Statistical Methods in Czech-English Machine Translation

Jan Cuřín

Doctoral Thesis

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Prague 2006

Supervisor Doc. RNDr. Jan Hajič, Dr.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1

Opponents Ing. Alexandr Rosen, Ph.D
Institute of Theoretical and Computational Linguistics
Faculty of Arts, Charles University in Prague
Celetná 13
110 00 Prague 1

Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1

I certify that this doctoral thesis is all my own work, and that I used only the cited literature. The thesis is freely available for all who can use it.

Prague, April 19, 2006

Acknowledgments

I would like to thank my supervisor Jan Hajič for guiding the direction of my work and for giving me opportunities to meet prominent people from the natural language processing research community.

Very special thanks go to Jarmila Panevová who inspired me to the computational linguistics through her outstanding seminars when I was looking for the future direction and who also encouraged me during the whole study. Without her aid, valuable comments, and moral support I would not be able to finish this work.

I cannot omit to thank my friend and closest colleague for the last decade Martin Čmejrek and the actual member of our "machine translation team" Jiří Havelka. This work could not have been even started without the solid ground and support provided by all colleagues from the IFAL.

I am also grateful to my parents, brother, sister, and to my designate wife Lenka for their endless belief that I would write this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Think, by analogy, of individuals living in a series of tall closed towers, all erected over a common foundation. When they try to communicate with one another, they shout back and forth, each from his own closed tower. It is difficult to make the sound penetrate even the nearest towers, and communication proceeds very poorly indeed. But, when an individual goes down his tower, he finds himself in a great open basement, common to all the towers. Here he establishes easy and useful communication with the persons who have also descended from their towers.
>
> **Warren Weaver**

Automatic translation from one human language to another using computers, better known as machine translation (MT), is a longstanding task for computational linguists. This may be closely related to the fact that (machine) translation is perceived by the public as a more practical and useful product than some other tasks pursued by theoretical and computational linguists.

To be able to perform such a task, the computer has to “know”, in addition to the proper translation dictionary, the two languages’ morphological rules, syntactic and semantic features, and at least a basic world knowledge. One way to incorporate such knowledge into a computer is to hire bilingual experts and linguists and let them hand-craft the necessary information to the computer system. Another way is to let the computer learn some of these things automatically by examining large amounts of parallel texts, i.e. documents which are manual translations of each other. Let us call the former *linguistic* (or “rule-based”) approach and the latter *statistical* approach to machine translation.

In rule-based MT, the sentence from the input language is first analyzed by morphological, syntactic and semantic rules, then translated into the output language by so-called transfer rules, and finally other rules are used for the generation of the output language surface form. Such rules are to be manually crafted and usually cost many person-years of hard work. The advantage of such rule-based MT systems is a relatively high quality or readability of its output, whereas disadvantage is the large amount of effort to cover all phenomena specific to a given pair of languages. Moreover such a system might not be flexible enough to follow the natural evolution of the language, including the addition of new words or terms and changes in the usage of certain grammatical constructions.

In the last decade, the focus of machine translation has shifted from the rule-based to the statistical-based approach, which has its roots in the work done by a group at IBM published in the late 80's [Brown et al., 1988]. They adapted a statistical model called *noisy-channel model*, used in signal processing, to recover the original signal from a signal with noise. The idea of casting the task of translation as corrupted code recovery was already suggested by Warren Weaver in late the 40's [Weaver, 1955].

Machine translation from/to Czech has a long tradition at the Charles University in Prague. From 1977 to 1986, an English-to-Czech translation system called APAČ was designed and implemented by the group lead by Zdeněk Kirchner [Kirschner, 1987, Kirschner and Rosen, 1989].

In the late 80's, a team comprising Jan Hajič, Karel Oliva, Hana Skoumalová, Alexandr Rosen, and Alla Bémová built a Czech-Russian translation system: RUSLAN [Hajič, 1987]. The work on the Czech/English language pair continued in the translation system called MATRACE (MAchine TRAnslation between Czech and English) [Hajič et al., 1992]. Nowadays, beside the statistical approach to MT, a shallow machine translation based on the morphological analysis is applied for the translation between Czech and a set of close languages, such as Slovak, Polish or Lithuanian [Hajič et al., 2000b]. The most mature system among then – Česílko [Hajič et al., 2000a] – translates from Czech to Slovak.

This thesis focuses on the task of automatic machine translation from Czech to English. We propose two different strategies of using linguistically enriched parallel corpora to improve machine translation. The first strategy is to incorporate linguistic knowledge into the statistical approach to machine translation and the second strategy is to introduce automatically trained parameters of statistical models into the linguistic approach to machine translation. We also suggest additional methods for exploiting parallel corpora, such as the automatic extraction of a dictionary or the weighting of manual dictionary sources with respect to a specific domain.

