Recent Advances in Intelligent Information Systems
ISBN 978-83-60434-59-8, pages 219–232

Shallow Transfer Between Slavic Languages

Petr Homola[1], Vladislav Kuboň[1], and Jernej Vičič[2]

[1] Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic

[2] University of Primorska, Koper, Slovenia

Abstract

This paper describes an architecture of a machine translation system designed primarily for Slavic languages. The architecture is based upon a shallow transfer module and a stochastic ranker. The shallow transfer module helps to resolve the problems, which arise even in the translation of related languages, the stochastic ranker then chooses the best translation out of a set provided by a shallow transfer. The results of the evaluation support the claim that both modules newly introduced into the system result in an improvement of the translation quality.

1 Introduction

The demand for translation of various kinds of documents seems to be growing constantly in recent years. Although the general quality of automatic translation systems is far from from at least acceptable level, it makes sense to try to develop new approaches and methods, which can be used either in particular context (limited thematic domain) or for a particular language pair. The machine translation (MT) of related languages seems to be one of the areas where relatively simple (or simplified) methods may bring some success.

One of the methods, which guarantees relatively good results for the translation of closely related languages is the method of a rule-based shallow-transfer approach. It has a long tradition and it had been successfully used in a number of MT systems, the most notable of which is probably Apertium (Corbi-Bellot *et al.*, 2005).

Shallow-transfer systems usually use a relatively linear and straightforward architecture where the analysis of a source language is usually limited to the morphemic level. The architecture usually exploits a morphological disambiguator (tagger), which precedes any kind of more or less deterministic transfer phase. This is obviously a huge limitation, especially for lexical transfer, since in most language pairs there are many words whose translation depends upon the syntactic and/or semantic context. If the system contains some (shallow) syntactic parser and/or structural transfer, they also tend to produce ambiguous output relatively often.

Although the description of a shallow parsing module is the most important part of this paper, we have decided to complement its description by a description of an improvement of the artchitecture of a typical shallow-transfer MT system.

The main reason behind this decision is the fact that the new architecture influences the function of the transfer module to a great extent. The transfer module deals with an ambiguous input in the new architecture and as a consequence it also provides a wider variety of the output variants, which are later resolved by a stochastic ranker.

The paper is organized as follows: Section 2 contains a brief description of related research. In Section 3, we describe a modification of the commonly used shallow-transfer approach that leads to higher translation quality. In Section 4, we explain the implementation of the transfer. Section 5 describes the evaluation our MT experiments and finally, we conclude in Section 6.

2 An overview of MT systems between related languages

MT between closely related languages has a long tradition and it has experienced a rebirth in the last decade. The first experiments were done for Slavic and Scandinavian languages. The shallow-transfer approach has been shown to give viable results for related languages with very rich inflection as well as for analytical and agglutinative languages. We give a brief overview of several systems in the following sections.

2.1 Slavic languages

2.1.1 RUSLAN

Probably the first MT system for the translation between closely related Slavic languages was RUSLAN (Hajič, 1987; Bémová *et al.*, 1988), translating from Czech into Russian. The system aimed at the translation in a limited domain of manuals of operating systems of mainframes. The authors deliberately ignored the transfer in the initial implementation phases, the last phase of a deep syntactic analysis of Czech was immediatelly followed by a phase of syntactic synthesis of Russian. The reason for this strategy was obviuous, the close relatedness of both languages and the limited translation domain (explanatory technical texts) was supposed to decrease the number of conventional transfer problems encountered in MT of non-related languages.

The strategy of the minimal transfer had to be abandoned in the later stages of the implementation. Even technical texts being translated between two closely related languages required a specific solution of problems as, e.g., a Czech present tense auxiliary *jsem* "I am" which is not used in Russian or verbal negation which is in Czech formed by an inseparable prefix *ne-*, while the corresponding negative particle is typically separated from the verb in Russian, etc. The experience gained in RUSLAN clearly shows that even for closely related languages it is necessary to include some kind of a transfer module, probably not as complex as in the case of two unrelated languages.

