INFORMATICA, 2010, Vol. 21, No. 1, 95–116

© 2010 *Institute of Mathematics and Informatics, Vilnius*

Reduction of Morpho-Syntactic Features in Statistical Machine Translation of Highly Inflective Language

Mirjam SEPESY MAUČEC, Janez BREST
Faculty of Electrical Engineering and Computer Science, University of Maribor
Smetanova 17, 2000 Maribor, Slovenia
e-mail: mirjam.sepesy@uni-mb.si

Received: April 2008; accepted: May 2008

Abstract. We address the problem of statistical machine translation from highly inflective language to less inflective one. The characteristics of inflective languages are generally not taken into account by the statistical machine translation system. Existing translation systems often treat different inflected word forms of the same lemma as if they were independent of each other, although some interdependencies exist. On the other hand we know that if we reduce inflected word forms to common lemmas, some information is lost. It would be reasonable to eliminate only the variations in inflected word forms, which are not relevant for translation. Inflectional features of words are defined by morpho-syntactic descriptions (MSD) tags and we want reduce them. To do this the explicit knowledge about both languages (source and target language) is needed. The idea of the paper is to find the information-bearing MSDs in source language by data-driven approach. The task is performed by a global optimization algorithm, named Differential Evolution. The experiments were performed using freely available parallel English–Slovenian corpus SVEZ-IJS, which is lemmatized and annotated with MSD tags. The results show a promising direction toward optimal subset of morpho-syntactic features.

Keywords: statistical machine translation, inflective language, morpho-syntactic description and Bleu metric.

1. Introduction

Automatic translation from one human language to another is a longstanding goal of computer science and artificial intelligence. Statistical data analysis has a prominent place in this research field. Statistical translation process uses two models (translation model and language model) as knowledge sources of a decoder, which produces a translation of a given input sentence. Decoding is decomposed into smaller steps, each is tied to words. Word-based translation process (Och and Ney, 2003) was afterwards improved by using phrases instead of words (Vogel *et al.*, 2003).

Estimating word-to-word (or phrase-to-phrase) alignments for the translation model is hindered by sparse data. This obstacles are most evident when dealing with morphologically rich languages. Inflective languages belong to them. Highly inflective languages

are, for example, Slavic languages. Our work is mainly devoted to the highly inflective Slovenian language. It is a South Slavic language. It shares its characteristics, in varying degrees, with many other inflective languages, especially the Slavic. As in the case of other inflective languages, we concentrate on reducing the perceived data sparsity. The techniques we investigate are language-independent and, as such, also applicable to other highly inflective languages.

Most frequently studied topic is the translation to English. From this perspective much of the morphological variation in inflective language is not reflected in English. It would be reasonable to eliminate variation in word forms, which is not reflected in English, for example by lemmatization or stemming. We have a more general situation, when the target language is not fixed to be English. To determine which morphological features in source language are important in translation process and which are not, depends tightly on the target language. If we translate from one inflective language to another inflective language, more morphological features need to be considered in the search for the appropriate translation.

The idea of the paper is data-driven morphology reduction. We try to eliminate the annotation of morphological features of Slovenian words to the subset, which reduces data sparseness and improves the translation quality. We avoid language-dependent rules with the aim the method to be applicable to any language pair.

1.1. *Organisation of This Paper*

The paper is organized as follows: We discuss the state of the art and related work in the continuation of this section. Section 2 describes the basics of statistical machine translation. It outlines two main types of probabilities used in translation model: translation probabilities and reordering probabilities. The induction of linguistic information in learning these two probabilities is discussed. In Section 3 we describe the lemma-MSD representation of linguistic information. The main idea of the paper is described in Section 4, where the reduction of MDS is performed by DE algorithm. Experimental results are reported in Section 5. The Section 6 concludes the paper with the discussion of the achievements of this work and an outlook on possible future directions is given.

1.2. *Previous Work*

Many researchers have studied the effects of morphological features on machine translation. In Niessen and Ney (2000) the authors explore morpho-syntactic restructuring to improve English-German alignment. Their experiments show significant improvement in translation quality. Their techniques are specific to this language pair. In Niessen and Ney (2004) they introduce the idea of a hierarchical lexicon, where a word is represented at various levels of inflectional specificity, starting with the base form. Their models were able to infer translations of word forms from translation of other word forms of the same lemma. They also suggest solutions for two specific aspects of structural difference, namely, question inversion and separated verb prefixes, specific for German

language. Their technique yielded a reduction in subjective semantic error rate. In the paper Popović and Ney (2004a) the authors build on the concept of the hierarchical lexicon. They use a modified EM alignment algorithm to treat each of these representations as a hierarchy, with alignment possible at any of three levels. They evaluate effects on a variety of baseline systems, producing at least modest improvements in every case. Moreover in Popović and Ney (2004b) they use knowledge of the specific source language to remove inflectional morphemes or function words that are not translatable, leading to a significantly reduced lexicon, and improved translations. In Spanish–English translation (Gispert *et al.*, 2006) corpus transformation on a word basis was studied. Words were mapped to linguistically-enriched tokens. The effect on the word alignment quality was studied. Results of different transformations were evaluated against a manual reference. In our experiments manual reference is not available. In Lee (2004) the author observes that in Arabic–English translation, one Arabic word frequently aligns with multiple English words, owing to functional affixes in Arabic. They developed the algorithm, which identifies morphemes to be merged or deleted in the morphologically rich language to induce the desired morphological and syntactic symmetry. The technique improves Arabic-to-English translation. In Goldwater and McClosky (2005) a variety of experiments in Czech–English machine translation has been set up. They found that certain morphological tags were more useful when treated them as discrete input words, while others provided a greater benefit when attached directly to their lemmas. English–Czech factored machine translation was studied in Bojar (2007). In their experiments the target language was highly inflective. They explicitly model Czech morphology using a separate morphological language model. In Šveikauskiene (2005) the syntactical structure of Lithuanian language is described by graphs to help machine translation. Morphology of highly inflective languages was widely studied in the field of statistical language modelling for speech recognition (Maučec *et al.*, 2003; Vaiciunas and Raskinis, 2006; Rotovnik, 2006; Žgank *et al.*, 2001). In the present paper we are studying translation from highly inflected language into English, where only a language model of English is needed. A language model of Slovenian language (Maučec *et al.*, 2009) would be a valuable knowledge source if we would translate in the opposite direction.