An integral part of this work is the collection and further processing of various data sources. One of the main contributions is the building of a rich, syntactically annotated parallel corpus of Czech and English, which benefits from the well elaborated theory represented by the Prague linguistic school and the Functional Generative Description [Sgall et al., 1986].

An introduction into Statistical Machine Translation theory is in the Chapter 2. We start by the analogy with the task of automatic speech recognition and continue with a description of translation models and several decoding approaches.

Chapter 3 lists available parallel corpora, briefly presents methods of automatic sentence alignment, and describes the process of building the already mentioned syntactically annotated parallel corpus – the Prague Czech-English Dependency Treebank.

Chapters 4 and 5 represent the author's main contribution.

Chapter 4 presents methods for creating, filtering, and weighting of translation dictionaries. Next Chapter 5 introduces the two strategies how to use linguistically enriched parallel corpora to improve machine translation. In Section 5.1, we propose a method based on a preprocessing of syntactically annotated input (the analytical dependency structure) to enhance the performance of the statistical machine translation system. Section 5.2 describes the dependency-based machine translation approach and discusses the use of an automatically built and weighted dictionary in this system.

Discussion of the results and description of the evaluation metric can be found in Chapter 6. Final remarks are made in Chapter 7.

Chapter 2

Statistical Machine Translation

> A translation is like health: if it's good, you don't notice it. But if it's not, it becomes very important and can ruin a whole book, just as even a minor health problem can ruin your whole life.
>
> **Elborg Forster**

2.1 Introduction

The statistical modeling approach to machine translation (Statistical Machine Translation – **SMT**) was first introduced in the system called Candide built by a group of researchers from IBM in the late 80's [Brown et al., 1988, Brown et al., 1990]. They adapted a statistical model called *noisy-channel model*, used in signal processing, to recover the original signal from a signal with noise. The idea of casting the task of translation as corrupted code recovery had already been suggested in [Weaver, 1955].

Let us first explain the noisy-channel model on the task of automatic speech recognition (**ASR**). Figure 2.1 shows the schema of the noisy channel, introduced by Jelinek in the eighties [Jelinek, 1985]. The story is that the speaker, having in mind the sentence he wants to say (*word source* – $\mathbf{W}$), encodes it into an acoustic signal, and sends it into the noisy channel by pronouncing it (*speech production*). On the other side of the noisy channel, the acoustic sensor (or *acoustic processor*) 'hears' a noise (*acoustic signal* – $\mathbf{A}$) that might be garbled on the way by many different means.

For illustration, we can imagine a chat over a drone phone line with a mumbling, non-native speaker, having bad pronunciation of the consonants 'r' and 's'. How it is possible that you can understand at least part of what the speaker said from such noise? The answer is: because of the context and the knowledge of the spoken language. In ASR, this knowledge is embodied in the so-called *Language Model*, which is a statistical model trained on a large amount of text from the target language. Finding the best interpretation in words ($\widehat{\mathbf{W}}$) of the received acoustic signal is then the task for a *linguistic decoder*.

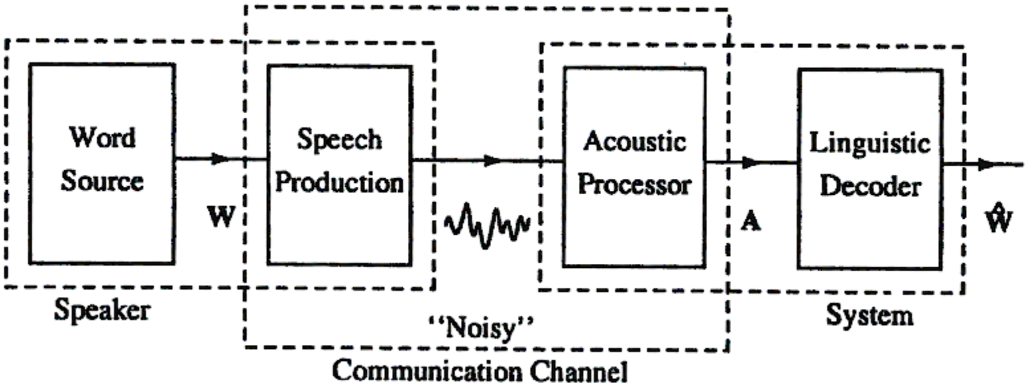

Figure 2.1: Noisy-Channel Model for ASR (from [Jelinek, 1985])