2.1.2 Česílko

An MT system from Czech into Slovak is described in Hajič *et al.* (2000). As there are practically no syntactic nor semantic differences between the two languages, the

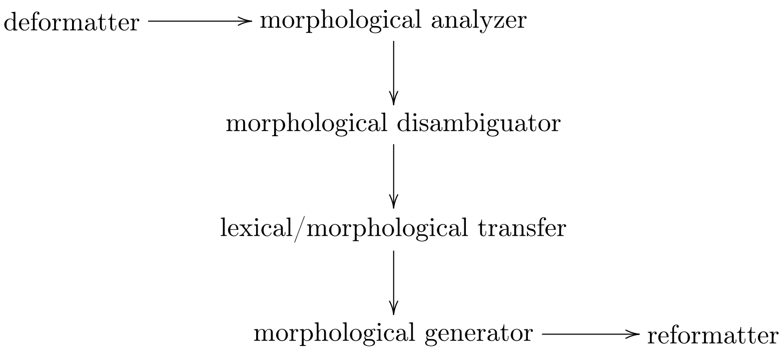


Figure 1: Architecture of the first version of the system Česílko

system uses a direct lemma-to-lemma lexical transfer with a one-to-one dictionary.

Later, the system was adapted to the language pair Czech-Polish Dębowski *et al.* (2002) and finally, the shallow-transfer approach has been suggested and implemented by Hajič *et al.* (2003) after experiments with translation from Czech into Lithuanian.

The MT system *Česílko* originally was an experimental system for automatic translation as a supporting module for pre-filled translation memories. Since the source and target language of the system were closely related, the system did not perform any syntactic analysis but it translated the input text on a lemma-to-lemma and tag-to-tag basis. The system consisted solely of the following modules (we have reused some of them in our experiments):

1. morphological tagger for Czech
2. bilingual glossaries
3. morphological synthesis for Slovak or Polish.

Czech is a language with rich inflection, i.e., a word usually has many different endings that express various morphological categories. The morphological analyzer assigns a set of lemmas and tags to each word. As it was necessary to have only one tag for each word determined by the context of the sentence, a statistical tagger was used with an accuracy of approx. 94% (see Hajič and Kuboň (2003)). The use of the tagger was necessary since the input of the lexical transfer (which was the immediately following module) was expected to be disambiguated.

The final phase generates word forms in the target language, which is comparatively simple.

2.1.3 GUAT

An MT system from Slovenian into Serbian, based on Apertium, has been experimentally implemented by Vičič (2008) (the architecture of the framework is described in Section 2.5.1). The system utilizes the available Slovenian morphological analyzer. The other linguistic resources were built automatically by exploiting available corpora for both languages. Even transfer rules are intended to be induced automatically in the future versions of the system. Currently, there are only a few hand-written rules.

2.2 Scandinavian languages

2.2.1 PONS

There has been an extensive research in MT between various Scandinavian languages. The first extensive experiment was the PONS (Partiell Oversettelse mellom Nærstående Språk = Partial translation betweenclosely related languages) system (Dyvik (1995)) that translated from Norwegian into Swedish. The authors argue that if two languages are close enough, it is mostly not necessary to "waste time finding a lot of redundant grammatical and semantic information about the expressions". They suggest that for closely related languages, one should choose a different strategy than for distant languages. Concretely for Scandinavian languages, "formal equivalence will often imply denotational and stylistic equivalence". The general principle is to use as much of the structure of the source sentence as possible "within the limits imposed by idiomacity". In particular, semantic and stylistic properties of translated sentences are not taken into account, relying on the closeness of both languages at the corresponding levels, since "in closely related languages, similar effect can be achieved with similar means". The source sentence serves as a template for the encoding of the target sentence.

Besides Norwegian-to-Swedish, the system has also been tested for English and Norwegian.

2.2.2 Norwegian-Danish

A similar approach was used in the MT system from Norwegian (bokmål) into English that used Danish as an interlingua Bick and Nygaard (2007). As there are almost no syntactic differences between these two Scandinavian languages, and there is a widely corresponding polysemy, they generate the Danish translation from the output of a Norwegian tagger by substituting lemmas using a one-to-one dictionary. The output of a newly constructed Norwegian-to-Danish MT system is piped into an existing Danish parser and further processed. This approach exploits the fact that "the polysemy spectrum of many Bokmål words closely matches the semantics of the corresponding Danish word, so different English translation equivalents can be chosen using Danish context-based discriminators".

The first step in the system is disambiguation of lemmas and PoS tagging. After the tagger, Norwegian lemmas are substituted by Danish ones. Additionally, there is a special handling of compound nouns based on partial translation of words. The morphology of the two languages is not completely isomorphic and there are also some structural differences that are handled by a Karlsson's Constraint Grammar (for example, double definiteness in Norwegian, which is solved by substitution rules).

