A Simple Multilingual Machine Translation System

Jan Hajič, Petr Homola, Vladislav Kuboň

Inst. of Formal and Applied Linguistics, Charles University
Prague, Czech Republic
{hajic,homola,vk}@ufal.mff.cuni.cz

Abstract

The multilingual machine translation system described in the first part of this paper demonstrates that the translation memory (TM) can be used in a creative way for making the translation process more automatic (in a way which in fact does not depend on the languages used). The MT system is based upon exploitation of syntactic similarities between more or less related natural languages. It currently covers the translation from Czech to Slovak, Polish and Lithuanian. The second part of the paper also shows that one of the most popular TM based commercial systems, TRADOS, can be used not only for the translation itself, but also for a relatively fast and natural method of evaluation of the translation quality of MT systems.

1 Introduction

One of the most widely used techniques for machine-aided human translation of the last decade is without doubts a method of human translation supported by a translation memory. This technique can substantially speed up the translation process especially when dealing with large amount of repeated translations, e.g. in the area of localization of various kinds of technical documentation.

The multilingual machine translation system described in the first part of this paper demonstrates that the translation memory (TM) can be used in a creative way for making the translation process more automatic (in a way which in fact does not depend on the languages used). The second part of the paper also shows that one of the most popular TM based commercial systems, TRADOS, can be used not only for the translation itself, but also for the evaluation of translation quality. The evaluation method proposed here is very simple and fast and thus it can be used especially for the routine evaluation of improvements during the development and debugging of the system, while still being very close to the evaluation used in commercial translation bureaus

2 The use of the translation memory in our system

The process of localization of technical texts (manuals, software documentation etc.) is a relatively specific area of translation. Typically, there is only a single source language and many target languages, but in most cases the localization into any of these languages is being performed separately even though many of the target languages require solving similar problems during the translation.

It is quite clear that the independent localization of the same document into several typologically similar target languages is a waste of effort and money, but it seems that only a little effort has been devoted recently to the solution of this problem. One of the probable reasons might be the fact that especially for "small" languages the producers of source texts have localization vendors for each "small" language in each of the countries speaking that particular language (although there are some exceptions to this rule). The local vendors then do not have access to the localized texts produced by some other vendor in a different country and thus they cannot exploit it for solving their (similar or even identical) localization problems.

2.1 Use of a pivot

The use of one language from the target group as a pivot and to perform the translation through this language seems to be a quite natural solution for these problems. It is of course much easier to translate texts from Czech to Polish or from Russian to Serbian than from English or German to any of these languages. It is of course true that applying the pivot language approach has a serious drawback - the translation quality, which needs to be very high, may deteriorate in this two-step process. A negligible shift of the meaning during the translation into a pivot language may be amplified by a subsequent translation from the pivot language to the actual target language.

We hope that the approach proposed in the following parts of this paper, which combines the human translation into a pivot language with the machine translation among the typologically similar target languages and which to a great extent exploits the potential offered by current translation memory based systems, represents a solution which overcomes most of the problems of the pivot language translation model.