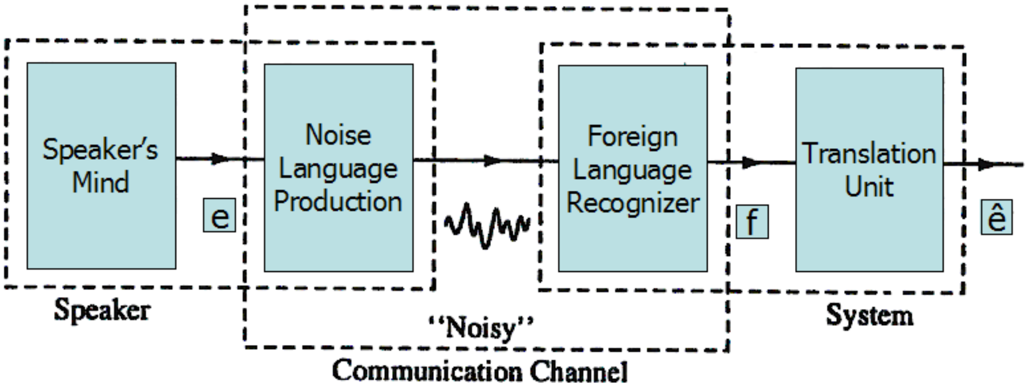

Figure 2.2: Noisy-Channel adopted for Machine Translation

Formally, the ASR is finding the sequence of words W maximizing the probability given the acoustic signal A. Using the Bayes theorem, the formula can be rewritten as a product of the *acoustic model* $\mathbf{P}(A|W)$ and the *language model* $\mathbf{P}(W)$. The probability of the acoustic signal itself $\mathbf{P}(A)$ can be ignored, as it is constant wrt. W:

$$\arg\max_W \mathbf{P}(W|A) = \arg\max_W \frac{\mathbf{P}(A|W)\ \mathbf{P}(W)}{\mathbf{P}(A)} = \arg\max_W \mathbf{P}(A|W)\ \mathbf{P}(W)$$

The advantage of this decoupling is the use of an independent language model which can be trained on potentially unlimited amounts of monolingual data. For comparison, sources of speech signals and their transcriptions needed for the training of the acoustic model are much rarer.

For the machine translation task we replace the acoustic model with a *translation model.* The interpretation then becomes as follows (see Figure 2.2):

The speaker (or writer in text-to-text translation) mumbles the sentence in such a way that he says/writes it by mistake in French instead of in English. The translation unit then tries to recover the English text from the noisy French. We use French-English as an example as this language pair was used in the first SMT articles. f, which originally stood for French, was later considered as '*foreign*'.

Assuming a translation from French to English, we use the English language model $\mathbf{P}(e)$ and a translation model estimating the probability of the French (or *foreign*) sentence given the English sentence $\mathbf{P}(f|e)$. The formula is then:

$$\arg\max_e \mathbf{P}(e|f) = \arg\max_e \frac{\mathbf{P}(f|e)\ \mathbf{P}(e)}{\mathbf{P}(f)} = \arg\max_e \mathbf{P}(f|e)\ \mathbf{P}(e)$$

Again, this decoupling allows us to train the English language model on a big amount of monolingual data, whereas for training of the translation model we need a scarce parallel corpus.

Fortunately, the task of creating a reasonable language model is shared among both speech recognition and machine translation. The basic idea of the language model is a so-called n-gram model which estimates the probability of a word in the context consisting of the $n-1$ preceding words in the sentence. A simple 3-gram model can be counted from the number of occurrences c of appropriate triples and doubles of words in the text corpus:

$$\mathbf{P}(w_n|w_{n-1}, w_{n-2}) \approx \frac{c(w_{n-2}\ w_{n-1}\ w_n)}{c(w_{n-2}\ w_{n-1})}$$

Then, the probability of a sentence W is the product of the probabilities of the individual words w_i of the sentence:

$$\mathbf{P}(W) = \prod_i w_i$$

State-of-the-art language modeling tools of course use more advanced techniques such as smoothing, pruning, and incorporation of linguistic knowledge. Details and references about this research area can be found, for example, in [Stolcke, 2002,Krbec, 2005]. For details about the language models used in our experiments, refer to Sections 5.1.4 and 5.2.2.

2.2 Translation Model

The purpose of the translation model is to assign a probability $\mathbf{P}(f|e)$ to a given pair of sentences.

Even though the first statistical translation model was proposed by IBM in 1988 [Brown et al., 1988], the boom of research in SMT started after the Natural Language Processing (NLP) Workshop held at Johns Hopkins University in 1999, where the publicly available tool for building translation models called GIZA was implemented. GIZA, which was part of the Egypt system [Al-Onaizan et al., 1999], supported *IBM models 1, 2, and 3*, as proposed in [Brown et al., 1993]. Implementation of *IBM models 4 and 5* is available in GIZA++ [Och and Ney, 2000]. Details about IBM models follow.