2.3 Turkic languages

For Turkic languages, an experimental MT system from Turkish into Crimean Tatar has been implemented Altintas and Cicekli (2002). They claim that for languages with shared historical background and similar culture, there is no need for a semantic analyzer. As most parts of the grammar are common in both languages, the system focuses on differences at the morphemic level, thus translation from Turkish into Crimean Tatar is basically "disambiguated word-for-word translation".

For the implemented language pair, there are several categories of transfer rules. The rules can generally be applied in any order, except for the rules that change the root. The system is implemented using finite-state tools with an interface written in Java. The system outputs all possible results of rule application and lexical ambiguities.

2.4 Celtic languages

An MT system between Irish and Scottish Gaelic (both Insular Celtic/Goidelic languages) is presented in Scannell (2006). Both languages are not mutually intelligible, at least in their spoken variant, but their grammars are very close since they have a common ancestor — Middle Irish, and a shared literary tradition written in the so-called Classical Gaelic (Gaeilge Chlasaiceach) up through the 18th century. The aim of the system is information retrieval for all Goidelic languages.

It is noteworthy that the input is normalized before being translated since the orthography of processed texts may differ. The differences between the two languages are comparatively small, thus chunking is believed to be sufficient instead of parsing in most cases. Due to the syntactic closeness of both languages, the biggest translation problem occurs at the semantic level; therefore, a word sense disambiguation is an integral part of the system.

Syntactic transfer is a necessary part of the system due to periphrastic constructions, which are present only in one language. The rules are transformed into a finite state recognizer, which can be compiled for fast matching against the tagged and chunked input stream. In the current version, there are less than 100 transfer rules. Their number is expected to grow rapidly as new rules for handling additional multiword expressions will be added.

The prevalent part (90%) of the lexicon has been extracted automatically from two electronic dictionaries — Irish-English and Scottish-English.

Finally, there is a post-processing phase performing local corrections (such as incorrect initial mutation), which is based on the Gramadóir grammar checker.

2.5 Romance languages

2.5.1 Apertium

For the Romance languages of Spain, the system Apertium has been implemented Corbi-Bellot *et al.* (2005). The system is largely based on the older MT systems interNOSTRUM Forcada *et al.* (2001) and Tradutor Universia[1]. The authors

[1] http://tradutor.universia.net

claim that a word-to-word translation may give an adequate translation of 75% of the text. The system uses the shallow-transfer approach. Open source data are available for a number of language pairs.

The system actually uses the same architecture as the older system Česílko, with only one added module, a post-generator, which adapts the surface representation of the translation in the target language, e.g., *me* "to me" and *o* "it/him" is in Portuguese contracted to *mo*, etc.

It is also claimed that this architecture be suitable even for pairs of distant languages, such as Spanish-Basque, which is an intended language pair to be implemented within Apertium. For this language pair, a deeper-transfer architecture is being designed.

As the main source of translation errors is morphological ambiguity, a tagger has been prepended before the transfer. The dictionaries contain single equivalents as well as multiword expressions. Transfer rules, which handle, for example, the rearrangement of clitic pronouns, have the form pattern-action and there are approx. 90 of them. The system is able to process about 5,000 words per second.

MT from Portuguese into Spanish within Apertium is presented in Armentano-Oller *et al.* (2006). The system is able to recognize 9,700 Protuguese lemmas and to generate the same amount of Spanish lemmas. The bilingual dictionary contains 9,100 lemma-to-lemma pairs.

3 Increasing the accuracy of the shallow-transfer approach

As has been already mentioned, the statistical tagger used to disambiguate the input text at the beginning of the translation process introduces too many errors into the processed data. For example, the taggers used in Apertium for Romance languages have accuracy of approx. 96% Corbi-Bellot *et al.* (2005), thus the error rate is too high and it checkmates the subsequent modules since they get incorrect data. The accuracy of best available taggers for Czech (as a main source language used in Česílko) reaches about the same level. Although these numbers may seem relatively good (although not as good as the results of taggers for less inflected languages), they in fact mean that every 25th input word has an icorrect tag, or, in other words, approximately every second input sentence contains an incorrect tag. These tags may then cause additional errors in the phase of morphological synthesis of the target language.

