Investigating the Possibilities of Using SMT for Text Annotation

László J. Laki[1]

1 **MTA-PPKE Language Technology Research Group –**
Pázmány Péter Catholic University, Faculty of Information Technology,
50/a Práter street, Budapest, 1083, Hungary
laki.laszlo@itk.ppke.hu

Abstract

In this paper I examine the applicability of SMT methodology for part-of-speech disambiguation and lemmatization in Hungarian. After the baseline system was created, different methods and possibilities were used to improve the efficiency of the system. I also applied some methods to decrease the size of the target dictionary and to find a proper solution to handle out-of-vocabulary words. The results show that such a light-weight system performs comparable results to other state-of-the-art systems.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases SMT, POS-tagging, Lemmatization, Target language set, OOV

Digital Object Identifier 10.4230/OASIcs.SLATE.2012.267

1 Introduction

A wide spectrum of opportunities has been opened due to the fast development of information technology in almost all disciplines. This evolution could be detected on the field of computational linguistics as well. Processing of huge text materials has become easier, even the efficiency of these systems is increasing. Marking texts with syntactic and/or semantic information, or the morphological analysis of the language are really important tasks for computational linguistics. The task of part-of-speech (POS) tagging has not yet been perfectly solved, even though several systems have been implemented to achieve better results to this complex problem. The most popular ones are based on machine learning, in which the rules recognized by the systems themselves are based on different linguistic features. Further difficulties lie in determining the features, since these could be hardly formulated. Instead statistical machine translation (SMT) systems are able to recognize essential translation rules and features without any previous linguistic knowledge [8].

Based on this assumption the application of SMT systems for text analysis could be successful. With the help of the standard frameworks and tools [11, 10, 15] used for statistical machnie translation tasks, it is straightforward to handle complex POS structures. In this work I examine the applicability of these systems to solve the task of part-of-speech disambiguation and lemmatization.

2 Basic Concepts

2.1 Statistical Machine Translation

Statistical machine translation (SMT) is a method of statistical language processing usually applied to translation between human languages [12]. It has a great advantage over rule-

1st Symposium on Languages, Applications and Technologies (SLATE'12).
Editors: Alberto Simões, Ricardo Queirós, Daniela da Cruz; pp. 267–283
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

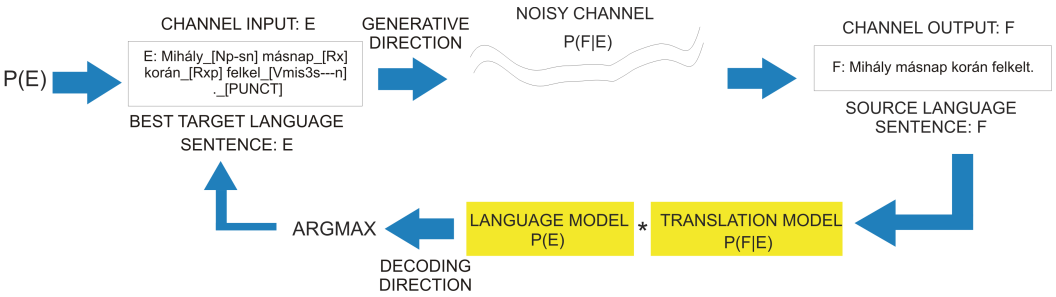

Figure 1 Representation of SMT method.

based translation: only a bilingual corpus is needed to set up the training set of the system, but knowledge about the grammar of the language is not required to create the architecture of a baseline SMT system. The system is trained on this corpus, from which statistical observations and rules are determined.

The phrase, which we want to translate – i.e. the source sentence, is the only certain thing we know prior to the translation. Therefore, the system is defined as a noisy channel [8]. A set of target sentences are passed through this channel and the output of the channel is compared with the source sentence.

This process can be formulated by Bayes' theorem as the product of two stochastic variables called language model and translation model. The result of the translation is the phrase, which provides the most appropriate match with the source sentence. In addition this match is a probability that could be determined from the language model $p(E)$ and the translation model $p(F|E)$ according to the following formula [12, 8]:

$$\hat{E} = \underset{E}{argmax}\ p(E|F) = \underset{E}{argmax}\ p(F|E) * p(E) \tag{1}$$

2.2 Part-of-Speech Disambiguation

POS-tagging is the process of assigning a part-of-speech or other lexical class marker to each word in a corpus. The input to a tagging algorithm is a string of words and a specified tagset. The output is a single best tag for each word [9].

However POS tagging is harder than just having a list of words and their part of speech, because some words can represent more than one part of speech depending on the context. A simple example is the Hungarian word "vár" that has two different meanings – "wait" and "castle" – with different part of speech, i.e. verb and noun.

Most solutions apply analysis of the text based on pre-specified rule systems. The disadvantage of these methods is the huge cost of the creation of rules. Other frequently used approaches are based on machine learning, in which there are also some kind of rules used, however these are not of the same kind as linguistics rules, but are developed by the algorithms themselves based on relevant features. Further difficulties lie in the determining of these features, since these could be hardly formulated. It is very hard to determine and create a complete rule system that covers all the linguistic features and which can be processed by a computer.