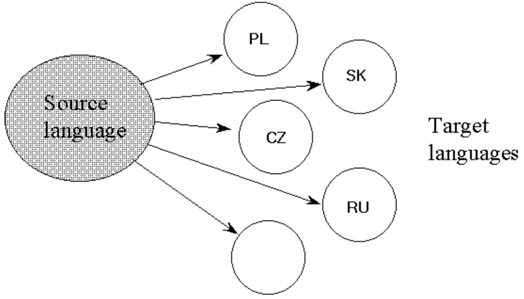

Fig.1 A traditional model of localization

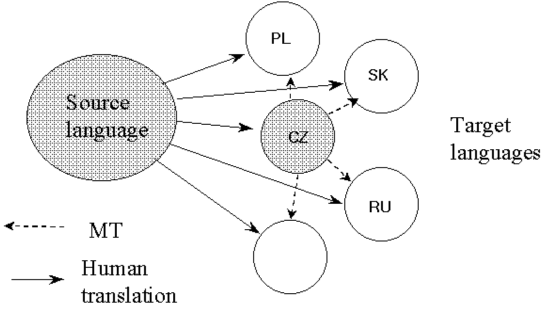

Fig.2 Our model based on a pivot language

2.2 The role of a translation memory in our system

The concept of a combination of a fully automatic machine translation system with a translation memory plays a key role in our system (although our core machine translation module can be used also as a standalone application). As a representative of the class of currently available commercial translation memory-based systems we have chosen the system TRADOS Translator's Workbench, an example-based translation tool, which currently seems to be a market leader in the category of professional translation tools. The translation memory contains pairs of previously translated sentences. When a human translator starts translating a new sentence, the system tries to match (at least partially) the source language sentences (more precisely, segments - although most of the segments are sentences, there are several exceptions) with sentences (segments) already stored in the translation memory. If it is successful, it suggests the translation and the human translator decides whether to use it, to modify it or to reject it.

The segmentation of the translation memory (the texts are stored as relevant pairs of source/target language segments) is the key feature of our method. The TRADOS translation memory may be exported into a text file and thus allows for an easy manipulation with its content; as we show below, it is the (reverse) translation memory import function that is the key to our method.

The first step of our method is the human made translation from the source into the pivot language. At the end of the translation process we have at our disposal a translation memory covering all sentences of the source text. After the export of the translation memory into a text file each bilingual segment in the exported translation memory looks as follows (English is the source language, Czech is a pivot):

```
<TrU>
<CrD>08052003, 12:47:55
<CrU>HT
<Seg L=EN_US>Choose Macro from the
Tools menu.
<Seg L=CS_01>V nabídce Nástroje
zadejte příkaz Makro.
</TrU>
```

A very simple transformation of the exported translation memory will lead us to the source format for the machine translation module (the following sample represents a translation unit of a new, half-empty translation memory for the Czech - Slovak translation):

```
<TrU>
<CrD>08052003, 12:47:55
<CrU>HT
<Seg L=CS_01>V nabídce Nástroje
zadejte příkaz Makro.
<Seg L=SK_01>n/a
</TrU>
```

Our translation module then extracts segment after segment, translates it into a target language (Slovak in our example) and replaces the ”n/a” string with the relevant translation equivalent (all these operations are being performed on the same translation memory):

```
<TrU>
<CrD>08052003, 12:47:55
<CrU>HT
<Seg L=CS_01>V nabídce Nástroje
zadejte příkaz Makro.
<Seg L=SK_01>V menu Nástroje
zadajte príkaz Makro.
</TrU>
```

At this moment we have at our disposal two translation memories – one is human-made for the source/pivot language pair and the other one has been created by our MT system for the pivot/target language pair. The substitution of segments of a pivot language by the segments of a target language is then only a routine procedure. Notice that also the signature of the creator of the translation segment has been changed from HT to MT; this change helps the system recognize those translation segments that had been created by the MT system:

```
<TrU>
<CrD>08052003, 12:47:55
<CrU>MT
<Seg L=EN_US>Choose Macro from the
Tools menu.
<Seg L=SK_01>V menu Nástroje
zadajte príkaz Makro.
</TrU>
```

The human translator translating from the source to the target language then gets a translation memory for the required pair (source/target); s/he might not even know there was a pivot language involved. The system of penalties applied in the TRADOS Translator’s Workbench guarantees that if there is already a human-made translation present in the memory, it gets higher priority than the translation obtained as a result of MT.

This method has at least three advantages:

- the use of machine-made translation memory only as a resource supporting the direct human translation from the source to the target language has no negative effect on the quality of translation,
- from the user’s point of view there is no difference (except for the small difference in the quality of translation memories) when our method is used compared to the original process of working with the support of solely human-made translation memories, and
- given a sufficient quality of the MT from the pivot to the target language, our method may substantially increase the speed and reduce the costs of the translation from the source to the target languages.

3 Basic properties of the system

The method described above has been applied in the system Česílko (Hajič and Kuboň, 2000) for the translation between Czech as a pivot language and Slovak as a target language. The basic premise of the system was to use as simple method of analysis and transfer as possible. The experience from an existing MT system RUSLAN (Czech-to-Russian MT system) aimed at the translation of software manuals for operating systems of mainframes – cf. (Oliva, 1989) made it apparent that a full-fledged syntactic analysis of Czech is both unnecessary and too unreliable and costly. The system Česílko therefore uses the method of direct word-for-word translation (after necessary morphological processing), the use of which is justified by the similarity (even though not identity) of syntactic constructions in both languages.