Unfortunately, the only way to avoid these errors is to omit the tagger from the system and work with ambiguous input. Obviously, the exclusion of the tagger from the system has to be compensated somewhere else in the translation process.

Let us have a look at an example. We would like to translate the following Czech phrase into Slovak:

(1)	*auta*	*jezdila*
	cars-NEUT,NOM,PL	went-PAST,NEUT,PL.

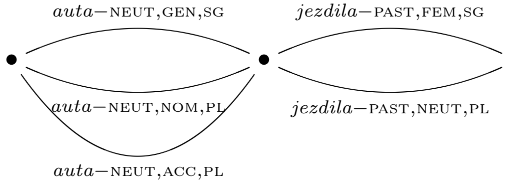
"the cars moved"

If we used a tagger, and if its results were correct, the output would be as follows:

(2) *auta*-NEUT,NOM,PL *jezdila*-PAST,NEUT,PL

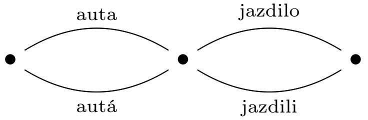
and a word-to-word translation into Slovak would give a correct translation. However, both words are morphologically ambiguous and if we omit the tagger, each input word form would split in several morphologically distinct lemma-tag pairs. For example, some Czech adjectival word forms can have up to 27 distinct morphological meanings. The following structure would be the input of the subsequent modules:

(3)



Without a parser or another module, which would resolve the ambiguity, the system would output the following Slovak representation after the morphological synthesis:

(4)



We see that two edges have been merged into one due to morphological syncretism but there are still four possible outputs if one would consider all paths through the multigraph from the initial node to the end node.

We decided to add a module to the system that would find the 'best' path through the multigraph. We suggest to use a language model for the target language. In our experiments, a trigram model based on word forms and trained on about 20 million words from the Wikipedia has been used.

In the resulting Slovak representation (in the above example), the correct path through the multigraph would be found correctly. Nevertheless, there is another problem — for longer sentences, this approach leads to a combinatorial explosion. Fortunately, the solution is comparatively simple: we have added a non-deterministic partial parser based on LFG Bresnan (2002) and our experiments show that even if we parse only prepositional and noun phrases, the morphological ambiguity gets reduced significantly even for languages with rich inflection, such as Czech. Syntactic analysis is needed anyway to mark local dependencies that will be used in the structural transfer. The improved architecture is given in Figure 2.

4 Transfer

Transfer and syntactic synthesis are performed jointly in one module. The task of the transfer module is to adapt complex structures created by the parser, which

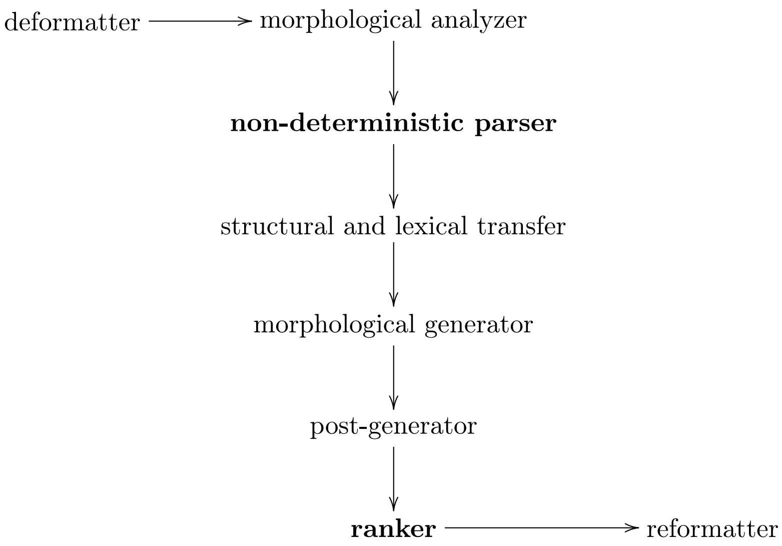


Figure 2: Improved shallow-transfer approach

cover the whole source sentence continuously to the target language lexically, morphologically and syntactically. In the following sections, we describe the phase of the lexical transfer and the structural transfer, the latter being split further in structural preprocessor and syntactic decomposer.

4.1 Lexical transfer

The aim of the lexical transfer is to 'translate a feature structure lexically', i.e., the lemmas associated with features structures are translated. Morphological features may be adapted as well where appropriate.

