated bilingual parallel corpora.

SMTbyP is one of the most promising directions of statistical machine translation (SMT) Brown et al. (1993) and Melamed (2004) and machine translation (MT) in general. The most important advantage of such systems over traditional SMT systems is in the ability to handle recursive structures in sentences. Parsing models are used in syntactical tree production and parsing. Most state-of-the-art parsing models Collins (2003) and Charniak (2000) are trained on previously prepared syntactically annotated corpora (treebanks) like Marcus (1993). Less used languages still lack translated treebanks.

We propose a method that enables the use of parse tree based statistical translation algorithms for language pairs with a less used source language and a widely used target language (a language with a treebank).

Our method uses POS tags of a source sentence as additional information to identify a target parse-tree and produce a hierarchical alignment. The method is divided in two parts. The first part is used during the training phase to learn the alignments on training data. All target training sentences are parsed using a parser Collins (2003) previously trained on a large treebank Marcus (1993). Source sentences from the training corpus are tagged with a POS tagger and aligned with nodes in the corresponding parse trees. Translation is done in the second part where trained alignments are used.

The method was implemented and incorporated in GenPar Burbank (2005), a SMTbyP toolkit. It was tested on a text from a classic novel "1984" Orwell (1949) from MULTEXT-East corpus Erjavec (2004). The empirical evaluation was done on the language pair Slovenian-English.

The first part of the article describes the research area; the motivation is introduced in the following chapter. The main part of the article describes the method with the empirical evaluation. The article concludes with some final conclusions and a description of further work.

2 Known results

2.1 Research area description

Designers of SMT systems have begun to employ parse tree translation models due to a growing awareness in the SMT research community that major advances can come only from a deeper understanding about the relationship of our models to the phenomena being modeled.

Melamed (2004) proposes a reduction of the conceptual complexity of tree-based translation models naming the new area Statistical Machine Translation by Parsing - SMTbyP. GenPar, a complete system construction toolkit has been developed following Melamed (2004) guidelines.

The prerequisite for a SMTbyP system is a parallel, sentence aligned, bilingual corpus and sentence aligned bilingual treebank of the source and target language pair Melamed (2004).

A basic SMTbyP system is composed of two stages: the training stage and the translation stage.

The first stage - the training stage - uses a syntactical parser such as in Collins (2003) and Charniak (2000) that has been previously trained on a large treebank like in Marcus et al. (1993). Each sentence in the source and the target part of the corpus is parsed; the results are pairs of source and target parse trees from aligned sentences. The next step constructs hierarchical alignments between the source and the target parse trees. A statistical word alignment model Brown et al. (1993) and Wu et al. (2005) is used to model word alignments in the corpus. Training data is stored for later use in translation stage.

The second stage – the translation stage - constructs a parse tree of the input sentence in the source language, the appropriate target parse tree using the training data and replaces the source words with the target words using word alignment model.

The RBMT systems are based on rules that employ explicit linguistic knowledge in the process of translation. Morphological information, mostly in the form of POS tags presents the basis for translation.

Most of the models in SMT and SMTbyP are not suitable for morphologically rich languages like Spanish or Southern Slavic languages like Slovenian. Niessen and Ney (2001) report that the introduction of morphological information improved the overall translation quality. Ueffing and Ney (2003) introduced the morphological information (POS tags) into a SMT system and report major improvements in translation quality. Toutanova et al. (2002) propose usage of several tags, among them the POS tags, that enhance the translation quality of a basic SMT system.

2.2 Language pair Slovenian – English

Slovenian language is a small language, spoken by two million people, mostly in Slovenia. The language technology resources for such a small language are naturally limited. There are a few parallel tagged corpora available such as Erjavec et al. (2003) and Erjavec (2006), mostly paired with English language. A POS tagger has been developed constructing the referential corpus FIDA Erjavec et al. (1998) on the basis of recommendations and tag-sets from Erjavec et al. (2003). The tagger is not available for public usage but a POS tagger trained on a smaller corpus is freely available Erjavec et al. (2000). A small syntactic treebank Ledinek and Zele (2005) is also available.

English language is the best supported language by language technologies resources like big corpora, syntactic treebanks, tested language processing methods and tools and is the most widely used language in the electronic media

There are big differences between these two languages, particularly on the syntactical level that should be best handled by parse tree based translation models Melamed (2004).