The system has been tested on texts from the domain of documentation of corporate information systems. It is, however, not limited to any specific domain; it has also undergone thorough testing on rather difficult texts of a Czech general encyclopedia, and in an cross-lingual treebank annotation transfer project. Its primary task is, however, to provide support for translation and localization of various technical texts.

3.1 The original pair of languages (Czech to Slovak)

Since Czech and Slovak have almost the same syntax, the greatest problem of the word-for-word translation approach is the problem of ambiguity of word forms. For example, in Czech there are only rare cases of part-of-speech ambiguities (*stát* [to stay/the state], *žena* [woman/chasing] or *tři* [three/rub(imper.)]), however, the ambiguity of gender, number and case is very high (for example, the form of the adjective *jarní* [spring] is 27-way ambiguous). Even though several Slavic languages have the same property as Czech, the ambiguity is not preserved at all or it is preserved only partially, it is distributed in a different manner and the ”form-for-form” translation is not applicable.

We have applied a stochastically based morphological disambiguation for Czech whose accuracy seems to be sufficient. Thus the system consists of the following steps:

1. Import of the source (Czech input) sentence (a segment from an ”empty” translation memory)
2. Morphological analysis of Czech
3. Morphological disambiguation of Czech
4. Domain-related bilingual glossaries
5. General bilingual dictionary
6. Morphological synthesis of Slovak
7. Export to the original translation memory (Slovak target sentence) with an appropriate markup.

3.1.1 Morphological analysis of Czech

The morphological analysis of Czech is based on the morphological dictionary described in (Hajič, 2001). The dictionary covers over 800,000 lemmas and it is able to recognize about 20 mil. word forms. The morphological analysis uses a system of 15 positional tags: each morphological category, such as Part of speech, Gender, Case, etc., has a fixed one-letter place in the tag.

3.1.2 Morphological disambiguation of Czech

The module of morphological disambiguation (tagging) is a key to the success of the translation. The tagging system is based on an exponential probabilistic model (Hajič and Hladká, 1998), trained on roughly one million words using the level 1 manual annotation of the Prague Dependency Treebank (Hajič, 1998 and PDT, 2001). The average accuracy of tagging is now over 95% (measured on tokens of running text). Lemmatization chooses the first lemma with a possible corresponding tag and works with accuracy close to 98%. This works well for lemma homonymy with a different part of speech, but for true polysemy resolution (word sense disambiguation for words with the same part of speech) we will have to add word sense disambiguation such as the one described in (Cikhart and Hajič, 1999).

3.1.3 Domain-related bilingual dictionaries (glossaries)

The domain-related bilingual glossaries contain pairs of individual words and pairs of multiple-word terms. The glossaries are organized into a hierarchy specified by the user; typically, the glossaries for the most specific domain are applied first. There is one general matching rule for all levels of glossaries – the longest match wins.

Currently, the system handles well *n:n* term translation (two-word terms translated at two-word terms, etc.), and uses heuristic guessing for asymmetric cases (*m:n*, i.e. when the length of the source and target term differs). More sophisticated system for handling the tags correctly in the *m:n* translation case is under development.

3.1.4 General bilingual dictionary

The main bilingual dictionary contains data necessary for the translation of both lemmas and tags. The translation of tags is necessary, because both tagsets are similar but not identical. Also, the tags do not always correspond exactly, e.g. there are some Slovak nouns that have different gender, or tags with variants that do not exist in the other language. Therefore, a Czech tag is not translated into a single tag, but into a priority-ordered list of tags.

3.1.5 Morphological synthesis of Slovak

The morphological synthesis of Slovak is based on a monolingual dictionary of Slovak, developed by J. Hric (1991-99), covering more than 100,000 lemmas. The coverage of the dictionary is still growing. It aims at a similar coverage of Slovak as has currently been achieved for Czech.

4 New language pairs

After the initial tests of the Czech-to-Slovak module the development of the system took two directions - the first one, the enrichment of all dictionaries of the system, has been inspired by the above mentioned tests on the general encyclopedia. Those tests showed that even though the morphological dictionary covers more than 800 000 basic lemmas, there were still about 70 000 new lemmas encountered in the encyclopedia. About 50 000 of them were geographical or human names, but more than 20 000 were general lemmas (albeit of ”encyclopedic” nature) that were missing in the dictionary. Also the bilingual Czech/Slovak dictionary has been enlarged, so that currently the Czech-to-Slovak translation module is ready for practical exploitation.