The following is a fragment of the dictionary used in lexical transfer (between Czech and Slovenian):

(5)
```
hvězda|zvezda
dodat|dodati
kůň|konj
strom|drevo|gender=neut;
```

Let us have a brief look on the last line of the example. The Czech noun *strom* "tree" is masculine while its Slovenian counterpart *drevo* is neuter, that is why there is the additional information *gender=neut*, which instructs the transfer module to adapt the feature *gender* of the corresponding feature structure so it can be correctly synthesized morphologically.

4.2 Structural transfer

The task of the structural transfer is to adapt the feature structures of the source language (their properties and mutual relationship) so that the synthesis generates a grammatically well-formed sentence with the meaning of the source sentence. It is to note that the well-formedness can generally be guaranteed only locally for the part of the sentence the feature structure covers (this is one flaw of shallow parsing).

When changing the structure, one may do one of the following:

1. Change values of atomic features in the feature structure, add atomic features with a specific value or delete some atomic features.
2. Add a node to the syntactic tree.
3. Remove a node from the syntactic tree.

There are two types of structural changes:

Preprocessing of feature structures Such changes are performed prior to the lexical transfer.

Decomposition of feature structures These changes are performed after the lexical transfer and build up the syntactic synthesis.

We give a couple of examples of transfer rules. The formal language of the rules is relatively transparent, let us only explain the role of some of the attributes.

The following rule is used to translate a preposition (the presence of a preposition depending on a noun is indicated by *hasChildren (prep)*), which requires a different case in the target language (the requirement for a specific preposition and a case is located in the *lexChild* attribute). In the feature structure of the noun that governs the preposition, its case is changed to the correct one (by copying the case required by the target language preposition to its governing noun by means of an attribute *(copyup (case))*.

```
(
preproc
(head= ((type word) (pos n)))
(hasChildren (prep))
(child= ((type word) (lemma u-1) (case gen)))
(lexChild ((lemma pri) (case loc)))
(copyup (case))
)
```

The following rule adds an auxiliary (by means of creating a new child node *newChild* in the target language) to an *l*-participle in the third person (the fact that there is no auxiliary present in the source language is marked by *noChildren (aux)*), which may be required, for example, when translation from Czech to Slovenian. The attribute *(relorder -9)* indicates that the new child should be inserted to the leftmost position in the subtree of the verb.

```
(
preproc
(head= ((type word) (pos verb) (vform lpart) (person 3)
   (number $number)))
(noChildren (aux))
(newChild ((gfunc aux) (relorder -9) (lemma být)
   (pos verb) (vform fin) (tense pres) (person 3)
   (number $number)))
)
```

The following rule removes an auxiliary (the presence of the auxiliary is indicated by *hasChildren (aux)*) from an *l*-participle in the third person, which may be required, for example, when translation from Slovenian to Czech. The removal is indicated by *removeChild 1*.

```
(
preproc
(head= ((type word) (pos verb) (vform lpart) (person 3)))
(hasChildren (aux))
(removeChild 1)
)
```

The following rule rewrites the features gender, case and number of an adjective, which is being detached by values of these features from the governing noun to preserve agreement between an adjectival attribute and a noun. Unlike the previous examples, this rule is applied after the transfer, during the syntactic synthesis in the target language. The actual rewriting of the features mentioned above is done by copying the values from the governing node to the dependent one (*copydown (gender case number)*).

```
(
decomp
(recursive 1)
(head= ((type word) (pos n)))
(child= ((type word) (pos a)))
(copydown (gender case number))
)
```

An example of this rule's use would be the translation of the phrase *velký strom* "big tree" (Cze) into Macedonian *големо дрво* where the gender has changed from masculine to neuter. Without this transfer rule, we would get **голем дрво*

The following rule changes the infinitive to an *l*-participle in periphrastic future tense constructions as required, for example, when translating from Czech to Slovenian. The rewriting is indicated by the command *rewriteHead* specifying the attributes, which should be rewritten.

```
(
decomp
(head= ((type word) (pos verb) (vform inf)))
(child= ((type word) (lemma být) (vform fin)
   (tense fut) (gender $gender) (number $number)))
(rewriteHead ((vform lpart) (gender $gender)
   (number $number)))
)
```

A similar rule operating on VPs would be used, for example, when translating the Czech verb phrase *napsal jsem* "I wrote/I have written" to its Macedonian counterpart (*написав/имам написано*) since a word-for-word translation would give *написал сум*, which would be well-formed with different word order (*сум написал*) but still semantically different (renarrative).