State of the art and related work show that morphological information can be useful. Using all the morphological information available results in increased data sparsity and consequently error-full translations. The aim of this paper is to extract the useful subset of morphological features in source language using an evolutionary algorithm. The problem could also be seen as a clustering problem, where data sparsity is reduced by grouping words sharing the same morphological information into one cluster. The techniques used in Serban and Câmpan (2008), Krilavicius and Zilinskas (2008) could be applied in this case. We do not claim that an evolutionary algorithm is the best choice. The main contribution of the paper is to show that the reduction of morphological information in highly inflective language improves the translation results when the target language is less inflected, like for example English language.

1.3. *Machine Translation Evaluation*

The evaluation of machine translation systems is a growing field. Human evaluation is not always affordable. Recent evaluation activities focus on determining the best methodology for evaluating translation quality with automatic metrics. Statistical translation systems are most commonly judged by *BLEU* metric (Papineni *et al.*, 2002). It is also used in this paper.

2. Statistical Machine Translation

Statistical machine translation is formulated as follows: a source string of words $\mathbf{f} = f_1^J = f_1 \dots f_j \dots f_J$ is to be translated into a target string of words $\mathbf{e} = e_1^I = e_1 \dots e_i \dots e_I$. Words f_j belong to the source vocabulary $\mathbf{F}$ and the words e_i to the target vocabulary $\mathbf{E}$. The string with the highest probability is chosen from among all possible target strings, as given by the Bayes decision rule: $\mathbf{e} = \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e})P(\mathbf{f}|\mathbf{e})$. $P(\mathbf{e})$ is the language model of the target language, whereas $P(\mathbf{f}|\mathbf{e})$ is the translation model. The language and translation models are independent knowledge sources. The $\arg\max$ operation denotes the search for an output string in the target language. Language model is standard word-based trigram model.

The paper focus on translation model. In translation model $P(\mathbf{f}|\mathbf{e})$ the term target language refers to the Slovenian language and the source language refers to the English language.

2.1. *Alignment Model*

The translation model is based on word alignment (Och and Ney, 2003). The training corpus is usually sentence-aligned and we perform word-alignment search. Given an English string $\mathbf{e}$ and a Slovenian string $\mathbf{f}$, a word alignment is a many-to-one function that maps each word in $\mathbf{f}$ onto exactly one word in $\mathbf{e}$, or onto the NULL word. The NULL word is an invisible word in the initial position of an English sentence e_0. It accounts for Slovenian words that have no counterpart in the English sentence. More than one Slovenian word can be mapped onto the same English word. An example of alignment is given in Fig. 1.

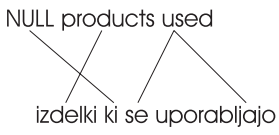

Fig. 1. Example of an alignment. NULL word is counterpart of Slovenian words that have no translation in English.

We can rewrite the probability $P(\mathbf{f}|\mathbf{e}) = P(f_1^J|e_1^I)$ by introducing the "hidden" alignment $a_1^J = a_1 \dots a_j \dots a_J$; $a_j \in 0, \dots, I$:

$$P\big(f_1^J|e_1^I\big) = \sum_{a_1^J} P\big(f_1^J, a_1^J|e_1^I\big). \tag{1}$$

IBM Model 4 (Brown *et al.*, 1993) computes the probability $P(f_1^J, a_1^J|e_1^I)$ of a particular alignment a_1^J and a particular sentence f_1^J given a sentence e_1^I. This probability is a product of five individual decisions:

$t(f_j|e_i)$ – translation probability. It is the probability of Slovenian word f_j being a translation of English word e_i.

$n(\phi_k|e_i)$ – fertility probability. An English word can be translated into zero, one or more than one Slovenian word. This phenomenon is modelled by fertility. The fertility $\phi(e_i)$ of an English word e_i is the number of Slovenian words mapped to it. The probabilities of different fertility values ϕ_k for a given English word are trained.

p_0, p_1 – fertility probability for e_0. Instead of fertilities $\phi(e_0)$ of a NULL word, one single parameter $p_1 = 1 - p_0$ is used. It is the probability of putting a translation of a NULL word onto the some position in a Slovenian sentence.

$d_1(\Delta j|\mathcal{A}(e_i), \mathcal{B}(f_j))$ – distortion probabilities for the head word. Δj is the distance between the head of current translation and the previous translation. It may be either positive or negative. Distortion probabilities model different word order in the target language in comparison to the word order in the source language. Classes of words are used instead of words (mappings $\mathcal{A}$ and $\mathcal{B}$).

$d_{>1}(\Delta j|\mathcal{B}(f_j))$ – distortion probabilities for the non-head words. In this case Δj denotes the distance between the head and non-head words.

This was a short overview of the translation model. Readers interested in a more detailed description are referred to the paper (Brown *et al.*, 1993).

Correspondence between the words in the source and the target language are learned from bilingual corpora and usually little or no linguistic knowledge is used in this learning process. The application of some knowledge about the language under consideration is reasonable.

We will take a look only at two estimated probabilities, which are part of an alignment model, namely word translation probability and word reordering probability.

2.2. *Word Translation Probability*

Translation probability $t(f|e)$ is the probability of source word f being a translation of target word e. In general, the set of translation probabilities contains all word forms occurring in this training corpus, not taking into account whether or not they are inflected forms of the same lemma. Many words are only seen once in training (will be evident in

section with experiments) and it is difficult to learn the correct translation of them. New input sentences are expected to contain unknown word forms for which no translation probability can be retrieved from the translation model.

It is straightforward to aim at taking into account the interdependencies between the inflected forms of the same lemma when learning word translation probabilities.

2.3. *Word Reordering Probability*

In general, words in target language are not in the same order as in the source language. The more differences in word order between two languages, the more difficult to extract alignment and the more challenging the translation task. Slovenian and English exhibit a quite remarkable differences in word order.

In learning reordering probabilities we distinguish the heads from the non-heads. The head is the leftmost word of the group mapped to the same English word. All subsequent words in the same group are non-heads. A group of words does not always contain neighboring words.

$d_1(\Delta|\mathcal{A}(e), \mathcal{B}(f))$ denotes the distortion probability for the head word f. Δ is the distance between the head of current translation, and the previous translation. It may be either positive or negative. $d_{>1}(\Delta|\mathcal{B}(f))$ is distortion probability for the non-head word. In this case Δ denotes the distance between the head and non-head word.

Words are mapped into classes before training the distortion probabilities is performed. There are two independent mappings, $\mathcal{A}$ for English words and $\mathcal{B}$ for Slovenian words. The grouping of words into classes is based on the assumption, that word displacement depends on some features, which are common to many words. Words can be grouped into classes by automatic clustering algorithms. We assume that using morphosyntactic information as indicator for word grouping is useful.