The second direction, more interesting from the research point of view, is the direction of making the system truly multilingual by testing other language pairs. It is clear that a word-for-word approach to MT as it was described in previous sections is applicable only to languages with high degree of similarity. An open question is where is the real limit of applicability of the method, which pairs of languages are close enough for the method to provide reasonable quality of translation and which are not. It was therefore quite natural to try to extend the system to other Slavic languages.

4.1 Czech-to-Polish module

Due to the fact that, as far as we know, no other Slavic language has so many resources for stochastic natural language processing, it is quite natural that we kept Czech as a pivot language (source language of the machine translation module). The best candidate for a new target language was Polish. It is close enough to Czech but it contains several phenomena that are different and provide thus the natural ”next step”.

The Polish morphological data was kindly provided to us by Morphologic, Inc. (Budapest, Hungary). We converted the data for use with our morphological generator. In general, according to our expectations, with the decreasing similarity level also the quality of results has decreased. The main translation problems we have encountered:

- Word-order problems

While Slovak has almost identical word order as Czech, Polish contains several phenomena causing the necessity of word-order adjustments during the translation. The most obvious difference is the change of the word order in some types of nominal groups. Concerning congruent attributes, Czech prefers in most cases the order <Adj N> (i.e.adjective first, then noun), while Polish typically uses the order <N Adj> for adjectives defining a "species" of the nominal head, while the order <Adj N> is reserved for adjectives defining a "feature" of the noun.

- Problems of agreement

All kinds of differences in gender or case are another source of relatively frequent errors. Both Czech and Polish are languages with strong requirements of gender, number and case agreement not only between subject and verb (gender and number agreement), but also in several other types of constructions.

- Differences in cases

The first problem is the difference of valency frames (together with the associated subcategorization information) between source and target words. Unlike Slovak, Polish contains several words that have different valency frame than their Czech counterparts. This of course results in a translation error, because the main bilingual dictionary does not contain any valency (and subcategorization) information.

The second problem is the difference in prepositional constructions. For example, the Czech preposition *pro* [for] requires the use of the accusative case, while the corresponding Polish preposition *dła* requires the genitive case. Similarly (or even worse), some Czech cases are expressed by Polish prepositions.

- Lexical problems

The problem of polysemy or even plain homonymy is also quite serious. Often more Polish lexical units correspond to a single Czech one. A typical example is the Czech copula *nebo* [or], which may be translated either as *lub*, *bądź* (in more complex coordinations) or *czy* (yes-no questions only).

- Addressing the reader

One very interesting problem is the use of the gender-based *Pan/Pani* ([Mr./Mrs.] in the Polish 3rd Pers. Sg.) rather than genderless Czech polite form *vy* [You] (2nd Pers Pl. (auxiliary verb) / Sg.(predicate)).

- In Polish, the copula *być* [to be] usually cannot be omitted as it is in Czech, therefore, it must be inserted at the appropriate places.
- Polish 1st and 2nd person clitics

Czech forms *jsem* [I am], *jsi, jste* [you are], *jsme* [we are] are clitized to Polish floating suffixes -*(e)m*, -*(e)ś*, -*(e)śmy*, -*(e)ście*. These suffixes can attach to almost any word before the main verbal form but usually they go after the verbal form expressed by past participle, *powinien* and *jest* (present tense of *być* is reduplicated).

- For expressing that "something is something" Polish grammar admits only:
NP(Nom.)+ *być* (finite form)+NP(Instr.)
NP(Nom.)+ *być* (finite form)+Adj(Nom.)
Inf.+ *jest*(finite form)+adverb.

NP(Nom.)+ *to* (finite form)+NP(Nom.) (here *to* is a kind of predicative verb).

4.2 Czech-to-Lithuanian module

The tests of the Czech-to-Polish module confirmed our assumption that with decreasing similarity of both languages the quality of results will also decrease. It was also confirmed by an analysis of the planned Czech-to-Russian module described in (Homola, 2002). The paper suggested that one possible way of improving the quality of the translation would be an exploitation of a partial transfer.