3 Motivation

Machine translation that uses syntactical information in the translation process, in particular SMTbyP uses syntactical parsers such as Collins (2003) and Charniak (2000) to construct parse trees that are used as the basis for translation. Such parsers or parsing methods are not available for the majority of natural languages.

Automatic POS tagging algorithms have become accurate to the extent of efficient use in many NLP tasks. Most of these methods can be reused on a new language although the process is not straight-forward (the development of a new tagset or the adjustment of an existing tagset, the development of an annotated training corpus). These technologies have already been developed for most European languages through the projects MULTEXT Ide and Veronais (1994) and MULTEXT-EAST Erjavec et al. (2003).

The POS tagset varies among definitions and among different languages, but most tagset definitions can be translated from one definition to another using simple translation tables. Most syntactical parsers use the PENN treebank Santorini et al. (1993) tagset. MULTEXT-EAST Erjavec et al. (2003) project defined the morphosyntactic descriptors (MSD), the same tagset was used for all languages of the project. The MSDs contain the same morphological information as a common POS tag, but also include the syntactical information.

Our hypothesis is that the POS tags contain enough syntactical information to support word abstraction in the training corpus. Words are modeled in a separate

model. The search space is greatly reduced using only POS tags instead of real words, therefore less data is needed to efficiently model the translation rules.

A POS tag string can be constructed from a sentence tagged with the POS tags abstracting original words. Such strings represent leaves in parse trees (abstracting the original words).

SMTbyP constructs a parse tree from the source language sentence and aligns it to target sentence parse tree. Our approach uses same algorithms with the only difference: the POS string constructed from the source sentence and is aligned to the target sentence parse tree.

The presented method uses POS tags of a less used language sentence to model the source sentence to the target parse tree alignment.

4 Method description

Most of the methods in SMT are language independent and work in both ways from the source language to the target language and vice-versa. Language independency is achieved by inducing translational knowledge from parallel data with no additional language knowledge.

The method, described in this article, lacks in both universalities. It expects a language with a treebank as the target language, one of the world's mostly used languages, and a language with a solid POS tagging technology. The method is divided in two parts.

First part constructs alignments between POS tags of source sentences and the induced parse trees of the target language.

Second part searches through trained data and selects the n-best-set³ of possible candidates of target translations.

4.1 Training

Translation model is trained on a bilingual parallel corpus such as Erjavec et al. (2003). The corpus consists of source-target sentence pairs, an example is shown in Figure 1.

A standard SMTbyP algorithm constructs a parse tree from source language sentence and aligns it to the target sentence parse tree. The words are modeled in a separate model where basically any available word-by-word alignment model can be used. Our approach changes only the actions involving source sentences as the presumption of the method was that there was no syntactic parser available for the source language. Word alignment model is presented in the next section.

³ a set of predefined number of parse trees based on score values

Each training sentence pair is handled separately. Target sentence is parsed using a parser Collins (2003) previously trained on a large treebank Marcus (1993) as described in Section 1, producing parse trees with confidence score for each target sentence. An example of a parse tree is presented in Figure 1.

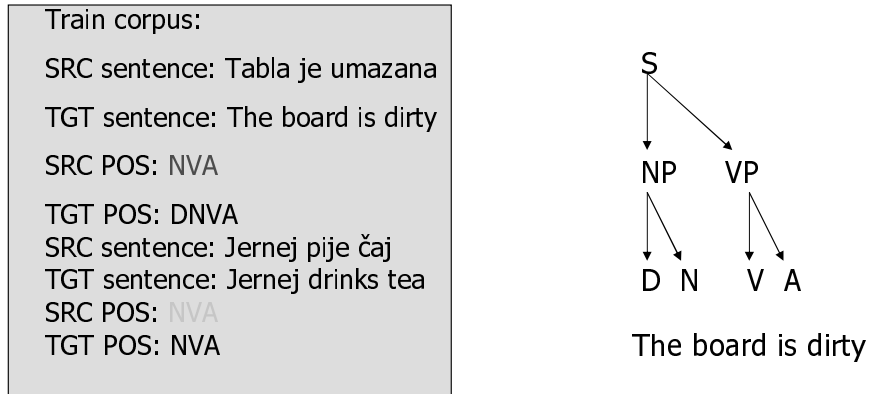


Figure 1: Bilingual aligned corpus on the left-hand side and parse tree on the right hand side. POS tags of a whole sentence are glued into symbol words: NVA means NounVerbAdjective phrase, DNVA DeterminerNounVerbAdjective. The same symbols are used in the parse tree example.