3. Morphosyntactic Features

For each word in the corpus we have the following information:

- word form, i.e., the word as it appear in the sentence;
- lemma, i.e., a word in its canonical form (e.g., infinite for verbs, nominative singular for regular nouns, etc.);
- part of speech (POS), e.g., noun, verb, adjective, adverb, etc.;
- additional morpho-syntactic tags, i.e., gender (masculine, feminine or neuter), number (singular, plural or dual), etc.

Last two types of informations are concatenated into MSD tag in lemma-tag representation. We have lemma and MSD tag attached to it. If f is word form, f^* denoted its lemma-tag representation. $\mathbf{F}^*$ is the set of all possible representations of vocabulary words. The first letter of an MSD encode the POS. The letters following the POS give the values of the position dependent morpho-syntactic features. For example, the MSD tag `Ncfsn` expands to POS `Noun`, `type:common`, `Gender:feminine`,

```
<seg lang="sl">
<w ana="Ncfsn" lemma="komisija">Komisija</w>
<w ana="Pp3msa--y-n" lemma="on">ga</w>
<w ana="Vmip3s--n" lemma="posredovati">posreduje</w>
<w ana="Aopmsd" lemma="evropski">Evropskemu</w>
<w ana="Ncmsd" lemma="parlament">parlamentu</w>
<c>.</c>
</seg>
```

Fig. 2. Annotated sentence (No. 12315) from Slovenian part of the SVEZ corpus. Translation into English: *The Commission shall forward it to the European Parliament.*

`Number:singular`, `Case:nominative`. Fig. 2 shows an example of an annotated sentence. The XML attribute `ana` contains the MSD tag.

The idea is to reduce the morpho-syntactic features to a subset of most useful ones. To formalize the reduction of morpho-syntactic features we have to define some parameters. N_{MSD} denotes the number of different MSD tags. The application of full or reduced kth MDS is determined by the transformation defined as

$$\mathbf{t}_k = (t_{k,0}, t_{k,1}, \ldots, t_{k,l}, \ldots, t_{k,n_k}), \qquad t_{k,l} \in \{1,0\} \wedge k \in \{1, \ldots, N_{MSD}\}, \quad (2)$$

n_k is the maximal number of morpho-syntactic features associated with the kth MSD. $t_{k,0}$ is associated with the lemma. If the value of $t_{k,l}$ is 0, lth feature is not used, and if $t_{k,l}$ is 1, lth feature is in use. If for kth MSD we ignore all morpho-syntactic features and use only lemmas, we have $\mathbf{t}_k = (1,0,0,0,\ldots,0)$. The rule R: $t_{k,0} = 1$ gives the assurance of keeping the lemmas in reductions and the rule $\bar{R}$: $t_{k,0} = 0$ eliminates all the lemmas in reductions.

The reduction is

$$\mathcal{R}\colon\ \mathbf{t}_k \to \mathbf{t}'_k, \quad (3)$$

where

$$0 < \sum_{l=0}^{n_k} t_{k,l} \leqslant \sum_{l=0}^{n_k} 1 = n_k + 1. \quad (4)$$

The same is true for the sum of $t'_{k,l}$.

The application of reduced MSD to a word f in case of preserved lemma (by using the rule R) is defined as

$$\mathcal{L}_1\colon\ (f, \mathbf{t}_k, R) \to f^*, \quad (5)$$

and in case of eliminated lemma (by using the rule $\bar{R}$)

$$\mathcal{L}_2\colon\ (f, \mathbf{t}_k, \bar{R}) \to f^*. \quad (6)$$

The set of all possible reductions is large. We are looking for a subset, which results in the best performance of the translation system. Let us define the $\mathcal{R}$ as a vector of N_{MSD} reductions:

$$\mathcal{R} = \big(\mathcal{R}(\mathbf{t}_1), \mathcal{R}(\mathbf{t}_2), \ldots, \mathcal{R}(\mathbf{t}_{N_{MSD}})\big). \tag{7}$$

A common choice for automatic evaluation of machine translation is *BLEU* metric. We decided to use it in a search for the optimal reduction $\mathcal{R}^*$. The search is formulated as

$$\mathcal{R}^* = \arg\max_{\mathcal{R}} \textit{BLEU}(Dev). \tag{8}$$

BLEU is calculated on some held-out data (Dev denotes development set), which is neither part of train set nor test set.

The reduction of MSDs is not orthogonal. The influence of the reduced MSD of one word in the sentence is spread among the whole sentence. The reduction of one MSD may change alignments of many sentences, having this MSD attached to its units (unit is lemma-tag representation of a word), and consequently the estimated probabilities of all its units. We see the problem as global optimization problem and MSDs being the parameters to be optimized.

3.1. *MSD in Word Translation*

Word translation probability $t(f|e)$ estimates how probable it is that the inflected word form f is the translation of word e. Which features of f give us useful information? The lemma is the main knowledge source and the reduced MSD add some information. To determine the translation probability the transformation $\mathcal{L}_1(f)$ is used for source word f: $t(\mathcal{L}_1(f)|e)$.

3.2. *MSD in Word Reordering*

Word order depends on syntactic features of words and do not depend on words meaning. Distortion probability in its basic form introduce two mappings, $\mathcal{A}$ for target words e and $\mathcal{B}$ for source words f. For source words mapping we use the transformation $\mathcal{L}_2(f)$: $d_1(\Delta|\mathcal{A}(e), \mathcal{L}_2(f))$ for head words and $d_{>1}(\Delta j|\mathcal{L}_2(f))$ for non-head words.

4. Differential Evolution Algorithm

The optimal subset of morpho-syntactic features is to be determined by a search algorithm, evaluating a number of possible reductions. Because the space of reductions is large, it is not reasonable to search the whole search space. Heuristic functions are used to apply some knowledge about the search space, but the mechanism is needed to keep out of danger to be stuck in local optima. The idea of evolution is used in this paper. The search of optimal subset of morpho-syntactic features is performed by Differential

Evolution (DE) algorithm (Storn and Price, 1997; Price *et al.*, 2005), being a simple and powerful evolutionary algorithm, which has been applied on a wide range of research areas (Tvrdik, 2006; Feoktistov, 2006; Brest *et al.*, 2006, 2007)

The advantages of the DE are mainly good convergence property and robustness.

Before our problem is reformulated in the sense of DE, let us introduce some common terms used in DE algorithm. Fitness function is a function, which is to be maximized in this work. Fitness function depends on the set of optimization parameters. A set of parameters values is called an individual. It is represented as a vector. Each individual is a candidate for the solution. A population consists of NP parameter vectors. Many generations are created by means of evolution. We have one population for each generation. Differential evolution algorithm creates new candidate solutions by combining the parent individual and several other individuals of the same population. We have three operations: mutation, crossover, and selection. The mutation and crossover are used to generate new candidates, and the selection determines which of the candidates will survive into the next generation. A candidate replaces the parent only if it has better value of fitness function. Readers interested in more details of DE are referred to Price et al. (2005).