The interesting question was whether it is possible to cross a borderline between different language groups. Due to the fact that Slavic and Baltic languages are relatively typologically similar (rich morphology, relatively free word order), it was decided to test the limits of the method by developing a Czech-to-Lithuanian module.

The initial comparative study showed that for Czech-to-Lithuanian translation it is necessary to enrich the scheme of the system by creating a shallow parser working with the results of the tagger and preceding the dictionary lookup phase.

Although we do have a full Czech statistical parser for Czech (Collins et al., 1999), its current accuracy (about 82-84% correct dependencies) was deemed not being sufficient for our task, while we even did not need a full parse. Therefore, the module of a shallow syntactic analysis of Czech is based on the LFG formalism, even though it does not use the complete LFG framework, as described in (Bresnan, 2001). We leave out e.g. the completeness and coherence conditions and anaphoric binding. The main goal of the module is to analyze only the simpler parts (constituents) of the sentence, such as nominal and prepositional phrases. The result of this module is an underspecified dependency tree.

The grammar consists of a set of phrase structure rules. Constraints (equations) are assigned to every element of the right-hand side of the rules. The application of the phrase structure rules gives the c-structures, whereas the constraints define the associated f-structures.

The module encountered the following translation problems:

- Aspect of verbs

  All perfective verbs in Czech express in their (grammatical) present tense a future action. As the property of aspect (perfectiveness) is inherent, it can be stored in the lexicon for every verb. This information can invocate switching the tense tag to ”future” in Lithuanian equivalents.
- Past tense

  In Czech, past tense is created by the auxiliary verb *být* [to be] and a past participle of the main verb (the auxiliary is omitted in the 3rd person of both numbers). In Lithuanian, the past tense is created by inflection.
- Reflexive verbs

  Reflexive particles of verbs are not translated at all, the non-reflexive variant must be used instead. The reason is that Czech reflexive verbs use the auxiliary pronoun *se* or *si*, whereas in Lithuanian, a suffix or infix is used.
- Neuter of adjectives

  Unlike Czech, Lithuanian does not have the neuter gender. This poses no problem for nouns, since it is handled by the bilingual dictionary, but it is a serious problem for adjectives and adjectival pronouns that syntactically depend on a noun in neuter gender in Czech. Because the text is translated word for word, no dependencies between words are created. If a neuter substantive with depending adjectives occurs in the source sentence, the morphological tag specifying gender is changed only for the noun. All adjectives keep their morphological tags unchanged and thus they are a source of errors. Most occurrences of this problem are solved by the shallow syntactical analysis of noun phrases we employ.

The shallow syntactic parser solved some of the translation problems satisfactorily and allowed the overall quality of the translation to achieve almost the same level as the quality of the Czech-to-Polish module.

5 Evaluation of results

The evaluation of MT systems can be, in general, performed from two different viewpoints. The first one is that of a developer of such a system, who needs to get a reliable feedback in the process of development and debugging of the system. The primary interest of such a person is the grammar or dictionary coverage and system performance. Such an evaluation method should also be cheap, fast and simple in order to allow frequent routine tests indicating the improvements of the system during the development of the system.

The second viewpoint is that of a user, who is primarily concerned with the capability of the system to provide fast and reliable translation requiring as few post-editing efforts as possible. The simplicity, speed and low costs are not of such importance here. If the evaluation is performed only once, in the moment when the system is considered to be ready, the evaluation method may even be relatively complicated, expensive and slow.

We have developed a very simple and straightforward evaluation method for our system. It was created more or less from the developer's viewpoint, but it also reflects the viewpoint of the potential user. It exploits the matching ability of TRADOS Translator's Workbench for expressing the degree of similarity of a text produced by our MT system and the text postedited by a human translator. The human translator receives the translation memory created by our system and translates the text using this memory. The translator is obviously free to make any changes to the text proposed by the translation memory. The target text created by the human translator is then compared with the text created by the mechanical application of translation memory to the source text. TRADOS then evaluates the percentage of match in the same manner as it normally evaluates the percentage of match of source text with sentences in translation memory.