2.2.1 IBM Models 1, 2, and 3

All IBM models work with word-to-word translations. Punctuation is considered as a separate word.

Model 1 is the simplest model. Technically speaking, it is a *table of translation probabilities* $[t(f|e)]$ for each possible word-translation pair, with the addition of a *NULL word*, where binding the English word to *NULL* means word-deletion, and binding the foreign[1] word to *NULL* indicates word-insertion.

Model 2 extends Model 1 by adding an *alignment table* $[a(i|j,l,m)]$ which says what the probability is that the English word appears in a particular position in the sentence (i), given the position of the aligned (or paired) foreign word (j) and the lengths of both English (l) and foreign (m) sentences.

Model 3 introduces the *fertility* (ϕ) of a word in the source language. *Word fertility table* $[n(\phi|e)]$ indicates the probability of the number of foreign words induced from a given English word. For example in Figure 2.3 the English word '*slap*' induces three Spanish words '*daba una bofetada*'. The alignment table from Model 2 is transformed to a *distortion table* $[d(j|i,l,m)]$ in Model 3 .

Model 3 represents a particular naive theory of how, given an English sentence, one might stochastically produce foreign-language equivalents. The theory shown in the example of Figure 2.3 goes like this:

1. For each English word e in the sentence, we get the fertility ϕ with probability $n(\phi|e)$. In our example the word '*slap*' has a fertility of 3, the word '*did*' 0, and other words 1,

[1] In this work "*foreign*" might be French, Spanish, or Czech depending on the context and examples.

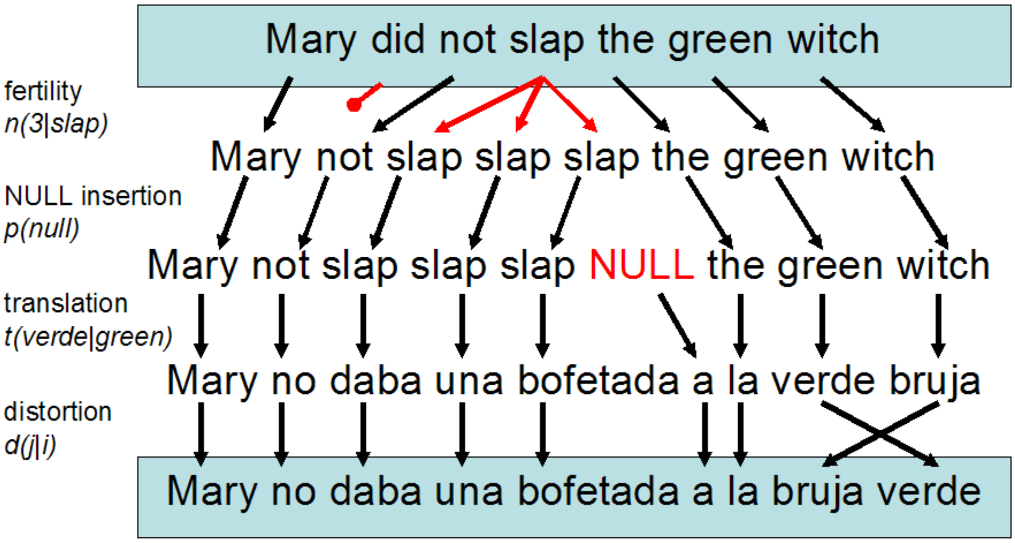
Mary did not slap the green witch
fertility
n(3|slap)
Mary not slap slap slap the green witch
NULL insertion
p(null)
Mary not slap slap slap NULL the green witch
translation
t(verde|green)
Mary no daba una bofetada a la verde bruja
distortion
d(j|i)
Mary no daba una bofetada a la bruja verde

Figure 2.3: IBM Model 3 from [Knight and Al-Onaizan, 1998]

2. Call the sum of all those fertilities m', $m' = 8$ in this example,

3. Make a new English string by deleting words with fertility zero, copying words with fertility one, duplicating words with fertility two, etc. (second line),

4. After each of these m' words, make a decision to insert a NULL (with probability p_1) or not (with probability p_0). One word later translated as '*a*' was added (third line),

5. Let ϕ_0 be the number of NULL words inserted, $\phi_0 = 1$ here,

6. Let m be the total number of words in front of us now, $m' + \phi_0$ ($m = 9$),

7. Replace each word e with a foreign-language translation f, according to the probability table $t(f|e)$ (line four),

8. Assign target-language positions to foreign language words not generated by NULL, according to the probability table $d(j|i,l,m)$. Here, j is the foreign-language position, i is the position in the English string of the words that generated the foreign word now being placed, l is the number of words in the English string, and m is the number of words in the foreign string. Swap the words '*bruja*' and '*verde*',

9. If any target-language position is oversubscribed (contains more than one word), then return failure (fortunately, this is not the case in our example)

10. Assign target-language positions to the NULL-generated words. These should fall into empty positions. Any assignment is deemed equally likely as any other, so any assignment can be carried out with probability $1/\phi_0$. Word '*a*' in the example.