2.3 Lemmatization

In computational linguistics, lemmatization is the algorithmic process of determining the lemma for a given word. The lemma is the dictionary form of a word. Since the process may involve complex tasks such as understanding context and determining the part of speech of a word in a sentence (requiring, for example knowledge of the grammar of a language), it is a hard task to implement a lemmatizer for a language like Hungarian, because words appear in several inflected forms.

Several implementations exist to solve this problem (e.g. HUMOR [18]), but most of them are based on complex methods of preprocessing that separate this task from that of POS-tagging, though these are very strongly related.

3 POS-tagging as SMT Problem

As described above both POS-tagging and lemmatization could involve huge amomunt of resources and complexity especially when applied to more complex languages, like Hungarian. In English a word can only have a limited number of forms, however in agglitunating languages this number is several orders of magnitude higher. Each affixum of a word contains some morphological information and also might produce a change in the lemma of the word. Therefore even more sophisticated algorithms are necessary to handle such a behaviour properly. These considerations deduce the application of the methods of statistical machnie translation to POS-tagging. In such a case POS-tagging is considered as a translation between sentences (F) and their tagged versions ($\hat{E}$) [12, 8]:

$$\hat{E} = \underset{E}{argmax}\ p(E|F) = \underset{E}{argmax}\ p(F|E) * p(E) \tag{2}$$

In equation $p(E)$ is the language model of the POS tags and $p(F|E)$ is the translation/analysis model. The source sentence is a set of phrases that are to be translated to tags. POS-tagging is a simpler task for an SMT system than the translation between natural languages, since the change of word order in a sentence is not required. The number of elements in the source and target side is equal; the system does not make item insertion or deletion [14, 5]. That is why an SMT system might be applied successfully to solve the task of POS-tagging.

Even though POS-tagging would not require to use the sophisticated tools of an SMT framework, these might have an effect on handling the deeper structure of the sentence or the dependency and the context of the words that is to be annotated.

Handling and analyzing out-of-vocabulary words that are not included in the training set (OOV words) has a significant influence on the success of a POS-tagging system. The type frequency of OOV words might vary in different languages. In English an OOV word will probably be a proper noun. In some other languages – such as Hungarian – OOV words would equally be nouns or verbs as well. This is due to the practically infinite number of word forms that might appear, thus it is impossible to have a corpus containing all forms of each word.

The benefit of this method is that the system is able to find rules without defining feature sets and it could do POS-tagging and lemmatization simultaneously. Another advantage is that it is a language independent method, where the the performance of the system only depends on the quality of the bilingual corpus used to train the system. Though my purpose was only to test the system on Hungarian, in later works it might be extended to other languages easily.

3.1 Coding Systems of POS Tags

There are three types of coding systems used for morphological coding in Hungarian language; namely KR [13, 4] , HUMOR [18] and MSD [4] systems. The morphology of Hungarian language was taken into consideration, when KR-coding system was developed. It is basic syntax however is language-independent. The HUMOR morphological coding system [18] is based on unification. Different labels are used as tags, based on the capability of fusing morphemes with others. Different labels could allow or contradict each other. One word can be built up from morphemes, for which the labels do not exclude each other. The MSD-coding system [4] was used to analyze the corpus from morph syntactical point of view. The MSD-coding system is applied for coding different attributes of words – mainly morphologically –, and could be used for most European languages. The morph syntactical attributes of words – for example type, mood, tense, number, person etc. – are represented as a character set. In the place of attributes, which are missing or not interpreted in natural languages, character '-' is used. At the first position the main POS categories of words are available. In this work I use MSD coding only for the reason that the only available tagged corpus of Hungarian is provided with MSD codes.

3.2 Framework

3.2.1 Corpus

In this study the Szeged Corpus 2 [3] was used as parallel corpus, which was created by the Language Technology Group of the University of Szeged. This XML-based database contains both plain texts and their POS annotated version using the MSD-coding system. The advantage of the corpus is that it was manually corrected; therefore it is a highly accurate data set. Further benefit is that it is general and not topic-specific. In order to have such a reliability, it is a rather small corpus containing 1.2 million words, which cover 155.500 different word-forms and 250.000 inter-punctuation signs. In contrast to natural language translation, where this size is unusably small, it is not such a relevant problem for POS-tagging, since the target language has a very limited vocabulary compared to any natural languages. For testing the system, 1500 randomly selected sentences of the corpus were used.

3.2.2 Training and Decoding

Several methods of obtaining information from parallel corpora have been studied. Finally, I decided to use IBM models, which are relatively accurate, and the used algorithm was adaptable to the task. Based on these findings I decided to use the MOSES framework [11, 10], which implements the above mentioned IBM models. This system includes algorithms for the pre-processing of the parallel corpus, for the setup of translation and language models and for the decoding and the optimization to the BLEU score [17]. An improved SMT system framework is JOSHUA [15], which not only applies word- or phrase-level statistics, but takes into account the morphological characteristics of the language. Chomsky's generative grammars are used to solve this task. The languages, which could be described with grammatical rules, belong to the class of regular languages and context-free grammars (CFG). The advantage of the JOSHUA system is that it is able to translate between these CFG rules in such a way, that rules can be specified for both source and target languages, furthermore the probability of the transformations into each other.