There are several advantages of this evaluation method: It is simple and it is fast enough to provide a valuable feedback in the process of development of each bilingual translation module. Moreover, it in fact shows how much work the user needs to do in order to edit the results of our module and thus it is a good estimate of postediting costs: As far as we know, the TRADOS match computation is often used in practice for adjustments of per-word costs between a customer and a translation service bureau.

5.1 Translation results for all three bilingual modules

The testing of Czech-to-Slovak module was performed on relatively large texts (tens of thousands of words). The translation achieved on average a **90%** match (as defined by the TRADOS match module) with the human translation.

The testing of the remaining two modules was performed on much smaller sample of texts. This was mainly due to the experimental nature of both modules.

The weighted (length-adjusted) average match throughout the testing sample reached **71.4%**.

By further investigating the results, we found that

- 25,6% of sentences from the test sample did not require any postediting
- 33,3% of sentences achieved a match between 75% and 99%
- 24,4% of translated sentences had a match between 50% and 75%
- 16,7% of sentences were marked with less than 50% match against the correct, post-edited sentences

A match lower than 50% does not mean that the sentences are not usable for postediting. For example, one of the sentences with very low match was the following one:

Czech original:
Požadavky starší třiceti dnů se mažou.
[The requests older than 30 days are deleted.]
The result of MT:
Żądania starszy trzydziestu dzieni się smarują.
Post-edited Polish sentence:
Żądania starsze niż trzydzieści dni są wymazywane.

The match between the result of MT and the correct Polish sentence was 32% (according to TRADOS Translators Workbench standard computation), even though we need only 21 elementary operations to get the correct sentence (50 characters long) from the automatically translated one.

The weighted average match for the Czech-to-Lithuanian module (with the shallow analysis of Czech included) was almost as good as for the Polish, it scored **69%**.

In order to put all these numbers into a right perspective, we have performed one more test. We have taken 256 English sentences from the Penn Treebank and made them translated into Czech by human translators. The results of the translation were then translated back into English using one of the commercial Czech-English machine translation systems available on the market - PC Translator

2003. It has scored the weighted average match of **30%**.

6 Conclusion

We are of course far from drawing far reaching conclusions out of the above mentioned numbers, but according to our opinion they justify the hypothesis that word-for-word (lemma-based) translation might be a solution for MT of certain well chosen pairs of languages even across the borders of a language group. The results of Czech-to-Lithuanian module show that syntactic similarity is a more important factor than a simple typological close relatedness of languages.

7 Acknowledgements

The work described in this paper was supported by the grant No. 405/03/0913 and partially supported by the grant No. 405/03/0914 of the Grant Agency of the Czech Republic

8 References

Joan Bresnan. 2001. *Lexical-functional syntax.* Blackwell Publishers, Oxford.

Jan Hajič. 1998. *Building and using a syntactically annotated corpus: The Prague Dependency Treebank.* In: Festschrift for Jarmila Panevová, Karolinum Press, Charles University, Prague, pp. 106–132.

Jan Hajič and Barbora Hladká. 1998. *Tagging Inflective Languages. Prediction of Morphological Categories for a Rich, Structured Tagset.* ACL-Coling'98, Montreal, Canada, pp. 483-490.

Jan Hajič and Vladislav Kuboň. 2000. *Machine Translation of Very Close Languages.* In: Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, Washington, USA, pp. 7-12

Jan Hajič. 2001. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Karolinum, Charles University Press, Prague, Czech Republic.

Pavel Cikhart and Jan Hajič. 1999. *Word Sense Disambiguation for Czech Texts.* In: Proceedings of Text, Speech, Dialogue. Brno, Czech Republic, pp. 109-114

Petr Homola. 2002: *Machine translation among Slavic languages*. In: Proceedings of the WDS, Charles University, Prague, Czech Republic.

Karel Oliva. 1989. *A parser for Czech implemented in Systems Q*. Explizite Beschreibung der Sprache und automatische Textbearbeitung XVI, MFF UK, Prague, Czech Republic.

PDT. 2001. *The Prague Dependency Treebank,*. Available from LDC, www.ldc.upenn.edu, Catalog #LDC2001T10.

Michael Collins et al. 1999. *A statistical parser of Czech*. In Proceedings of the 37th ACL'99, University of Maryland, College Park, MD, USA. 505-512.