Parse trees consist of words in leaves; POS tags in the first level; POS tags are grouped into phrases that form next level. Each word has a corresponding POS tag. Abstraction of words in parse trees represents almost no informational loss from syntactical point of view. The inner nodes denote grammar symbols.

No parser is available for source language, source sentence is POS tagged, in our testing system we used an already tagged corpus; POS tags were extracted from the corpus. This sequence produces tuples of a form as shown in example a) in Figure 2.

- a)**
 <source sentence, target sentence, source POS tags,
 target parse tree, parse tree score>
- b)**
 <source sentence, target sentence, source POS tags,
 target parse tree, alignment, parse tree score,
 alignment score>
- c)**
 <tabla je umazana, the board is dirty, NVA, DNVA, (S
 (NP (D)(N))(VP (V)(A))), binary data, 0.786, 0.354>

Figure 2: Example a) shows partial training data; example b) shows final training data with scored alignments; example c) shows the first example from Figure 1 presented as final training data, the alignments are in binary format.

The next phase aligns POS tags of each source sentence with the inner nodes in corresponding target parse tree. The algorithm is shown in Figure 3 and an example alignment is shown in Figure 4.

```

for each source/target pair{
  sourcePOS = produceSourcePOSString(source)
  targetPT = produceTargetParseTree(target)
  for all substrings of sourcePOS{
    find the longest match in sourcePOS and
    lowest level of targetPT
    climb as far as possible in the target parse
    tree to still include the whole match
    store the alignment
  }
}

```

Figure 3: The alignment algorithm. Find the longest substring match between the source POS string and the lowest level of target parse tree which is target POS string. Climb as far as possible through the target parse tree to still include the whole substring match. Align the node in the target parse tree with source POS substring. Repeat the procedure until all source POS symbols are aligned.

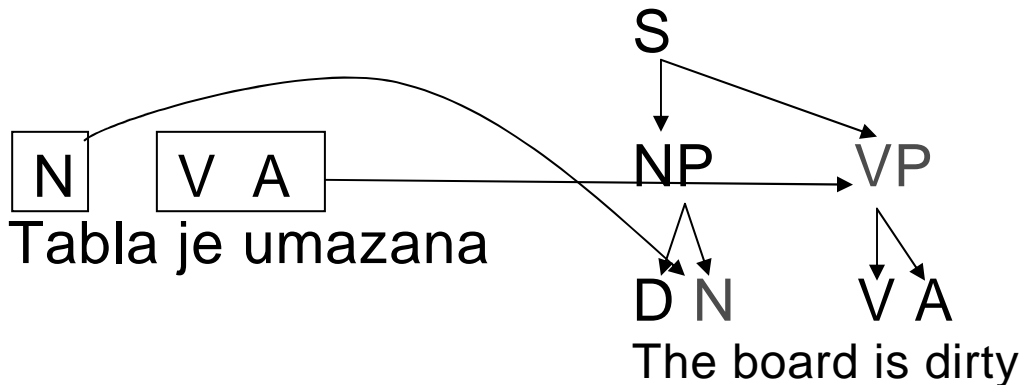


Figure 4: The POS tags of the source sentence are aligned with the inner nodes of the target parse tree. In this example the first source POS symbol (N) is aligned with the second POS symbol (N) in the target parse tree, remaining two symbols in the source POS string are aligned with a whole phrase in the target parse tree.

Alignments are scored according to a set of rules used in their production (each rule has a weight). The final product is a tuple shown in example b) in Figure 2.

4.2 Word alignment model

The word alignment model is basically a lexicon that assigns probabilities to each word pair (source-target).

Parts of the translation method rely heavily on the quality of word translations so the selection of a good word alignment model is crucial. IBM1 word alignment model Brown et al. (1993) implemented in GIZA++ Och et al. (2003) was used.