3.2.3 Evaluation

To evaluate the efficiency of traditional SMT systems an automatic method is used. The BiLingual Evaluation Understudy [17] – BLEU score. The essence of this method is that the translations are compared with the reference sentences of the test set. BLEU score is calculated both to each n-gram lengths, and to a comulated average as well. Since POS-tagging is a one-to-one mapping between tags and words the most relevant measure gains from the case of 1-grams. Since BLEU score is not the usual method of evaluating a POS-tagger and lemmatizer, I also calculated the accuracy of the system to be able to compare the efficiency to other systems. This evaluation was used in sentence and in token level as well.

4 Baseline System

In the following sections I describe each versions of the system and their results.

In the first test the system was trained with unmodified corpus. The source language corpus was created from the tokenized sentences without annotation. The target language corpus contained the lemmatized words and their POS tags. Table 1 displays the results of the system (SMT_Zero) for each decoder.

Table 1 Performance of the system SMT_Zero.

System	BLEU score	Precision
MOSES	98.35%	90.29%
JOSHUA	97.28%	91.02%

The relatively low results revelad some drawbacks of the applied method. The most relevant problem arises from the structure of the corpus. In the annotated corpus the lemma of each word is connected to the morphological tags. In the case of multi-word phrases (for example: multi-word proper names, verb phrases) the tag either joins to the last word of each phrase or stands after the last word. The lack of the marks of related phrases makes false probability values in the translation model. Consequently the system assigns a random tag after proper names which made the results even worse.

4.1 Elimination of Single POS Tags

To solve the problem of missing or unjoined tags, all independent tags were joint to the previous word (system SMT_NoSinglePOS. The average results show a slight improvement as displayed in table 2.

Table 2 Performance of the system SMT_NoSinglePOS.

System	BLEU score	Accuracy
MOSES	98.40%	90.80%
JOSHUA	97.25%	90.72%

Though the change in the BLEU score is not significant, the accuracy of the system is increased with 0.5-0.6 percent (in the case of MOSES, which proved to be more efficient

than JOSHUA for all the later methods as well). This is due to the fact, that unnecessary elements are not included in the translation.

4.2 Handling of Multiple-word Phrases

The toughness of this task is that the system analyses only words, therefore each part of a phrase is tagged separately. The goal is to handle these multi-word phrases as one unit. Most of these phrases are recognized as named entities. Thus the system was improved by joining multi-word phrases in the source side corpus. Table 3 displays the result of this system called SMT_Baseline1.

Table 3 Performance of the system SMT_Baseline1.

System	BLEU score	Accuracy
MOSES	98.49%	91.29%
JOSHUA	97.31%	91.07%

Numerically from the 1500 sentences of the test set 506 were absolutely correct and 994 sentences had mistakes. At first sight this is a quite strange rate, but if we see the result at token level (24557 correct and 2343 incorrect) we got much better evaluation. Table 3 shows that joining related words increased the accuracy of the system; however the BLEU score was lower than the result of the previous system.

The evaluation revealed that the wrongly annotated sentences could be divided into two categories. The first is when the system does not perform the translation, but returns the original word (1697 pieces). In most cases these words are not included in the corpus, so they could not be in the translation model. If the decoder does not find an entry in the translation model, it keeps the original form of the word in the translation, in my case that is the word form instead of a POS-tag. The other type of error is the case of incorrect annotations (646 pieces). Two subcategories can be distinguished in this case. The first is when the system can find correctly the main POS tag of the word but it fails in the further analysis; secondly when even the main POS tag is incorrect. Table 4 shows an example of the output of the system SMT_Baseline1.

Table 4 An example from the output of the system SMT_Baseline1.

System	Translation
Simple text:	ezt a lobbyerőt és képességet a diplomáciai erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	ez_[pd3-sa] a_[tf] lobbyerőt_[x] és_[ccsw] képesség_[nc-sa] a_[tf] diplomáciai_[afp-sn] erőfeszítés_[nc-pp] kívül_[st] mindenekelőtt_[rx] a_[tf] magyarországi_[afp-sn] multinacionális_[afp-pn] adhat_[vmcp3p—y] ._[punct]
SMT annotation:	ez_[pd3-sa] a_[tf] lobbyerőt és_[ccsw] képesség_[nc-sa] a_[tf] diplomáciai_[afp-sn] erőfeszítéseken kívül_[st] mindenekelőtt_[rx] a_[tf] magyarországi_[afp-sn] multinacionális_[afp-pn] adhat_[vmcp3p—y] ._[punct]

This system is considered as an SMT_Baseline1 system (the traditional baseline – i.e. rendering the most probable tag – for POS-tagging is used in later sections of the paper).