11. Finally, read off the foreign (Spanish in the example) string.

Training is a matter of inducing these probability tables from a bilingual corpus. The basic idea is to bootstrap, i.e. for a given English word e, we initially pretend that all foreign words are equally likely translations. For a given sentence pair, all alignments will therefore look equally likely as well. Then we count word pairs co-occurring in an alignment. After the traversal of the entire corpus, we normalize these counts to create a new word-translation table. The same goes for the fertility and distortion tables. In these new tables some alignments will now be more probable than others. In the next iteration we collect counts again, but now weigh co-occurrences by the probability of the alignments that they occur in. More details on the implementation of this training can be found in Section A.1.

2.2.2 IBM Model 4

Model 4 comes with more intuitive handling of distortion than the preceding models where word reordering depended only on the length of the sentences, completely ignoring the words in both languages. Model 4 deals with *word classes* and relative positioning. Word classes $[C(e), C(f)]$ are automatically derived for both languages independently using a clustering algorithm [Brown et al., 1992]. The distortion table d known from Model 3 is divided into two tables: d_1 and $d_{>1}$. Table d_1 is used for words with fertility 1 or for the 'head' (= first word) of the foreign-language string resulting from higher fertility ($\phi > 1$). For example, $d_1(+2|C(e_{-1}), C(f_0))$ means that if the foreign word f_0 belongs to class $C(f_0)$ and the previous English word e_{-1} belongs to class $C(e_{-1})$, then the position of f_0 is $+2$ relative to the previously computed position of f_{-1}. Likewise, $d_{>1}(+1|C(f_{+1}))$ determines the relative position to the preceding word in the phrase of foreign words generated from more fertile English words.

Model 4 also introduces a mechanism for dealing with the so-called *deficiency* of Model 3. The model is called *deficient* if it assigns positive probabilities to output which should never happen. This is because the distortion probability table allows moving more than one English word to the same position in the foreign-language output sentence, since the word-move operations are independent.

table name *formula* *used in*	*description*
translation table $t(f\|e)$ all models	This two dimensional table (number of English words in vocabulary × number of foreign words in vocabulary) contains the translation probability for each word-translation pair. Translation probabilities are refined during the EM-training of every translation model.
alignment table $a(i\|j,l,m)$ model 2	Contains the probability of changing the position of the translation, i.e. what is the probability that word e will be on the position i when its translation f was on the position j, given the length of both sentences, l for original, m for translated.
distortion table $d(j\|i,l,m)$ model 3	Similar to *alignment table* in Model 2, where conditioning is swapped for English and foreign string positions. It makes the Model 3 deficient.
fertility table $n(\phi\|e)$ models 3 and 4	Contains the probability that a given English word is to be replaced by ϕ foreign words. $\phi = 0$ means word-deletion, $\phi > 1$ implies word-insertion.
word classes $C(e), C(f)$ model 4	Each word is assigned to a class. Word-classes are trained automatically for both languages. Example of such classes might be days of the week, names of countries, or different sorts of vegetable, i.e. words which appear in the same context.
distortion table* $d_1(\pm x\|C(f), C(e_{-1}))$ model 4	This table assigns the probability that a foreign word f of a given word-class $C(f)$ will be placed on the position relative to the position of the word-class of the previous English word. If the current English word has fertility > 1, it determines the position of the first foreign word in the resulting string.
distortion table$^+$ $d_{>1}(\pm x\|C(f))$ model 4	This table assigns the relative position of a word in the foreign string resulting from a more fertile English word given the word-class of the foreign word.

Table 2.1: Overview of probability tables used in IBM's translation models

In the original IBM article, and in some MT literature, the extension of Model 4 which removes this deficiency is called Model 5.

Let us summarize the parameters of IBM's translation models in Table 2.1. For the experiments described in this work we employ the non-deficient IBM Model 4 using the `GIZA++` tool.

In Section 5.1, we propose and evaluate a Czech-to-English translation system based on statistical methods further extended by incorporation of linguistic preprocessing specific to the Czech/English language pair.