4.3 Translation

This phase translates the input source sentence into, hopefully suitable, sentence in the target language. The input sentence, the sentence to be translated, is POS tagged.

```
<source sentence, target sentence, source POS tags,  
target parse tree, parse tree score, search score>
```

Figure 5: Temporary translation data.

The string of POS tags is searched in the training data. The simple search is augmented with the substring search and the similar string search. These methods are described in the following chapter. The results are scored according to the search method used (methods are weighted). The results of the search are the n-best-set of tuples in the form as shown in Figure 5.

Each tuple is independently used to produce a translation candidate in the last step of the translation method. The words of the target parse tree are combined through alignment with the words in the source sentence and later translated using the word alignment model. Translations are scored using the already accumulated scores during the training phase and multiplied by language model Clarkson et al. (1997) probability of the translation candidate. The best scored candidate is selected as the final translation.

4.4 The similar string search

The full string search often fails to find any translation candidate as the training corpus is relatively small in comparison to the language sentence set used. POS strings that have Levenshtein-edit distance Levenshtein (1965) at most a fixed small value, usually 1 or at most 2, are used as possible candidates. These candidates are scored with a penalty and later used as full string candidates.

4.5 The language model

A language model models sentences of one language. Basically it assigns a probability that a sentence is part of a language. CMU-Cambridge Statistical Language Modeling toolkit Clarkson et al. (1997) was used to model the target language. The probability produced by this model is multiplied by each translation candidate's score. The best solution becomes "the translation".

5 Empirical results

Three problems were addressed in empirical testing:

- translated POS string quality
- success rate of the POS symbol word search
- impact of the size of the training set on the success rate of the POS symbol word search

Each problem is presented in greater detail in section 5.3.

5.1 Experimental setting

The already available tools were used in the construction of the testing environment where possible. Many applications were suitably modified to suit the testing environment needs. A new module, the implementation of the presented method was developed and incorporated in GenPar system.

A short description of the testing environment follows:

- GenPar, a system for SMTbyP – Statistical Machine Translation by Parsing Melamed (2004), was used as the base translation system.
- Levenshtein (edit) distance Levenshtein (1965) metric was used in the quality estimation of the extracted POS tag strings.
- Corpus MULTEXT–East, Erjavec et al. (2003), which includes the annotated and tagged novel 1984 by George Orwell in several Eastern European languages and in English, was used as the training and the testing corpus for the method evaluation process. This is a relatively small corpus, around 6.000 sentences in total, but it is manually checked for errors and sentences are correctly formed. Corpus is MSD tagged Erjavec et al. (2003) including standard POS
- Corpus SVEZ-IJS Erjavec et al. (2006), the European legislation corpus in Slovene and English language. This is the biggest multilingual corpus with Slovene language. It contains around 270.000 sentences; sentences are

badly formed with lots of enumeration. It was used as the training data for the word alignment model.

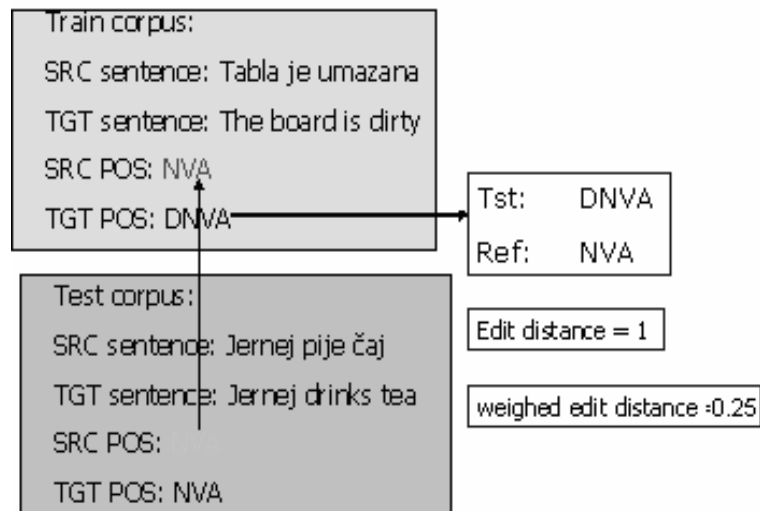


Figure 6: POS strings are gathered as shown by the arrows.