5 Decreasing the Size of Target Vocabulary

5.1 With only POS Disambiguation

An essential part of any statistical methods is the number of training instances, in this case the size of the corpus. Since I used the biggest available tagged corpus for Hungarian, there is no way to achieve better results with increasing the size of the training set. However another possibility is to decrease the vocabulary in order to have a relatively bigger training set with decreasing the complexity of the annotation task. One way to achieve this is if the simple text is translated to the "language" of the POS tags only without performing lemmatization. The size of the target dictionary was reduced from 152 694 to 1128 elements. This number of tags is much fewer than the number of Hungarian words; therefore a relatively accurate system could be built from a smaller corpus. On the other hand, if the lemmas are left out from the annotation and the translation is made only to the set of tags, the order of the morphemes in the sentence will be much more weighted in the translation and language models as well. The result of this system (called SMT_OnlyPOS later SMT_Baseline2) is displayed in table 5.

Table 5 Performance of the system SMT_OnlyPOS.

System	BLEU score	Accuracy
MOSES	96.22%	91.46%
JOSHUA	92.17%	91.09%

The system achieved worse BLEU score compared to the SMT_Baseline1 system, but the accuracy is better. Numerically there are 518 correct sentences and 982 incorrect ones that means 0.8% improvement compared to the SMT_Baseline1 system. Regarding the tokens, 24603 correct and 2297 incorrect ones were counted; that is 0.17% improvement. The number of not annotated words (1699 item) did not change; however the number of incorrect POS tags appeared only in 598 cases.

Thus the main improvement of the quality at this stage results from decreasing the number of incorrect POS tags. Deeper evaluations prove that besides this improvement (48 items), in some cases the previously correct tags failed. These failures were caused mainly by mixing adverbs with conjunctions or conjunctions with demonstrative pronouns. Output of system SMT_OnlyPOS showed in table 6.

5.2 With Simplifying POS Tags

Another method to decreace the size of the target language is to simplify the resulting POS-tags to include less morphological information. This method reduces the complexity of the system, but consequently the depth of analysis will be decreased as well. Only the main POS tags – the first characters of MSD codes – were used. This way the target dictionary consists of only 14 elements. The result of the system (called SMT_MainPOS) is displayed in table 7.

Table 6 An example from the output of the system SMT_OnlyPOS.

System	Translation
Simple text:	ezt a lobbyerőt és képességet a diplomáciai erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] lobbyerőt [ccsw] [nc-sa] [tf] [afp-sn] erőfeszítéseken [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]

Table 7 Performance of the system SMT_MainPOS.

System	BLEU score	Accuracy
MOSES	90.35%	92.20%

The evaluation results fit to the previously seen tendency; i.e. there is a decrease in BLEU score, but the accuracy of the system increased. 553 sentences were correct and 947 incorrect; it means 2.3% improvement compared to the system SMT_OnlyPOS and 3.1% to the SMT_Baseline1 system. 24803 tokens were correctly tagged and 2097 incorrectly; this is 0.77% improvement to SMT_OnlyPOS and 0.84% compared to the SMT_Baseline1 system. A sample from the output is shown in table 8.

Table 8 An example from the output of the system SMT_MainPOS.

System	Translation
Simple text:	ezt a lobbyerőt és képességet a diplomáciai erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	p t x c n t a n s r t a a v p
SMT annotation:	p t lobbyerőt c n t a erőfeszítéseken s r t a a v p

5.3 Conclusion

The above results are very promising, as the accuracy of the system is over 90% even if it was trained on a small-sized corpus. We have to note, however that the size of the dictionary of system SMT_OnlyPOS (1128 tags) is much smaller compared to that of the SMT_Baseline1 (152 694 tags), but the accuracy increased only with 0.17%. Furthermore in system SMT_MainPOS where 14 tags were used, the accuracy increased only with 0.88%. This 0.88% increase is not proportional to the significant information loss that is caused by the size minimization of the dictionary in system SMT_MainPOS. Furthermore despite the positive changes in the results, the above systems are still not able to tag OOV words (1698 cases). Consequently the next step should be to find a proper solution to handle words not included in the training set.

6 Handling OOV Words

The most obvious solution to reduce the number of OOV words is to increase the size of the corpus, so that all word forms would appear in it. Moreover it is important to have several occurences of each token in order to have a reliable statistics. Due to the agglutinative nature of Hungarian language, one stem could have many forms caused by the affixes; that is why an extremely large corpus would be needed to have all forms with the appropriate weight. This is an impossible requirement by itself, even more if a manually tagged corpus is expected. To eliminate this situation, I applied a method in which the system tries to find the appropriate tag for an unknown word based on the analyses of its context. In this capture the lemmatization is left out. All results will be compared with system SMT_OnlyPOS (from now SMT_Baseline2).

6.1 In the Original Text

To examine the characteristics of frequent OOV words, a further investigation is needed during training and decoding. My basic assumption is to infer OOV POS-tags from the context. Though this is quite a simple method, however the complexity of the problem can also be reduced by limiting the possible POS-tags of an unknown word to some of the most probable ones. Thus at decoding time, the system has to choose only from these few tags.