First we have to define fitness function and optimization parameters. We are searching for the optimal reduction $\mathcal{R}^*$. The dimensionality of the problem equals the number of different MSD, e.g., N_{MSD}. DE searches for the optimal $\mathcal{R}^*$ in iterative manner by exploring many individuals $\mathbf{x}_i \in \bigcup \mathcal{R}$. Fitness function is *BLEU* metric.

We follow the description of DE, presented in Tvrdik (2006), Feoktistov (2006):

ALGORITHM 1. Differential evolution
1 generate $P = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{NP})$ at random
2 **repeat**
3 **for** i = 1 **to** NP **do**
4 compute a mutant vector $\mathbf{u}$;
5 create a trial vector $\mathbf{y}$ by crossover of $\mathbf{u}$ and $\mathbf{x}_i$;
6 **if** *BLEU*$(\mathbf{y}) >$ *BLEU*$(\mathbf{x}_i)$ **then**
7 $\mathbf{x}_i = \mathbf{y}$;
8 **endfor**
9 **until** stopping condition;

The DE algorithm in pseudo-code is written as Algorithm 1. We keep the notations defined by DE community. In our case the vectors $\mathbf{x}_i, \mathbf{y}, \mathbf{u}$ are different reductions $\mathcal{R}$. Step 7 is a differs from that in the original DE algorithm, where $\mathbf{y}$ is assigned to $\mathbf{x}_i$ after completion of the loop (one generation). Therefore the original DE maintains two populations of the same size NP.

Description of Steps 4, 5, and 6 follows.

Step 4. Compute a mutant $\mathbf{u} \in \bigcup \mathcal{R}$.

The most popular strategy of DE (*'DE/rand/1/bin'*) generates the mutant $\mathbf{u}$ by adding the weighted difference of two individuals

$$\mathbf{u}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), \tag{9}$$

where r_1, r_2 and r_3 are three distinct indexes taken randomly. They are also different from the current i. F is a scaling factor, which is usually taken from interval $(0, 1]$. An example of the mutation operation is given in Fig. 3.

Before we define the addition (and subtraction) of two individuals (e.g., reductions), let us introduce one simplification. The reduction of morpho-syntactic features is restricted to be performed from the right to the left. The interdependence of morpho-syntactic features is more complex, but with some simplification we can make an assertion that more significant features are positioned before less significant ones.

Before we describe the generation of the mutant vector $\mathbf{u}$, let us define the length of the transformation $\mathbf{t}_k$ associated with the kth MSD:

$$len(\mathbf{t}_k) := \max_l\{l\colon t_{k,l} = 1;\ l = 1, \ldots, n_k\}. \tag{10}$$

The length of the kth transformation in mutant vector $\mathbf{u}$ is

$$len(\mathbf{u}_{i,k}) := len(\mathbf{x}_{r_1,k}) + F\big(len(\mathbf{x}_{r_2,k}) - len(\mathbf{x}_{r_3,k})\big). \tag{11}$$

The value of len should be between 1 and n_k. If it is under the lower bound, it is set to be 1. If it is over the upper bound, it is set to be n_k.

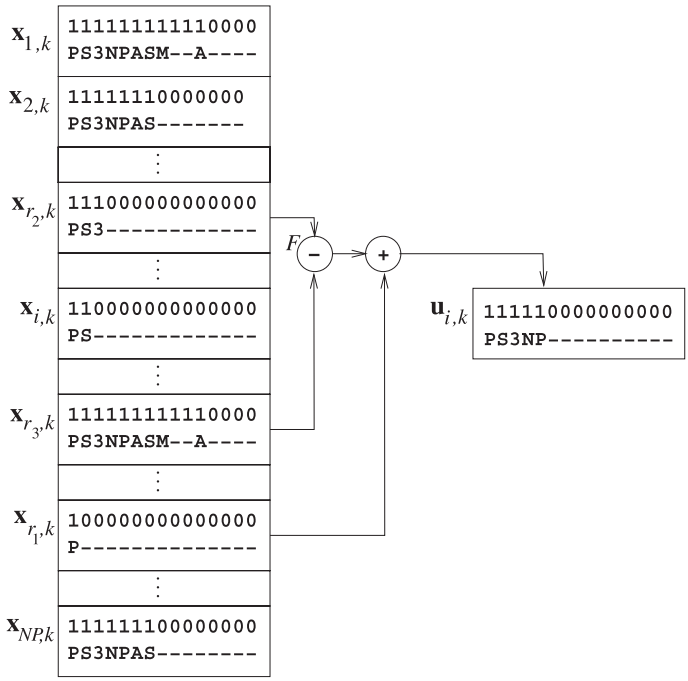

Fig. 3. Example of the mutation operation for $\mathbf{x}_k$ (kth MSD), assume that $F = 0.5$.

The components of the kth transformation in the mutant are

$$u_{i,k,l} = \begin{cases} 1, & \text{if } l \leqslant len(\mathbf{u}_{i,k}),\ l = 1, \ldots, n_k, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The same procedure is repeated for all k transformations.

Step 5. Create a trial vector $\mathbf{y}$ by crossover of $\mathbf{u}$ and $\mathbf{x}_i$.

The elements $\mathbf{y}_k$, $k = 1, 2, \ldots, N_{MSD}$ of trial vector $\mathbf{y}$ are built up by the crossover of its parents $\mathbf{x}_i$ and $\mathbf{u}$ using the following rule

$$\mathbf{y}_{i,k} = \begin{cases} \mathbf{u}_{i,k}, & \text{if } \mathrm{rand}(0,1) \leqslant CR \text{ or } k = k_{\mathrm{rand}}, \\ \mathbf{x}_{i,k}, & \text{otherwise,} \end{cases} \tag{13}$$

$$i = 1, 2, \ldots, NP \wedge k = 1, 2, \ldots, N_{MSD}.$$

CR is a crossover parameter or factor within the range $[0, 1)$ and presents the probability of creating parameters for the trial vector from a mutant vector. Index k_{rand} is a randomly chosen integer within the range $[1, N_{MSD}]$. It is responsible for, the trial vector containing at least one parameter from the mutant vector.

Steps 6 and 7. Selection.

The selection operation selects, according to the fitness value (i.e., *BLEU* score) of the population vector and its corresponding trial vector, which vector will survive to be a member of the next generation.