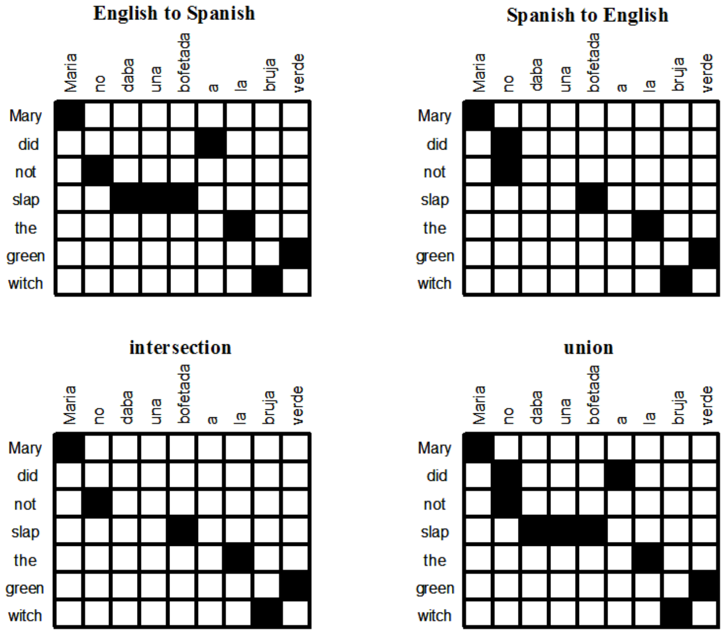

Figure 2.4: Intersection and union of two alignments produced by GIZA++

2.2.3 Alignment Templates

The *Alignment Templates model* (in literature also called *Phrase-based model* or *Statistical Phrase-based Machine Translation*[2]) was introduced by [Och, 2002]. Instead of finding word-to-word correspondences in the parallel corpus it aligns subsequences of words (phrases) in the two parallel sentences. IBM's word-to-word translation model has a serious drawback, as it allows only *1-to-many* mapping for English words, i.e. at most one English word can be aligned with each foreign word.

The alignment template model is computed as follows: first the parallel corpus is aligned bidirectionally using IBM's word-to-word model. Then, for each sentence pair and both translation directions, a Viterbi alignment is created, as shown for our sample sentence pair in Figure 2.4. The optimal alignment is then computed from these two.

Various techniques for finding the optimal alignment between the intersection and the union of the two alignments have been investigated. In Figure 2.5, we show two sets of possibly identified phrase mappings in the

[2] The label *Phrase-based MT* was introduced in [Koehn et al., 2003], but the use of the term *phrase* may be misleading. Unlike its usual linguistic meaning from syntax (noun phrase, prepositional phrase), it is rather an n-gram, i.e. a contiguous subsequence of words from the sentence.

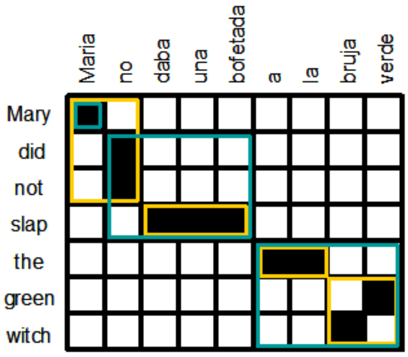

Figure 2.5: Example of phrase mapping in Alignment Templates model

optimal alignment, represented by colored rectangles over the grid.
The first alignment (in blue) represents the following phrase pairs:

```
(Maria, Mary), (no daba una bofetada, did not slap),
(a la bruja verde, the green witch)
```

The second alignment (in yellow) pairs are:

```
(Maria no, Mary did not), (daba una bofetada, slap), (a la, the),
(bruja verde, green witch)
```

The phrase translation probability table is then estimated by the relative frequency of all collected phrase pairs in the corpus ($\overline{f}$ stands for the phrase in the foreign language, and $\overline{e}$ for the English phrase):

$$\mathbf{P}(\overline{f}|\overline{e}) \approx \frac{c(\overline{f}, \overline{e})}{\sum_{\overline{f}} c(\overline{f}, \overline{e})}$$

For details see [Och, 2002]. A discussion about various statistical translation models can be found in [Och and Ney, 2003]. For the SMT experiments in this work, we use the GIZA++ toolkit for word-to-word translation models training and the PHARAOH system [Koehn et al., 2003] for building alignment template model and decoding.