To eliminate this problem I applied Guillem and Joan Andreu's method [5]. To achieve good results for Hungarian I used their results for English with some changes. A dictionary is created from words whose frequency in the training set is over a certain threshold value. The word frequency is calculated from the corpus. The words not included in this dictionary are changed to an optional expression (in this case "UNK"). The basic idea of the method is to change the less frequent words to the string "UNK".

Since OOV words are included in just a few word classes, therefore I assume that the annotation of the context of each OOV word is very similar. The SMT system performs the translation based on phrases, therefore the context of words and tags is taken into consideration already. By replacing the less frequent words to symbol "UNK", the annotation of the environment of these phrases will be more significant. Consequently the system can identify the POS tag for symbol "UNK".

The key question is the appropriate threshold value selection, since it determines the number of "UNK" symbols in the corpus. On one hand if this value is too high, too many tokens will be changed to "UNK" symbol; the probability of this symbol increases, therefore we will not receive correct annotation. On the other hand if the threshold is too small, too many rare words will be included in the dictionary causing that the advantage of the method could not be exploited sufficiently.

Therefore the system was trained with more threshold values (2, 4, 6, 8 and 10) to find the most appropriate one resulting in the best improvement of accuracy. The results of this system (SMT_OOV_token) can be found in table 9.

If the threshold is 1 the table gives us the result of SMT_Baseline2 system. That means none of the words were replaced with symbol "UNK". In the last column we can see the accuracy of the systems for each threshold value. For example: threshold 2 means that all words that appear in the corpus less than two times were changed to symbol "UNK". The second column of the table shows the percentage of words in the training set (of size 1 459 288) added to the dictionary. For example: in the case of threshold 2 almost 60% of the words became OOV words. The third column of the table contains the percentage of the words left original in the corpus.

Table 9 Performance of the system SMT_OOV_token.

Threshold value	Rate of words in the dictionary	Rate of words in the corpus	System's accuracy
SMT_Baseline2	100%	100%	91.46%
2	39.16%	93.87%	93.13%
4	19.18%	89.22%	90.40%
6	12.99%	86.48%	88.41%
8	9.96%	84.51%	87.07%
10	8.09%	82.92%	85.97%

From the above results it is straightforward that in the case of threshold value 2 the system achieved significant improvement compared to any of the previous ones. Only 38 words were not annotated against the 1697 in system SMT_Baseline1. If the threshold value is raised, it leads to a decrease in the accuracy of the system.

During deeper evaluation it turned out that this accuracy decrease is due to the lower rate of original words in the corpus (only small number of words are in the dictionary). In the case of threshold 2, symbol "UNK" was used for 6.13% of the words in the training set. This rate is 92% in the case of threshold being 10. This tendency matches with the theorem of Zipf's laws [21]. We have to note the 85.96% accuracy at threshold value 10. This result is quite good despite that only 8.09% of the training set was added to the dictionary. Table 10 shows an example from the output of the system SMT_OOV_token in the case of threshold 8.

Table 10 An example from the output of the system SMT_OOV_token.

System	Translation
Simple text:	ezt a **unk** és **unk** a diplomáciai **unk** kívül mindenekelőtt a magyarországi **unk unk** .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] [nc-sa] [ccsp] [vmis3p—y] [tf] [afp-sn] [nc-pn] [st] [rx] [tf] [afp-sn] [nc-pn] [nc-sa—s3] [punct]

6.2 In Case of Lemmas

From the results of table 10 it can be seen that if the threshold value is too high, too many of the words become "UNK". Due to the agglutinative features of Hungarian language the original text contains different forms of nouns, verbs and adjectives of the same stem. This is the reason that the number of these different forms is under the threshold. Consequently in most cases nouns, verbs and adjectives are also replaced with "UNK" in the sentences of the corpus, which makes the decreases the accuracy of the system. My goal was to reduce the number of symbol "UNK" with replacing only really rare words in the text. Therefore the threshold was determined based on the frequencies of the lemmas and not on different word forms. The results of this system (called SMT_OOV_lemma) can be found in table 11.

Table 11 Performance of the system SMT_OOV_lemma.

Threshold value	Rate of words in the dictionary	Rate of words in the corpus	System's accuracy
SMT_Baseline2	100%	100%	91.46%
2	70.00%	96.58%	92.57%
4	56.64%	94.50%	92.25%
6	50.18%	93.17%	91.81%
8	45.81%	92.13%	91.48%
10	37.08%	88.47%	91.10%

The results proved that calculating the threshold based on lemmas makes much fewer number of words to be marked as OOV (numerically only 3.42% of the words from the corpus).

Table 12 shows an example from the output of the system SMT_OOV_lemma in the case of threshold 8 similar to table 10. We can see that in this case only two words were replaced to "UNK" against the previous systems so the goal of reducing the number of OOV words was achieved.