4.3 Translation of multiword expressions

It is an obvious fact that some words of the source language are translated as multiword expressions in the target language and vice versa, for example:

(6) *babička* "grandmother" (Cze) → *stará mama* (Slv)
zahradní jahoda "garden strawberry" (Cze) → *truskawka* (Pol)

Since these cases require the removal or addition of a subordinated feature structure (for the adjective), which is equivalent to removing or adding a node from/to the syntactic tree, such cases are handled by special rules in the structural transfer.

5 Evaluation

Although BLEU (Papineni *et al.* (2001)) and NIST (Doddington (2002)) metrics became almost a standard in recent years, we have decided to use a different metric for the evaluation of our system. There were several reasons for this decision, the criticism of BLEU presented recently in a number of articles (e.g. Callison-Burch *et al.* (2006)) being only one of them.

More important reason why we have rejected BLEU is the insensitivity of these strongly n-gram oriented metrics to inflection. A small variation of a word form used in a target sentence will usually not negatively affect the understability of the whole sentence (although it might affect its syntactic correctness), but it will have a dire effect on the number of correct n-grams. Also the effort needed for post-editing of such an error is much smaller than if it is a real translation error (wrong lexical unit, syn tactically incorrect construction, etc.). Actually, the fact that the BLEU or NIST score does not have any real meaning with regard to the complexity of the post-editing of the MT output constitutes an additional reason why to use a different, more practically oriented metric.

Last but not least reason for exploiting a different metric is the lack of multiple references - in our experiments we usually have only a single reference translation and it is a well-known fact that the BLEU score is much reliable if multiple references are available. The lack of additional reference translations actually means that it is not possible to take into account a variation of a word-order (if there is only a single reference translation than it is not possible to take into account any other order of words than the one from the single reference), a fault very important for the translation between languages which have a very high degree of word-order freedom.

The above mentioned reasons led us to the exploitation of a metric which is simple, traditional and which correlates very well with the amount of post-editing work required after the automatic translation. The metrics we are using is the Levenshtein edit distance between the automatic translation and a reference translation. The test data for the Czech-to-Slovak experiments consist of 400 mainly newspaper sentences.

The metric works as follows:

There are three basic possibilities of the outcome of translation of a sentence:

1. The rule-based part of the system has generated a 'perfect'[2] translation (among other hypotheses) and the ranker has chosen this one.
2. The rule-based part of the system has generated a 'perfect' translation but the ranker has chosen another one.
3. All translations generated by the rule-based part of the system need post-processing.

In the first case, the edit distance is zero, resulting in accuracy equal to 1. In the second case, the accuracy is $1 - d$ with d meaning the edit distance between the segment chosen by the ranker and the correct translation divided by the length of the segment. In the third case, the accuracy is calculated as for 2 except that we use the reference translation to obtain the edit distance.

Given accuracies for all sentences we use the arithmetic mean as the translation accuracy of the whole text. The accuracy is negatively influenced by several aspects. If a word is not known to the morphological analyzer, it does not get any morphological information, which means that it is practically unusable in the parser. Another possible problem is that a lemma is not found in the dictionary. This does not happen very often due to the fact that we use the best available morphological analyzer for Czech, which is able to process 800 000 lemmas, but not even a dictionary of such a size has a complete coverage of all words used in specific domains. If the lemma of the word is not present in the dictionary of the analyzer, the original source form appears in the translation, which of course penalizes the score. Finally, sometimes the morphological synthesis component is not able to generate the proper word form in the target language (due to partial incompatibility of tagsets for both languages). In such a case, the target lemma appers in the translation.

The results are summarized in Table 1 and Table 2. The baseline system (called ”no transfer” in the table, although it contains a module of a lexical transfer) is the original system Česílko introduced in the Chapter 2. As can be seen, the improvement is very low for the language pair Czech-Slovak, which indicates that virtually no structural transfer is needed here. For Czech-to-Slovenian, on the other hand, the improvement is significant.