5.2 Dataset

The ten-fold cross-validation Kohavi (1995) was used as the method for estimating the generalization error as it is most suitable for small data sets. The evaluated values in each fold and the average final values are presented.

The corpus used Erjavec et al. (2003) was already POS tagged, so there was no need to use a POS tagger. The POS tags are extracted from the corpus. A part of the corpus, only sentences that were 14 words or less long, was used in the evaluation process due to time complexity of the parsing algorithms.

Each testing subset divides the corpus into testing and training data. Source language POS symbol words (SRC) were used as input translation data. Target language POS symbol words (REF) were used as the reference values in the evaluation process. The output of the system, the target POS symbol words (TST), are compared to the SRC and the REF values.

6 Results

6.1 Translated POS string quality

The quality of the found POS strings, the translation candidates, was evaluated using the Levenshtein-edit distance Levenshtein (1965) and the weighted Levenshtein distance Fu (1982). The weighted edit distance takes into consideration the length of compared strings and weighs the distance accordingly. Each POS string represents the leaves of a target parse tree and consequently the basis for the final translation.

The edit distance between the test POS string and the reference POS string shows how much the output of the testing system differs from a product made by a professional human translator. Smaller values indicate better results.

Figure 6 shows an example of a POS string comparison procedure.

The edit distance between a source POS symbol string and a reference POS symbol string shows how much the input (the source language) of the testing system differs from the product made by a professional human translator. These values were computed to test if the test output POS string is less distant from reference translation than the source strings. This would mean that the presented method produces better translation candidates than the original (source language) strings.

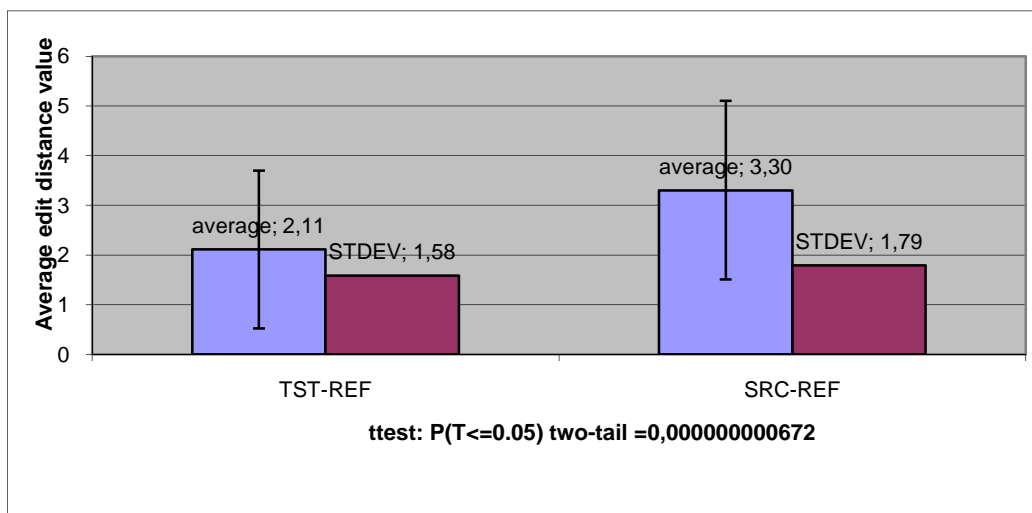


Figure 7: Quality of found POS, only POS with the edit distance=0, ttest shows significant difference between two average values.

Two groups of tests were done for:

- Quality of target POS strings returned by a search with null edit distance. The results are shown in Figure 7.
- Quality of target POS strings returned by a search with edit distance less than 3. The results are shown in Figure 8.