We can observe that besides threshold 2, the best result of system SMT_OOV_lemma (92.57%) is worse than in the case of SMT_OOV_token (93.13%). The deep evaluation showed that the number of not annotated words increased (1015 cases) compared to system SMT_OOV_token, and 984 words were incorrectly analyzed.

Table 12 An example from the output of the system SMT_OOV_lemma.

System	Translation
Simple text:	ezt a **unk** és képességet a **unk** erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] [nc-sa] [ccsw] [nc-sa] [tf] [afp-sn] erőfeszítéseken [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]

6.3 Multiple Thresholds

The above results have alraedy achieved high accuracy results of taggin Hungarian words, but still OOV words are included to several different POS types with quite high probability. In English such OOV words are mostly nouns. To have a more sophisticated method I applied numerical calculation of several threshold values that distinguish different POS-tags fo OOV words.

We can observe that the same thresholds in the above two systems divide the corpus in different proportions. Word forms with frequency values higher than the threshold are included in the dictionary of system SMT_OOV_token.; but if we determine the frequencies based on lemmas – such as in the case of system SMT_OOV_lemma. – the dictionary will contain more words. Thus a certain threshold divides the set of words to three parts. The first set contains the words, which are included in both dictionaries; these words are the

most relevant ones. In the second set we can find those OOV words, which are really rare and were under threshold in both cases. The words, for which the word form is not frequent enough, but their lemma is over the threshold were included into the third set.

I examined the types of OOV words in each set. The results showed that adjectives and other types of OOV words mostly belong to the second category (under both threshold level), while verbs to the third set. Nouns can be found in both category roughly in a similar measure. Based on this observation another system was trained, which is able to distinguish OOV words, if they belongs to the second or third categories. The results of this system (called SMT_OOV_multi) are shown in table 13.

Table 13 Performance of the system SMT_OOV_multi.

Threshold value	System's accuracy
SMT_Baseline2	91.46%
2	93.28%
4	90.65%
6	88.62%
8	87.40%
10	86.15%

The results reflect that the system with threshold 2 achieved the best performance (93.28%) of all the above systems. This improvement is caused by the fact that only 37 words were not analyzed. Furthermore in the case of incorrect analysis – numerically 1772 items – the error occurred mostly during the subanalysis of nouns.

According to the evaluation, the method of using multiple thresholds helped to distinguish adjectives and verbs; therefore lead to the improvement of the system.

6.4 Introducing Postfixes

Based on the results of the previous systems it is straightforward to conclude that using multiple thresholds – three classes – are not enough to separate nouns, verbs and other types of words. Due to the wide range of affixes in Hungarian, one word could have many forms. Different POS types however have characteristic prefixes and postfixes (in case of Hungarian language mainly postfixes). Therefore previous methods were extended to use information based on the last characters of an OOV word to determine the type.

The best method would be to use a morphological analyzer to separate postfixes of a word with different lengths, but one of the purpose of my method is its simplicity, therefore I applyed a simple implementation for this task as well. To continue the idea of the previous sections in this section the last 2, 3 or 4 characters of the original OOV words were joined to the "UNK" symbol. The results of this system (called SMT_OOV_postfix) are shown in table 14.

This system significantly outperforms any of the previous ones. The worst result is better than the result of the SMT_Baseline2. The best result was 95.96% which was achieved with threshold value 2 and 4-character-long postfixes of OOV words. The optimal length of the postfix might depend on the language, nevertheless in Hungarian most of the postfixes are 2 or 3 character long, that is why the system with 3-character postfixes and with the threshold value of 2 is above 95.83%. The slightly higher results in the case of four characters is due to the cumulative behaviour of suffixes.

Table 14 Performance of the system SMT_OOV_postfix.

Threshold value	System's accuracy Number of left characters 2	3	4
SMT_Baseline2	91.46%	91.46%	91.46%
2	95.17%	95.83%	**95.96**%
4	94.17%	95.32%	95.90%
6	93.48%	94.97%	95.73%
8	92.94%	94.70%	95.60%
10	92.61%	94.55%	95.55%

Table 15 shows an example from the output of the system SMT_OOV_postfix in the case of threshold 8 similar to previous ones. We can see that this system made correct annotations for all words of the sentence in contrast to the above ones.

Table 15 An example from the output of the system SMT_OOV_postfix.

System	Translation
Simple text:	ezt a **unk_erőt** és képességet a **unk_ciai** erőfeszítéseken kívül mindenekelőtt a magyarországi multinacionálisok adhatnák .
Reference annotation:	[pd3-sa] [tf] [x] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]
SMT annotation:	[pd3-sa] [tf] [nc-sa] [ccsw] [nc-sa] [tf] [afp-sn] [nc-pp] [st] [rx] [tf] [afp-sn] [afp-pn] [vmcp3p—y] [punct]

7 Evaluation and Comparison with Other Systems

There are several freely available part-of-speech taggers, that are used for Hungarian. First I am going to introduce the available tools, then comparing them with the my methods detailed above.