[2] By 'perfect' we mean that the result does not need any human post-processing.

	no transfer	shallow transfer
accuracy (character based)	96.35%	96.39%
accuracy (word based)	88.13%	88.24%

Table 1: Czech-to-Slovak evaluation

	no transfer	shallow transfer
accuracy (character based)	74.67%	80.43%
accuracy (word based)	65.52%	71.78%

Table 2: Czech-to-Slovenian evaluation

6 Conclusions

The work described in this paper is a part of a research devoted to an endeavour to find a proper level of transfer between related languages. The experience from the previous experiments clearly indicates that the more closely related the languages are, the more shallow transfer they require. This paper mentions one more aspect of the problem - the architecture of such a system. According to the results obtained in our experiments, it is worthwhile to preserve a certain level of ambiguity during transfer and to resolve it in the later stages by a stochastic ranker. The most natural next step in the research would be an examination how a more complicated statistical language model for the target language will influence the quality of the shallow-transfer approach.

Acknowledgments

The presented research has been supported by the grant No. 1ET100300517 of the GAAV ČR.

References

Kemal Altintas and Ilyas Cicekli (2002), A Machine Translation System between a Pair of Closely Related Languages, in *Proceedings of the 17th International Symposium on Computer and Information Sciences (ISCIS 2002)*, pp. 192–196, Orlando, Florida.

Carme Armentano-Oller, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A. Scalco (2006), Open-source Portuguese-Spanish machine translation, in *Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese*, Rio de Janeiro, Brasil.

Eckhard Bick and Lars Nygaard (2007), Using Danish as a CG Interlingua: A Wide-Coverage Norwegian-English Machine Translation System, in *Proceedings of NODALIDA*, Tartu, Estonia.

Joan Bresnan (2002), *Lexical-functional syntax*, Blackwell Textbooks in Linguistics, New York.

Alevtina Bémová, Karel Oliva, and Jarmila Panevová (1988), Some Problems of Machine Translation Between Closely Related Languages, in *Proceedings of the 12th conference on Computational linguistics*, volume 1, pp. 46–48, Budapest, Hungary.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn (2006), Re-evaluating the Role of BLEU in Machine Translation Research, in *Proceedings of the EACL'06*, Trento, Italy.

Antonio Corbi-Bellot, Mikel Forcada, Sergio Prtiz-Rojas, Juan Antonie Perez/Ortiz, Gema Remirez-Sanchez, Felipe Sanchez Martinez, Inaki Alegria, Aingeru Mayor, and Kepa Sarasola (2005), An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain, in *Proceedings of the 10th Conference of the European Association for Machine Translation*, Budapest.

Łukasz Dębowski, Jan Hajič, and Vladislav Kuboň (2002), Testing the limits — adding a new language to an MT system, *Prague Bulletin of Mathematical Linguistics*, 78.

George Doddington (2002), Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, in *Proceedings of the ARPA Workshop on Human Language Technology.*

Helge Dyvik (1995), Exploiting Structural Similarities in Machine Translation, *Computers and Humanities*, 28:225–245.

Mikel Forcada, Antonio Garrido, Raul Canals, Amaia Iturraspe, Sandra Montserrat-Buendia, Anna Esteve, Sergio Ortiz Rojas, Herminia Pastor, and Pedro Pérez (2001), The Spanish-Catalan machine translation system interNOSTRUM, *0922-6567 - Machine Translation*, VIII:73–76.

Jan Hajič (1987), An MT System Between Closely Related Languages, in *Proceedings of the third conference of the European Chapter of the Association for Computational Linguistics*, pp. 113–117, Copenhagen, Denmark.

Jan Hajič and Vladislav Kuboň (2003), Tagging as a Key to Successful MT, in *Proceedings of the Malý informatický seminář*, Josefův Důl.

Jan Hajič, Jan Hric, and Vladislav Kuboň (2000), Machine translation of very close languages, in *Proceedings of the 6th Applied Natural Language Processing Conference*, pp. 7–12, Seattle, Washington, USA.

Jan Hajič, Petr Homola, and Vladislav Kuboň (2003), A simple multilingual machine translation system, in *Proceedings of the MT Summit IX*, New Orleans.

Lauri Kartunnen (1986), D-PATR: A development environment for Unification-based Grammars, in *Proceedings of Coling*, pp. 74–80.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2001), BLEU: a method for automatic evaluation of machine translation, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania.

Kevin P. Scannell (2006), Machine translation for closely related language pairs, in *Proceedings of the Workshop Strategies for developing machine translation for minority languages*, Genoa, Italy.

Jernej Vičič (2008), Rapid development of data for shallow transfer RBMT translation systems for highly inflective languages, in *Proceedings of 6th Language Technologies Conference 2008.*