The search for the optimal subset of MSD features cannot be performed by the full search algorithm, because it has exponential time complexity. If $\overline{len(\mathbf{t}_k)}$ denotes the average length of the transformation $\mathbf{t}_k$, we have the complexity of

$$\mathcal{O}\big(2^{N_{MSD} \cdot \overline{len(\mathbf{t}_k)}}\big). \tag{14}$$

We simplified the search to be the reduction of $\mathbf{t}_k$ performed strictly from the right to the left. In this case full search has polynomial time complexity

$$\mathcal{O}\big({N_{MSD}}^{\overline{len(\mathbf{t}_k)}}\big). \tag{15}$$

Although search space is considerably reduced, it will take a reasonable amount of time only for small parameter values. In our experiments the value of N_{MSD} is 1042 and $\mathbf{t}_k$ is 8.79. One iteration takes approximately 1.5 hour. It can be seen that it is practically impossible to find the best solution. The idea is to find a good solution in a fixed amount of time. The time is limited by the number of iterations I. Because DE algorithm uses a population of size NP, we have $\frac{I}{NP}$ generations. In our experiments NP is set to 50 and I to 1500. We evaluated 30 generations. We see that $I \ll {N_{MSD}}^{\overline{len(\mathbf{t}_k)}}$.

5. Experiments

5.1. *SVEZ-IJS Corpus*

In our research we focus on translation from more inflective language to less inflective one. The experiments were carried out for the translation direction Slovenian to English.

The experiments were performed on the SVEZ-IJS parallel English–Slovenian corpus (Erjavec, 2006). It contains $2\times$ 5 million words of EU legal text and was made on the basis of the translation memory produced by the Translation Department at SVEZ (The Office of the Government for the European Affairs). The corpus was linguistically tagged in three steps:

1) tokenization of the text (split it into words, punctuation marks and sentences),

2) assignment of MSD to words with TnT tagger,

3) lemmatization of the text with CLOG.

The authors report the accuracy of linguistic annotation: 94.4% for POS, 86.6% for MSD and 98.4% for lemmatization. The inflectional nature of Slovenian language is evident from Table 1. It is interesting that English part contains 20% more words than Slovenian part. The average English sentence is 3 words longer than the average Slovenian sentence. One reason lies in determiners and pronouns. The subject pronoun in English (I, she, we, etc.) usually have a zero form in Slovenian. The Slovenian part contains more than twice as many unique words than English part, because of highly inflectional nature of Slovenian language.

The corpus was split into training, development and test sets in the ratio 8:1:1. The homogeneous data partitioning was used. The test and development sets were taken at regular intervals from the corpus. In the training set we only used sentences shorter than 16 words because of computational complexity. No limit regarding sentence length was set for sentences in the development and test sets. In Table 2 some statistics are collected.

5.2. *Translation Model Training*

Vocabularies of both languages (English and Slovenian) contained all the words of the training corpus. English words were mapped into 100 automatically built classes (mapping $\mathcal{A}$ in Section 2.1). Slovenian words were mapped into classes, based on MSDs, when

Table 1
SVEZ-IJS statistics

	Slovenian	English
Sentences	273,477	
Words (tokens)	4,527,771	5,429,878
Word forms (types)	182,212	76,625
Singletons	82,295	35,564
Lemmas	92,399	–
N_{MSD}	1042	–

Table 2
Statistics of train, development, and test sets

	Train		Development		Test	
	Slo.	Eng.	Slo.	Eng.	Slo.	Eng.
Sentences	98,291		12,839		12,839	
Words (tokens)	614,997	694,585	79,222	89,287	78,803	88,816

possible (see mapping $\mathcal{L}_2$ in Section 3.2). Each MSD defined it's own class. Number of classes varied depending on reductions in MSD tags. When MSD tags were absent (in the first and second experiments), Slovenian words were clustered automatically (mapping $\mathcal{B}$ in Section 2.1).

In experiments many translation model trainings were performed, which differ in extend of morphological information used in Slovenian part of the training corpus. The translation model training was performed using a program GIZA++ (Och and Ney, 2003). IBM Models 1–4 were used as stepping stones. 10 iterations for each Model were performed in all experiments.

Finally translation model (Model 4) and language model for English language were used in decoding development and test sets of sentences. Decoded development set was used in deciding whether or not the current translation model has improved. The translation model, which gave the best *BLEU* score on development set, was used in decoding the test set of sentences. Decodings were performed by the ISI ReWrite Decoder (Germann, 2003).

5.3. *Language Model Training*

All models use the same language model for English language. It was built only once. The English part of the training corpus was used for training the language model. It was made by using the CMU-SLM toolkit (Rosenfeld, 1995). The vocabulary contained 64,000 most frequent words. A conventional trigram model was built with Good-Turing discounting for bigrams and trigrams with counts lower than 7. No n-grams were discarded. The corpus is relatively small, so there are a lot of singletons with significant information. The language model perplexity of the test set was 198, and the OOV rate was 0.64%.

5.4. *Initial Experiments*

Four reference experiments were performed, and the obtained results are presented in Table 3. Results in the last row will be discussed in Section 5.6.

In all experiments corpus transformation was performed only on Slovenian part of the corpus, English part remained the same. During the set-ups of each experiment we counted the number of different units (i.e., types), and the number of different MSDs in Slovenian part of the corpus. In each experiment *BLEU* scores on the test set were

Table 3
The experiments

Modelling unit	Types (in SI part)	BLEU	Num. MSD (in SI part)
Words	63,575	34.44	–
Lemmas	44,916	34.21	–
Lemmas + POS	47,800	34.23	12
Lemmas + Full MSD	83,237	33.83	1024
Lemma + reduced MSD	53,565	35.77	105

computed. The first row reports those result obtained using words (full word forms) as modelling units. Slovenian words were clustered automatically into 1000 classes. 55% of word forms were only seen once in the training corpus (see Table 1), and it is obvious that learning the correct translations is difficult for many words.

In the second experiment lemmas were used instead of full-word forms (only in the Slovenian part of the corpus). In these experiments different word forms, derived from the same lemma are considered equivalent. In this experiment we had: $\mathbf{t}_k = (1, 0, 0, 0, \ldots, 0);\ k \in \{1, \ldots, N_{MSD}\}$. Slovenian lemmas were clustered automatically into 1000 classes. The number of types decreased considerably. Because data sparsity was reduced to a great extent, we expected improved translation results. Surprisingly, the results become slightly worse (see second row in Table 3). We reduced data sparsity, but we also lost some useful information. This experiment clearly demonstrate that some inflected forms of Slovenian are relevant for translation. Undoubtedly, morphosyntactic information should not be completely disregarded.