The most commonly known and used tool is HunPos [7, 6] which is an open source Hidden Markov model based disambiguator tool. It is a reimplementation of Brants' TnT [2] system. While TnT is only capable of generating a smoothed bi- uni- trigram contextual model and a unigram lexical model, HunPos generates a smoothed n-gram contextual model and a context sensitive lexical model. Both of them employs a trie based suffix guesser for determining the correct tags for unknown words, and a special lexical model for handling cardinals. Another enhancement of HunPos over TnT is that it is able to utilize a morphological table (MT[1]). HunPos uses the MT for reducing the search space of the decoding algorithm which may increase heavily for unknown words, enabling it to achieve a significantly better accuracy.

PurePos [16] is an open source tool based on HunPos and TnT. In its implementation it keeps the enhancements intruduced by HunPos, and mainly contributes by performing

[1] A MT is a list of words and their possible morphological labels.

a full morphological disambiguation[2]. It has an interface for employing a morphological analyzer, that is used for determining the correct lemma candidates and can also increase its part-of-speech tagging accuracy.

Another well-known system is the OpenNLP toolkit [1] which is an open source natural language processing tool, including a maximum entropy and perceptron based POS-tagger as well. The basis of this tool is the method developed by Ratnaparkhi [19] , that employs context sensitive features in the case of frequent words, and lexical features[3] for rare words (used for handling unknown words).

Besides OpenNLP there are many other tools applying the maximum entropy approach, most of them are able to perform almost state-of-the-art accuracy for several languages. One of them is the Stanford Log-linear Part-of-speech Tagger [20], that uses a dependency network representation in the log-linear framework. Magyarlanc [22] is an NLP toolkit that was developed for IR systems. It contains an adaptation of the Stanford tagger for Hungarian, a tokenizer and a lemmatizer as well. Unfortunately this system is not directly comparable with the others above since it uses its own tagset[4], that is generated by the integrated analyzer.

The previously detailed SMT based approaches are compared with a baseline and several state-of-the-art systems for Hungarian. For the comparison I used the following supervised learning based method: 1) the training algorithm registers the tags and their frequencies for each seen token 2) the tagger assigns the most frequently seen label for a previously seen token 3) in the case of unknown words it assigns the globally most frequently seen tag.

The evaluation was done on the same test set (a portion of the above described Szeged Corpus [3]) for each system.

Table 16 Comparison of part-of-speech tagging accuracy.

System	Token accuracy	Sentence accuracy
Baseline (BL)	89.66%	25.27%
SMT_Baselin2	91.46%	34.53%
SMT_OOV_postfix	95.96%	56.47%
PurePos	96.03%	55.87%
PurePos-MorphTable	97.29%	66.40%
OpenNLP Maxent (ONM)	95.28%	26.00%
OpenNLP Perceptron (ONP)	94.98%	26.67%

Table 16 shows the accuracy of the investigated Hungarian tagging methods. HunPos is not included since it produces exactly the same results as PurePos. PurePos-MorphTable and PurePos denotes PurePos with and without the morphological table while BL is for the baseline system. One can notice that it produces the highest accuracy when it employs morphological analyses, but without this extra information the method described in section 6 (SMT-OOV) is very close to the best one. Even more investigating the per sentence accuracy my method performs significantly better than the plain HMM based one. The accuracy of the OpenNLP maxent based (ONM) and perceptron based (ONP) methods are under the

[2] Full morphological disambiguation is the task of correctly identifying both the POS tag and the lemma.
[3] e,g, at most four character long prefixes and suffixes of the word.
[4] magyarlanc uses a reduced set of the MSD codes.

Table 17 Comparison of full morphological disambiguation accuracy.

System	Token accuracy	Sentence accuracy
SMT_Baseline1	91.29%	33.73%
PP	83.92%	10.00%
PP-MT	84.89%	11.60%

expectations. It is partly because they use features that are developed mainly for English and are not customized for Hungarian.

From Table 17 the advantage of the SMT based approach is clear especially in the case of per sentence accuracy: it outperforms the best-known one. It is important to notice that the high accuracy reported by Orosz and Novák [16] is mainly due to the usage of its integrated analyzer, that makes their tool language dependent. Without this PurePos performs lemmatizing only with a lemma guesser, that guesses the lemma for a word from its suffix and its part-of-speech tag. However the presented SMT based method is language independent and only needs a manually annotated and lemmatized corpus.

Table 18 Qualitative evaluation of the results produced by the SMT-based and the PurePos systems.