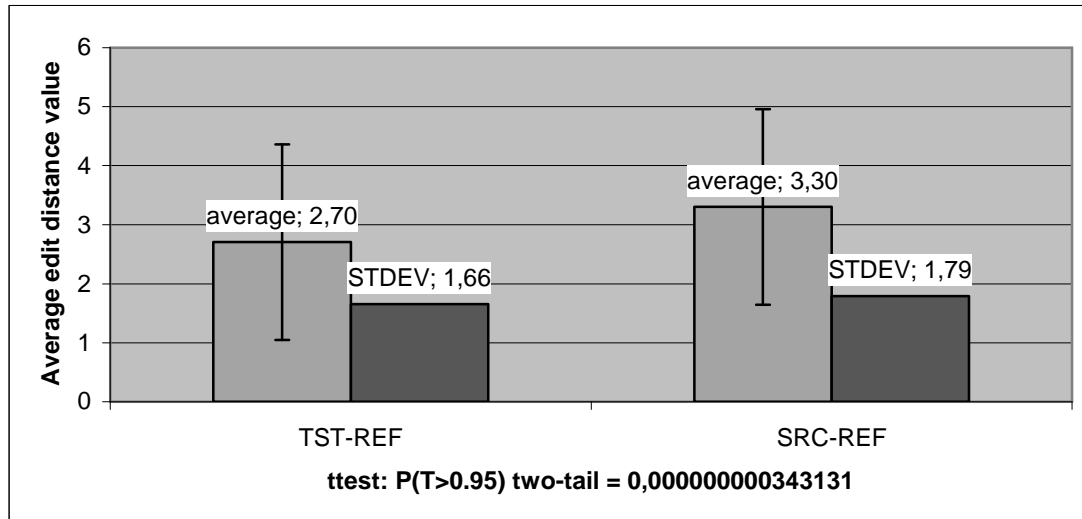


Figure 8: Quality of found POS, POS with the edit distance ≤ 2 , ttest shows significant difference between two average values.

T-tests in Figure 7 and Figure 8 show that the average edit distance between the test POS strings and the reference POS strings is significantly lower than the edit distance between the reference POS strings and the source POS strings. This means that there was a significant information gain using the method presented here.

6.2 Success rate of POS symbol word search

The second problem addressed in the empirical testing was to evaluate the success rate of the POS symbol word search; how many translation candidates, target POS symbol words, are actually found by the algorithm. If the POS symbol word produced from the input sentence, the sentence to be translated, is not found in the source part of the training corpus, then no translation candidate is available and the translation process stops; producing no translation product. This problem is addressed by an extended search that returns the POS symbol words whose edit distance is lower than a predefined threshold.

Table 1 shows the proportion of the input sentences, the test sentences, which have at least one translation candidate. The percentage increases by a big margin using threshold 2 for the edit distance.

Table 1: The proportion of the test sentences that have at least one translation candidate.

| Trainset = 1600 Testset 170 | | | |
|----------------------------------------------|---------------|---------------|---------------|
| | test 1 | test 2 | test 3 |
| Full (edit distance = 0) | 40% | 37% | 39% |
| Edit distance = 1 | 43% | 42% | 44% |
| Edit distance = 2 | 67% | 69% | 70% |
| Edit distance = 3 | 89% | 91% | 89% |
| Edit distance = 4 | 95% | 97% | 97% |
| Edit distance = 5 | 100% | 100% | 100% |

Table 2 shows how the percentage of the found POS symbol words increases with the increasing edit distance threshold. Increasing the edit distance threshold unfortunately decreases the POS symbol word quality and consecutively the translation quality. The algorithm with edit distance threshold greater than zero should be used only if no translation candidates are available. The threshold should be increased in minimal steps.

Table 2: Success rate of the POS symbol word search.

| algorithm\fold | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | AVERAGE | STDEV |
|----------------|------|------|------|------|------|------|------|------|------|------|---------------|-------------|
| Edit dist.=0 | 72 | 60 | 76 | 70 | 82 | 68 | 79 | 96 | 71 | 80 | 75,40 | 9,74 |
| average ed=0 | 0,42 | 0,35 | 0,45 | 0,41 | 0,48 | 0,40 | 0,46 | 0,56 | 0,42 | 0,47 | 0,44 | 0,06 |
| Edit dist.=2 | 128 | 114 | 126 | 130 | 135 | 127 | 137 | 140 | 129 | 132 | 129,80 | 7,18 |
| average ed=2 | 0,75 | 0,67 | 0,74 | 0,76 | 0,79 | 0,75 | 0,81 | 0,82 | 0,76 | 0,78 | 0,76 | 0,04 |

6.3 The impact of the size of training set on success rate of POS symbol word search

We evaluated our method on a small part of the corpus Erjavec (2003) of just 1700 bilingual sentence pairs, due to time and resources deficiency.