In the third experiment, POS tags were attached to lemmas. In this experiment we had: $\mathbf{t}_k = (1, 1, 0, 0, \ldots, 0);\ k \in \{1, \ldots, N_{MSD}\}$. The results did not improve, when compared with lemma-based models. Using only POS tags is insufficient for determining the information relevant for translation.

In the next experiment, lemmas with full MSDs were used. In this experiment we had: $\mathbf{t}_k = (1, 1, \ldots, 1);\ k \in \{1, \ldots, N_{MSD}\}$. The worst results were obtained. In this experiment the corpus was the scarcest. A full MSD adds some for translation redundant information.

The above experiments confirm the hypothesis, that only some MSD tags are relevant for translation. This morphology-related problem can be handled by a human translator, who will separate relevant MSD tags from the irrelevant ones. The goal of the paper is to remain purely statistic.

5.5. *Experiments with Different Lengths of MSDs*

In this section we systematically study MSD tags of each POS in isolation. Two sets of experiments were performed where we studied MSD reduction of just one POS. For each POS under consideration we gradually increased MSD length (i.e., $len(\mathbf{t}_k)$) from

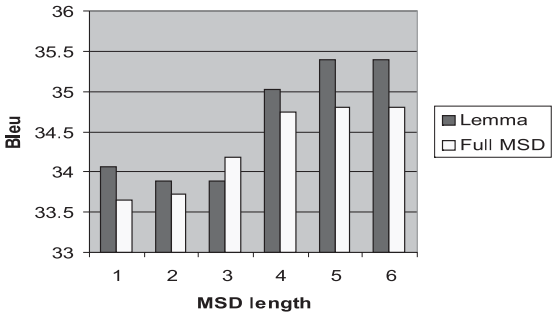

Fig. 4. Results of MSD reduction for noun.

1 to 6. In the first set of experiments all other words were reduced to lemmas and in the second set of experiment other words were mapped to lemmas with full MSD tag. We calculated *BLEU* score on development set. Results for noun, verb, adjective, and pronoun are given in Figs. 4–7. For noun the best MSD length seems to be 5 or 6. For verb the best results were obtained with MSD length 3. For adjective the best length is 4 (in the first experiment) or 3 (in the second experiment). For pronoun the length 6 brought best *BLEU* score. It is interesting to note that only for noun the lemmatized version outperformed full-MSD version in all experiments. For all remaining POSs it is vice versa (like in initial experiments). The biggest difference between obtained results in different set-ups is evident for adjective. The best result was obtained when MSDs for adjective were between 1 and 3 and all other MSDs were at full length.

Similar experiments were reported for Spanish–English translation (Gispert *et al.*, 2006). Bilingual lexicons were obtained form word-based statistical word alignments. Entropy was calculated with respect to each MSD tag corresponding to a given POS. With the respect to the value of entropy MSD tags were classified into irrelevant and relevant tags.

It is commonly known that non-monotonicity poses difficulties for statistical machine translation. The choice of word order is related to morpho-syntactic features of a word. The importance of MSD tag should not be studied on the level of word-by-word translations but on the level of sentence alignment. The reduction of MSDs is not orthogonal. We studied it as an optimization problem in the next section.

5.6. *Experimental Results with MSD Reduction Using DE Algorithm*

There are many optimization algorithms. DE is one of them. We have chosen it because it has good convergence properties. DE is a population based algorithm. The initial population was generated uniformly at random. The population contained 50 individuals. Scaling factor F was randomly taken from the interval $[0.3, 0.7]$, and crossover parameter CR from the interval $[0.1, 0.7]$. The fitness value was a *BLEU* score on the development set. Thirty generations were evaluated. Fig. 8 shows *BLEU* scores (on test and development sets) for all individuals in the initial and final population. It can be seen that the population moves (slowly) to the upper right-hand corner. The advantage of DE is that it is

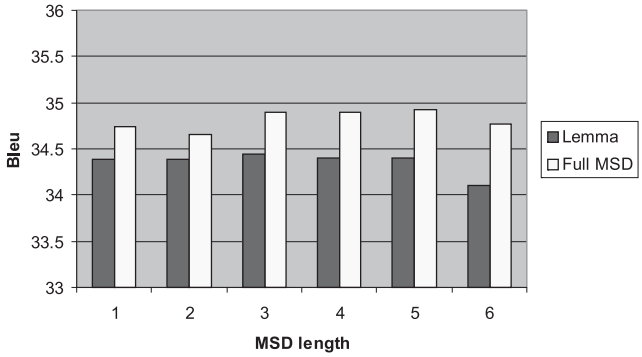

Fig. 5. Results of MSD reduction for verb.

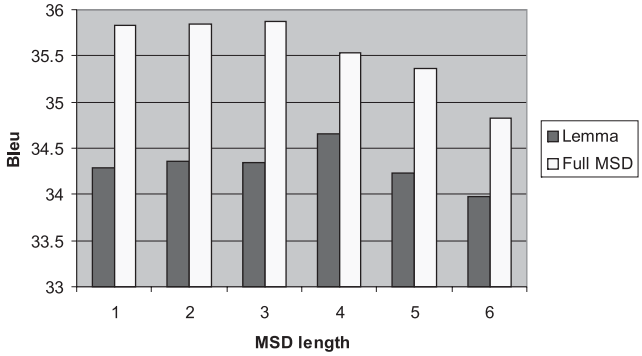

Fig. 6. Results of MSD reduction for adjective.

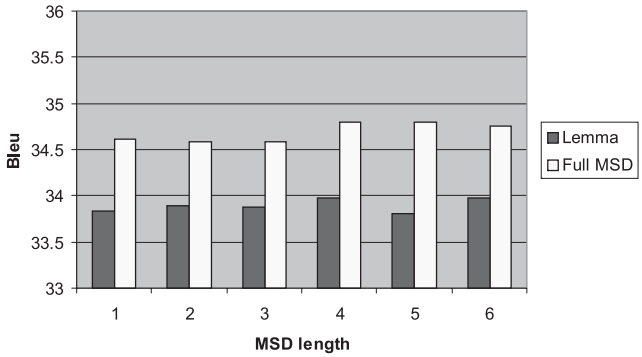

Fig. 7. Results of MSD reduction for pronoun.

population based. With DE we obtained more (good) solutions at the end of the optimization process. Using for example simulated annealing gives us only one solution. When we have more solutions, we have a control over similarities between development and test set. Although the results on test set are not allowed to influence the decisions made

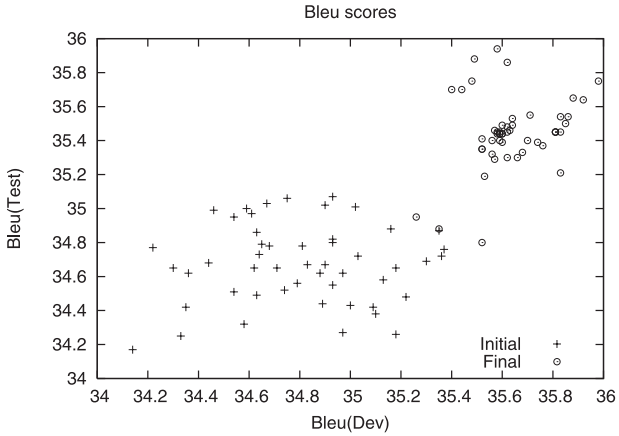

Fig. 8. Initial and final populations.

by DE, the population evenly dispersed in the right half of the figure will give us notice about dissimilarities between development and test set.