SMT			
Corpus frequency		Error type	
normalized logarithmic form	number of pieces	MSD tag form	textual form
-0.7764	41	[vmm]→[OOV]	imperative verb → OOV
-0.9777	12	[rv]→[OOV]	verbal adverb → OOV
-1.0294	20	[vmc]→[OOV]	conditional verb → OOV
-1.2529	40	[vmn]→[OOV]	infinitive verb → OOV
-1.3504	207	[vmi]→[OOV]	indicative verb → OOV
-1.6099	135	[afp]→[OOV]	qualificative adjective → OOV
-2.4850	18	[afp]→[vmi]	qualificative adjective → indicative verb
PurePos			
Corpus frequency		Error type	
normalized logarithmic form	number of pieces	MSD tag form	textual form
-1.2852	60	[pd3]→[tf]	demonstrative pronoun → definite article
-1.6460	19	[mc-]→[ti]	cardinal numeral → indefinite article
-1.9709	19	[rx]→[ccs]	adverb → coordinating conjunction
-2.3228	23	[vmi]→[afp]	indicative verb → qualificative adjective

Table 18 displays a qualitative evaluation of the results produced by the SMT_Baseline2 method of POS-tagging without lemmatization and that of PurePos. It describes the most frequent types of mistakes of each system. The main error types of the SMT-based method are not recognizing some words at all. However it is also clear that it performs a much better result on recognized words, since such expected errors (as in the case of PurePos) of

mistagging a pronoun to an article are not present. This result forecasts the application of a hybrid implementation, where the statistical behaviour would compensate the lackings of a more robust, however limited algorithm.

8 Conclusion

In this paper applicability of the SMT system was examined for part-of-speech disambiguation and lemmatization in Hungarian. Based on my observations these tasks can be considered as translations from plain text to analyzed one. The accuracy of such systems can achieve results of up to 96% accuracy. Although the quality of the above presented systems is behind the state of the art systems – still compareble to those available for Hungarian –, but in my work an absolutely automated system was created which finds the rules itself and we do not have to determine any features for training either. On the other hand this system is able to perform annotation and lemmatization simultaneously.

Some other observations are that we can achieve only minimal increase in the accuracy of the system with minimizing the target language dictionary, but this improvement is not proportional to the information loss. Further significant improvement was achieved by handling out-of-vocabulary words using a method based on word frequencies.

Results showed that only statistical methods are not enough to solve the task of POS-tagging; some kind of hybridization is necessary to improve the quality of the system. The achieved results were encouraging and they pointed out that this way of research contains further possibilities.

Acknowledgement This work was partially supported by TÁMOP – 4.2.1.B – 11/2/KMR-2011-0002 and the members of the Natural Language Processing Group of Pázmány Péter Catholic University.

References

1 Apache. Opennlp. `http://incubator.apache.org/opennlp/`, 2011.

2 Thorsten Brants. Tnt - a Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA, 2000.

3 D. Csendes, Cs. Hatvani, Z. Alexin, J. Csirik, T. Gyimóthy, G. Prószéky, and T. Váradi. Kézzel annotált magyar nyelvi korpusz: a Szeged Korpusz. In *I. Magyar Számítógépes Nyelvészeti Konferencia*, pages 238–247. Szegedi Egyetem, 2003.

4 Richárd Farkas, Dániel Szeredi, Dániel Varga, and Veronika Vincze. MSD-KR harmonizáció a Szeged Treebank 2.5-ben. In *VII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 349–353, Szeged, 12 2010. Szegedi Egyetem.

5 Guillem Gascó I Mora and Joan Andreu Sánchez Peiró. Part-of-Speech tagging based on machine translation techniques. In *Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I*, IbPRIA '07, pages 257–264, Berlin, Heidelberg, 2007. Springer-Verlag.

6 Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 209–212, Stroudsburg, 2007. Association for Computational Linguistics.

7 Péter Halácsy, András Kornai, Csaba Oravecz, Viktor Trón, and Dániel Varga. Using a morphological analyzer in high precision POS tagging of Hungarian. In *Proceedings of LREC 2006*, pages 2245–2248, 2006.

8 Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, Englewood Cliffs, NJ, 2. ed., [pearson international edition] edition, 2009.

9 Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 1 edition, 2000.

10 P. Koehn. Moses system. `http://www.statmt.org/moses/`.

11 P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180, Prague, 2007. Association for Computational Linguistics.

12 Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

13 A. Kornai, P. Rebrus, P. Vajda, P. Halácsy, A. Rung, and V. Trón. Általános célú morfológiai elemző kimeneti formalizmusa. In *II. Magyar Számítógépes Nyelvészeti Konferencia*, pages 172–176. Szegedi Egyetem, 2004.

14 László János Laki and Gábor Prószéky. Statisztikai és hibrid módszerek párhuzamos korpuszok feldolgozására. In *VII. Magyar Számítógépes Nyelvészeti K onferencia*, pages 69–79, Szeged, 12 2010. Szegedi Egyetem.

15 Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 135–139, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

16 György Orosz and Attila Novák. Purepos – an open source morphological disambiguator. In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science.*, Wroclaw, Poland, 2012.

17 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

18 Gábor Prószéky and Balázs Kis. A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 261–268, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics.

19 Adwait Ratnaparkhi. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, April 16 1996.

20 Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

21 G. Zipf. Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA, 1949.

22 János Zsibrita, Veronika Vincze, and Richárd Farkas. Ismeretlen kifejezések és a szófaji egyértelműsítés. In *VII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 275–283, Szeged, 12 2010. Szegedi Tudományegyetem.