2.3 Decoder

> Codebreakers are linguistic alchemists, a mystical tribe attempting to conjure sensible words out of meaningless symbols.
>
> **Simon Singh**

As outlined in Section 2.1 of this chapter, the primary goal of the translation model, combined with the language model, is to create a translation system. The task of finding the most probable sentence with respect to both translation and language models is called *decoding.* It was proved in [Knight, 1999] that, for an arbitrary word-ordering, the decoding in machine translation (using IBM Model 4) is an NP-complete problem.

This section briefly introduces decoding methods suitable for statistical machine translation.

2.3.1 A* Decoding

The A* decoding algorithm (also called *stack decoding*) is a best-first search method. Again, it was first introduced in the domain of speech recognition [Jelinek, 1969]. The decoder searches the solution space incrementally, storing particular hypotheses in a *stack.* Theoretically, using unlimited stack size and exhaustive search time, the stack decoder finds an optimal solution.

A generic A* decoding algorithm works as follows:

1. Initialize the stack with an empty hypothesis
2. Pop the best hypothesis h from the stack
3. If h is a complete sentence, output h and terminate
4. For each possible next word w, extend the current hypothesis h by adding w and push it into the stack
5. Repeat from step 2. (pop)

Now, the crucial question is how to find that the hypothesis is a complete sentence.

The main difference between the decoding process in speech recognition and machine translation is that speech is always produced in the same order as its transcription. Decoding in speech recognition follows the left-to-right correspondence between input and output sequences, but we cannot use the left-to-right relation in the machine translation task. As we do not know the order of the input, we have to consider all $n!$ permutations of the n-words long input sentence. Obviously, it is also more difficult to find good heuristics for the estimation of hypothesis weight and completeness in the MT decoding. Without any heuristic, the decoder would prefer shorter hypotheses. This effect can be eliminated by using multiple stacks, one stack for each subset of the input sequence. Multiple stack decoding has been patented by IBM [Brown et al., 1995] and used in the CANDIDE translation system [Berger et al., 1994]. We used an implementation of this decoder in the NLP workshop in 1999 [Al-Onaizan et al., 1999] for experiments with Czech-English translation, but we had difficulties (in computation time) to decode sentences longer than 25 words. Therefore we used a non-optimal but much faster *greedy decoder* in later experiments.

2.3.2 Greedy Decoding

Greedy decoding is based on small incremental improvements of an approximate starting solution, rather than on a search in global space as the stack decoder does. It is called greedy because it headlong chooses the most promising probable direction independently in each step. The greedy approach has been found useful for many NP-complete tasks in the last decades and it behaves surprisingly well even on the machine translation decoding task, as shown in a comparison of greedy, stack, and integer programming decoding for MT in [Germann et al., 2001].

The greedy decoder starts the translation process by replacing each word in the input (Czech) sentence by its most probable translation in the target language (English). We call this most probable translation the *gloss*.

Once the initial "glossy" translation is created, the greedy decoder tries to find in a neighborhood a target sentence with higher probability by applying one of the following operations:

- Translate one or two words in one step, including the removal of a target word if a selected translation is the NULL word.
- Translate one word and add one other translation (of a different word) if it increases the probability of the resulting sentence.
- Remove a translation for a word with zero fertility.
- Swap two non-overlapping segments.
- Eliminate one target translation and align any non-bound source word to another translation in the sentence.

The decoding stops if it can not find a more probable output sentence in the neighborhood, or if it takes too long (by limiting the decoding time per sentence or the maximum number of applied operations).

2.3.3 Beam Search Decoder

A beam search decoding algorithm generates the output (English) sentence from left to right. It uses multiple stacks, a hypothesis being put into the stack with an index corresponding to the number of foreign words covered by the hypothesis. The number of stacks is equal to the number of words in the input (foreign) sentence (f). Let $\theta(h)$ be a function representing the number of covered foreign words in the hypothesis h. The algorithm works as follows:

- Create $f+1$ empty stacks (**stack**[0] ... **stack**[f]).
- Add an empty hypothesis to **stack**[0].
- Iterate over stack index. Start with stack index $i = 0$.
- Try all possible extensions for each hypothesis h in **stack**[i]
    - Each new hypothesis n created by extending the hypothesis h by word or phrase (in phrase-based SMT) is added into **stack**[$\theta(n)$]
    - Prune **stack**[$\theta(n)$] if it become too large.
- Continue with next **stack**[$i+1$], i.e. next iteration.
- After all hypotheses in the **stack**[$f-1$] are processed, find the best hypothesis in **stack**[f]
- Output the path to the best hypothesis

The tricky part of the algorithm is the implementation of pruning "too large" stacks. Without pruning, the number of items in a stack will grow exponentially with the input sentence length. In [Koehn, 2004], the pruning is based on the actual probability of a partial hypothesis and on an estimation of the future cost. The future cost is tied to the foreign words that have not yet been translated, combining translation and language model probabilities. Future costs for all contiguous sequences of words are precomputed for each input sentence.