The last row in Table 3 contains the best *BLEU* test set score from among all individuals in the final population. The results show that it outperformed all previous experiments. The relative improvement in *BLEU* score is 3.86%. The number of different MSDs reduced, approx., to tenth the original set. Table 4 compares the number of MSDs for different POSs in described experiments. First column gives the name of POS. Next two columns contain the lengths and the number of different MSDs when no reduction was performed. Following two columns give the results of experiments described in Section 5.5. In the last two columns the results obtained with DE are given. We see the evident difference between the numbers of different MSD, obtained in distinct experiments. Based on this data the question on quality loss arise. Some information got lost, but the translation quality has improved, because we succeeded in data sparsity reduction. The improvement depends on a corpus size (Maučec *et al.*, 2006).

If the translation expert were to look at the reduced tag set, he/she will probably disagree that these features are the most relevant for translation. We have extracted them

Table 4

Number of different MSDs per POS

POS	MSD at ful length		Reduction of one POS		Reduction by DE	
	$len(\mathbf{t}_k)$	N_{MSD}	$len(\mathbf{t}_k)$	N_{MSD}	$len(\mathbf{t}_k)$	N_{MSD}
Noun	8	108	5	104	4	18
Verb	15	88	3	11	3	11
Adjective	9	189	4	17	3	6
Pronoun	13	550	6	367	4	42
All POS	–	1042	–	606	–	105

Table 5

Example of translations obtained by using different modelling units. f is input sentence and e a translation. e_0 is a reference translation

Sentence No. 444
f: KER SO BILI V ALBANIJI ZABELEŽENI PRIMERI KOLERE
e_0: WHEREAS CASES OF CHOLERA HAVE BEEN RECORDED IN ALBANIA
Modelling unit: **word**
f: KER SO BILI V ALBANIJI UGOTOVLJENI PRIMERI KOLERE
e: WHEREAS ESTABLISHED TO EXAMPLES OF CHOLERA ACUTELY WITHDRAWN
Modelling unit: **lemma**
f: KER BITI BITI V ALBANIJA ZABELEŽEN PRIMER KOLERA
e: WHEREAS IN ALBANIA BE REPORTED EXAMPLES OF CHOLERA
Modelling unit: **lemma + POS**
f: KER[C] BITI[V] BITI[V] V[S] ALBANIJA[N] ZABELEŽEN[A] PRIMER[N] KOLERA[N]
e: WHEREAS IN ALBANIA REPORTED EXAMPLES OF CHOLERA
Modelling unit: **lemma + full MSD**
f: KER[CSS] BITI[VCIP3P-N] BITI[VCPS-PMA] V[SPSL] ALBANIJA[NPFSL] ZABELEŽEN[AFPFDA] PRIMER[NCFDA] KOLERA[NCFSG-]
e: WHEREAS THEY HAVE BEEN ACUTELY IN ACKNOWLEDGMENT OF CHOLERA UNKNOWN
Modelling unit: **lemma + reduced MSD (by DE)**
f: KER[CS] BITI[VCIP3] BITI[VCP] V[SPS] ALBANIJA[N] ZABELEŽEN[A] PRIMER[NCFD] KOLERA[NCFS]
e: WHEREAS IT HAS BEEN REPORTED EXAMPLES OF CHOLERA IN ALBANIA

using the data-driven approach without expert knowledge about the language pair under consideration. The aim was to improve translation quality.

Differences in *BLEU* scores between all experiments are relatively small. We see the reason in corpus characteristics. About 40% of words and about 50% of lemmas are singletons. If a word only appears a handful of times in the training corpus, the system will have difficulty determining its alignment. The complexity of alignment depends on sentence length. The experimental corpus is an example of juristic language. Almost half of sentences are longer than 16 words and syntactically complex. At the time of this writing SVEZ-IJS was the only available Slovenian–English corpus.

An example of translations obtained in different experiments is given in Table 5. It can be seen, that we obtained comprehensible translations using lemmas, lemmas + POS, and lemmas + reduced MSD.

The goal of our work was to improve translation results by making input language (Slovenian) more similar to output language (English). This was done by suppressing unnecessary morphological distinctions. While human judgment would probably be the

best metric for evaluating our achievements, *BLEU* scores allowed us to easily compare many different translation models.

6. Conclusion

In this work we studied translation from the more-inflected Slovenian language to the less-inflected English language. We have presented an approach for the data-driven reduction of MSD tags, with the aim of improving statistical translation (by data-sparsity reduction). The reduction of MSDs was performed by differential evolution algorithm. We do not claim that this is the best choice. Instead of the DE algorithm, any other evolutionary algorithm could be used during the optimization process, and moreover, one might as well use any kind of search algorithm that returns a single solution of the problem.

No language specific rules were used. The same approach could be used for other pairs of languages with the proviso that we translate from more-inflected language to less-inflected language.

The results of our experiments show the existence of an optimal subset of MSDs. This approach is time-consuming, because we used the program code of SMT as a whole. Future work should focus on the use of parallel computing systems.

References

Bojar, O. (2007). English-to-Czech factored machine translation. In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic, pp. 232–237.

Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6), 646–657.

Brest, J., Bošković, B., Greiner, S., Žumer, V., Sepesy Maučec, M. (2007). Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 11(7), 617–629. DOI: 10.1007/s00500-006-0124-0.

Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 263–311.

Erjavec, T. (2006). The English–Slovene ACQUIS corpus. In: *Proceedings of LREC*, pp. 2138–2141.

Feoktistov, V. (2006). *Differential Evolution: In Search of Solutions*. Springer, New York.