The evaluations on the impact of size of training set to the success rate of the POS symbol word search were made.

The size of the training set was gradually increased in chunks of 10% to reach full corpus size. The same testing examples were evaluated on each translation system and the results were compared.

Figure 9 shows the increase in percentage of the successful POS symbol word searches in the relation to the growth of the training corpus size. The bigger corpus means better results although the slope of functions tends to level relatively quickly, meaning that the systems trained on moderately sized corpora should perform almost as well as the systems trained on bigger corpora. Two functions are shown: one for a system that searches only the exact POS symbol word matches and one for the POS symbol word matches that have the edit distance value of two or less compared to the translated POS symbol word string.

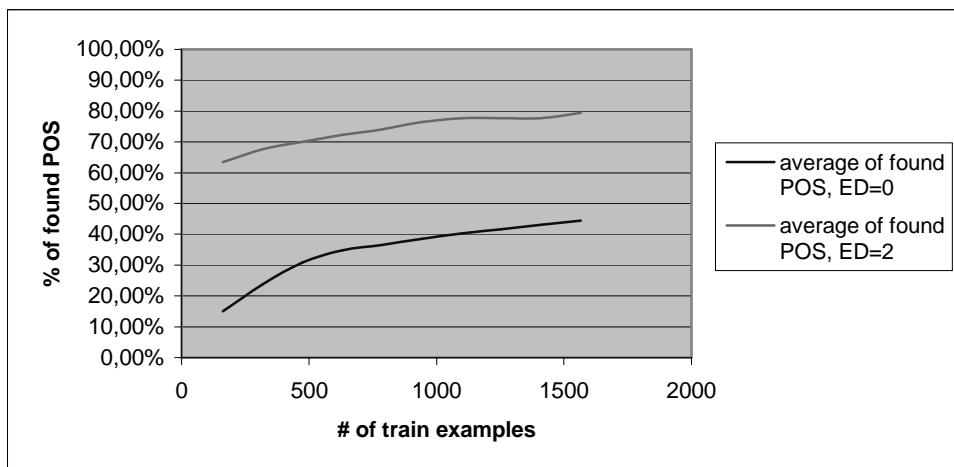


Figure 9: The impact of the size of train set on success rate of the POS symbol word search.

Figure 10 shows the gradient of functions from Figure 9. It shows that the derivation drops to a very low level with a moderate number of the training examples, meaning that further changes will be moderated.

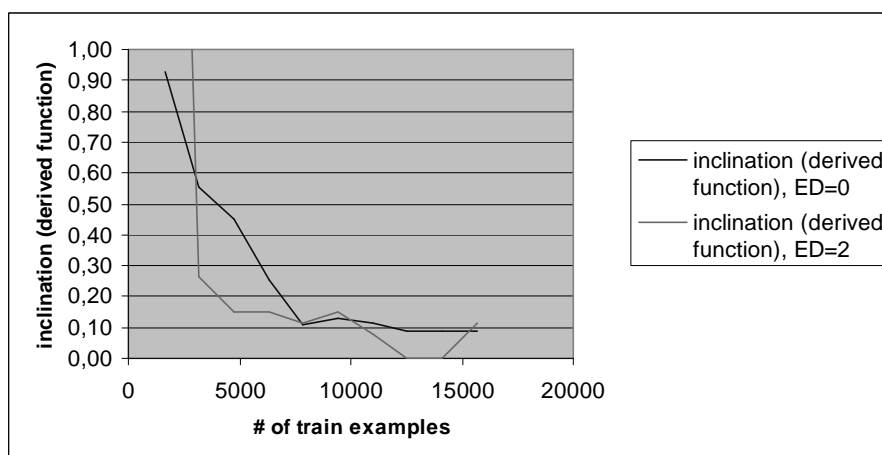


Figure 10: Results – Same results as in Figure 9, but the function is derived, showing how fast the function changes.

7 Conclusions and open issues

The method was tested on a relatively small corpus of only 1700 sentences, but results confirm the applicability of the method. The target sentence POS string quality evaluation (using edit-distance) shows a statistically significant difference in the original values and values gained using the presented method.

The search for the POS symbol string can return no result. The share of such cases is still very high. Using the edit distance (particularly 2 or less) enhances the success rate but decreases target POS string quality.