Germann, U.(2003). Greedy decoding for statistical machine translation in almost linear time. In: *Proceedings of HLT-NAACL.*

De Gispert, A., Gupta, D., Popović, M., Lambert, P. Mariño, J.B., Federico, M., Ney, H., Banchs, R. (2006). Improving statistical word alignments with morpho-syntactic transformations. In: *Proceedings of FinTAL – 5th International Conference on Natural Language Processing*. Turku, Finland. LNCS, Springer, pp. 368–379.

Goldwater, S., McClosky, D. (2005). Improving statistical MT through morphological analysis. In: *Proceedings of the Conference on EMNLP*. Vancouver, Canada, pp. 676–683.

Krilavicius, T., Zilinskas, A. (2008). On structural analysis of parliamentarian voting data. *Informatica*, 19(3), 377–390.

Lee, Y.-S. (2004). Morphological analysis for statistical machine translation. In: *HLT-NAACL 2004*: *Short Papers*. Saarbrucken, Germany, pp. 57–60.

Maučec, M. S., Rotovnik, T., Zemljak, M. (2003). Modelling highly inflected Slovenian language. *International Journal of Speech Technology*, 6, 245–257.

Maučec, M.S., Brest, J., Kačič, Z. (2006). Slovenian to English machine translation using corpora of different sizes and morpho-syntactic information. In: *Proceedings of the 9th International Multiconference Information Society IS*, pp. 222–225.

Maučec, M.S., Rotovnik, T., Kačič, Z., Brest, J. (2009). Using data-driven subword units in language model of highly inflective Slovenian language. *Int. J. Pattern Recogn. Artif. Intell.*, 23(2), 287–312.

Niessen, S., Ney, H. (2000). Improving SMT quality with morpho-syntactic analysis. In: *Proceedings of the 20th International Conference on Computational Linguistics*. Saarbrucken, Germany, pp. 1081–1085.

Niessen, S., Ney, H. (2004). Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2), 181–204.

Och, F.J., Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), 19–51.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In: *Proceedings of ACL*, pp. 48–54.

Popović, M., Ney, H. (2004a). Improving word alignment quality using morpho-syntactic information. In: *Proceedings of 20th International Conference on Computational Linguistics* (*CoLing*). Geneva, Switzerland, pp. 310–314.

Popović, M., Ney, H. (2004b). Towards the use of word stems and suffixes for statistical machine translation. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation* (*LREC*). Lisbon, Portugal, pp. 1585–1588.

Price, K.V., Storn, R.M., Lampinen, J.A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Springer.

Rosenfeld, R. (1995). The CMU statistical language modeling toolkit, and its use in the 1994 ARPA CSR evaluation. In: *Proceedings ARPA SLT Workshop*. Austin.

Rotovnik, T. (2006). Large vocabulary continuous speech recognition with subword units stem-ending. In: *Proceedings Advances in Speech Technology*. Maribor, Slovenia, pp. 147–157.

Serbian, G., Câmpa, A. (2008). Hierarchical adaptive clustering. *Informatica*, 19(1), 101–112.

Storn, R., Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.

Šveikauskiene, D. (2005). Graph representation of the syntactic structure of the Lithuanian sentence. *Informatica*, 16(3), 407–418.

Tvrdík, J. (2006). Competitive differential evolution. In: *MENDEL'06, 12th International Conference on Soft Computing*, Vol. 1, pp. 7–12.

Vaiciunas, A., Raskinis, G. (2006). Cache-based statistical language models of English and highly inflected Lithuanian. *Informatica*, 17(1), 111–124.

Vogel, S., Zhang, Y., Huang, F., Tribble, A., Venugopal, A., Zhao, B., Waibel, A. (2003). The CMU statistical machine translation system. In: *Proceedings of the Machine Translation Summit IX*, Vol. 29. New Orleans, USA, pp. 7–12.

Žgank, A., Kačič, Z., Horvat, B. (2001). Large vocabulary continuous speech recognizer for Slovenian language. In: *Text, Speech and Dialogue 2001. Lecture Notes in Artificial Intelligence*, Vol. 2166. Springer, Berlin, pp. 242–248.

M. Sepesy Maučec received her BS and PhD degree in computer science from the Faculty of Electrical Engineering and Computer Science at the University of Maribor in 1996 and 2001, respectively. She is currently a researcher at the same faculty. Her research interests include language modelling, statistical machine translation and computational linguistics.

J. Brest received BS, MSc, and PhD degrees in computer science from the University of Maribor, Maribor, Slovenia, in 1995, 1998, and 2001, respectively. He is currently an associate professor at the Faculty of Electrical Engineering and Computer Science, University of Maribor. His research interests include evolutionary computing, artificial intelligence, and optimization. His fields of expertise embrace also programming languages, web oriented programming, parallel and distributed computing research. Dr. Brest is a member of IEEE, and ACM.

Morfologinių-sintaksinių požymių sumažinimas labai kaitomos kalbos statistiniame mašininiame vertime

Mirjam SEPESY MAUČEC, Janez BREST

Mes nagrinėjame statistinio mašininio vertimo iš labai kaitomos kalbos į mažiau kaitomą problemą. Kaitomų kalbų charakteristikos paprastai nėra naudojamos statistinio mašininio vertimo sistemoje. Egzistuojančios vertimo sistemos dažnai traktuoja skirtingas kaitomo žodžio formas, priklausančias tai pačiai lemai (lema – pagrindinė žodžio forma) kaip tarpusavyje nepriklausomas, nors tam tikra tarpusavio priklausomybė egzistuoja. Iš kitos pusės, mes žinome, kad jeigu mes suvedame kaitomas žodžio formas į bendras lemas, dalis informacijos yra prarandama. Būtų prasminga eliminuoti tik tas variacijas kaitomų žodžių formose, kurios nėra svarbios vertimui. Žodžių kaitomi požymiai yra apibrėžiami naudojant morfologinių-sintaksinių aprašų žymes (MSD) ir mes norime sumažinti jų skaičių. Kad tai galėtume padaryti, reikia detalaus abiejų kalbų (šaltinio ir tikslo kalbų) žinojimo. Straipsnio idėja yra rasti informaciją saugančias morfologinių-sintaksinių aprašų žymes šaltinio kalboje naudojant duomenimis grįstą metodą. Tikslas yra pasiekiamas naudojant globalios optimizacijos algoritmą, vadinamą diferencine evoliucija. Eksperimentai buvo atlikti naudojant laisvai prieinamą lygiagretų anglų–slovėnų kalbų tekstyną SVEZ-IJS, kuris yra lemuotas ir anuotuotas naudojant MSD žymes. Eksperimentų rezultatai rodo teikiančią vilčių kryptį link optimalaus morfologinių-sintaksinių požymių poaibio.