A bigger training corpus gives better success rate, but the gradient of the function drops at relatively low values suggesting that substantially better results can be obtained using moderate sized corpora but the threshold for a corpus size that gives enough information for the training translation system is still to be defined.

Evaluation of the system with a corpus that is composed of bigger sentences as should be repeated in order to clearly show the value of presented results as the evaluation was performed on relatively small sentences.

Last step of the translation part of the method, the inclusion of the word alignment model is still to be implemented.

References

- [1] Al-Onaizan, Y. (1999): Statistical machine translation, Final report, JHU workshop 1999. Technical report, The Center for Language and Speech Processing, The Johns Hopkins University.
- [2] Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L. (1993): The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19**, 163-311.
- [3] Burbank, A., Carpuat, M., Clark, S., Dreyer, M., Fox, P., Groves, D., Hall, K., Hearne, M., Melamed, D., Shen, Y., Way, A., Wellington, B., and Wu, D. (2005): *Final Report of the 2005 Language Engineering Workshop on Statistical Machine Translation by Parsing*. <http://www.clsp.jhu.edu/ws2005/groups/statistical/documents/finalreport.pdf>
- [4] Charniak, E. (2000): A maximum-entropy-inspired parser. *Proceedings of NAACL-2000*.
- [5] Collins, M. (2003): *Head-Driven Statistical Models for Natural Language Parsing*. Computational Linguistics, MIT Press Journals.
- [6] Clarkson, P.R., Rosenfeld, R. (1997): Statistical language modeling using the CMU-Cambridge toolkit. *Proceedings of ESCA Eurospeech 1997*.
- [7] EAMT (2008): What is Machine Translation. <http://www.eamt.org/mt.html>.

- [8] Erjavec, T., Gorjanc, V., and Stabej, M. (1998): Korpus FIDA, In *Proceedings of International Multi-Conference Information Society - IS'98*, Ljubljana, Slovenia.
- [9] Erjavec, T., Džeroski, S., and Zavrel, J. (2000): Morphosyntactic tagging of Slovene: Evaluating PoS taggers and tagsets. Second International Conference on Language Resources and Evaluation, LREC'00, 1099-1104.
- [10] Erjavec, T., Krstev, C., Petkevič, V., Simov, K., Tadić, M., and Vitas, D. (2003): The MULTEXT-east morphosyntactic specifications for slavic languages. *Proceedings of the EACL 2003 Workshop on Morphological Processing of Slavic Languages*, 25-32, Budapest.
- [11] Erjavec, T. (2006): The English-Slovene ACQUIS corpus. In *Proceedings of Fifth International Conference on Language Resources and Evaluation*, LREC'06, May 24-26 2006. Genoa.
- [12] Fu, King Sun (1982): *Syntactic Pattern Recognition and Applications*. Prentice Hall, 1982.
- [13] Kohavi, R. (1995): A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1137-1143. San Mateo, CA: Morgan Kaufmann,.
- [14] Ledinek, N. and Žele, A. (2005): Building of the Slovene dependency treebank corpus according to the Prague dependency treebank corpus. In *Proceedings of Grammar and Corpus*, Prague.
- [15] Levenshtein, V. (1965): Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk, SSSR*, **163**, 845–848.
- [16] Melamed, I.D. (2004): Statistical machine translation by parsing. *Proceedings of ACL-04*, Barcelona, Spain.
- [17] Och, F.J., Hermann, N. (2003): A systematic comparison of various statistical alignment models. *Computational Linguistics*, **1**, 19-51.
- [18] Papineni, K., Salim, R., Todd, W., and Wei-Jing, Z., (2001): Bleu: a method for automatic evaluation of machine translation. *Technical Report*, RC22176, IBM, 2001.
- [19] Toutanova, K., Tolga Ilhan, H., and Manning, C.D. (2002): Extensions to hmm-based statistical word alignment models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA.
- [20] Santorini, B., Marcinkiewicz, M.A., Marcus, M.P. (1993): *Building a Large Annotated Corpus of English: The Penn Treebank Computational Linguistics*, **19**, 1993.
- [21] Ueffing, N. and Ney, H. (2003): *Using POS Information for Statistical Machine Translation into Morphologically Rich Languages*, EACL2003, Budapest, Hungary.