The beam search decoder is implemented in the PHARAOH system used for SMT experiments in this work.

2.4 Other Uses of The Translation Model

Another possible outcome of the translation model is the ability to automatically create various probability tables from the parallel text with no or limited need for human expertise. In Chapter 4, we show the application of these statistical methods to the task of building various types of translation dictionaries. The use of such dictionaries in a classical analysis→transfer→generation machine translation system is shown in Section 5.2.

Chapter 3

Parallel Corpora

There's no data like more data!
Speech researchers

No statistical method could be applied without data. This is even more true for such a complicated task as machine translation. At that time (1997) there was only one Czech-English parallel corpus available - the Czech-English part of the MULTEXT-East project containing a parallel corpus from the novel "1984" by George Orwell [Petkevič, 1999]. As the size of this corpus was not sufficient for SMT, we have decided to collect parallel texts and to create our own sentence-level aligned corpus.

In the experiments described in this work, the following three Czech-English parallel corpora are used:

- **Reader's Digest Corpus** – parallel text of articles from the Reader's Digest, 1993-1996. The Czech part is a translation of the English one. The Reader's Digest corpus consists of 60,000 sentence pairs from 450 articles.
- **Computer Oriented Corpus** – parallel text from localization of IBM operating systems and manuals. This technical text contains more than 1 million words in 120,000 sentences[1].
- **Prague Czech-English Dependency Treebank** – a syntactically annotated parallel corpus containing 21,000 parallel Czech-English sentences. More than half of the English Penn Treebank was translated into Czech and annotated for MT research purposes.

The first two data sources – Reader's Digest and Computer Oriented Corpus – were in plain-text or HTML form with neither sentence nor paragraph alignment. The process of retrieving a parallel corpus from plain text with

[1] Unfortunately the IBM corpus is not publicly available and can be used only for internal experiments at IFAL.

details about these two corpora is described in Section 3.1. For a detailed description, refer to [Čmejrek, 1998,Cuřín, 1998]. The Prague Czech-English Dependency Treebank is introduced in Section 3.2.

3.1 Building a Parallel Corpus from Parallel Texts

The alignment of texts in two languages is a long-standing task for NLP, the complex overview of this problem can be found in [Melamed, 2001]. A comparison of methods for the sentence alignment was published in [Rosen, 2005]. We have used the statistical model from [Gale and Church, 1993]:

3.1.1 Statistical Alignment of Paragraphs and Sentences in Czech-English Bilingual Corpora

The task of identifying matching paragraphs and sentences between two languages can be formalized as follows:

Let us have Czech and English texts (typically paragraphs) $T_\mathbf{c}$ and $T_\mathbf{e}$. The alignment is a set of pairs of parts of texts (typically $0, 1, 2 \ldots n$ sentences) $\{(L_{c,1} \rightleftharpoons L_{e,1}), \ldots, (L_{c,n} \rightleftharpoons L_{e,n})\}$ such that $L_{c,1}, \ldots, L_{c,n} = T_c$ and $L_{e,1}, \ldots, L_{e,n} = T_e$. We are looking for the best alignment $\mathcal{A}$ that maximizes the likelihood over all possible alignments on $T_\mathbf{c}$ and $T_\mathbf{e}$:

$$\arg\max_{\mathcal{A}} \mathbf{P}(\mathcal{A}|T_\mathbf{c}, T_\mathbf{e}). \tag{3.1}$$

Now, we have to make several approximations to obtain an effectively computable model: first, the set of possible types of matching parts of texts is limited to six categories: 1-1, 0-1, 1-0, 1-2, 2-1, 2-2 (in number of aligned paragraphs or sentences). We also assume that the probabilities of individual aligned parts of texts $\mathbf{P}(L_{c,i} \rightleftharpoons L_{e,i})$ are independent. And finally, we assume that the probabilities of individual alignments depend only on the function δ of the lengths of aligned parts $l_\mathbf{c}$ and $l_\mathbf{e}$ (the length might be considered in words or in characters, we use $l_\mathbf{c} = \mathit{length\_in\_chars}(L_\mathbf{c})$) and $l_\mathbf{e} = \mathit{length\_in\_chars}(L_\mathbf{e})$)). Therefore, the following approximation holds:

$$\arg\max_{\mathcal{A}} \mathbf{P}(\mathcal{A}|T_\mathbf{c}, T_\mathbf{e}) \approx \arg\max_{\mathcal{A}} \prod_{(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}) \in \mathcal{A}} \mathbf{P}(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}|\delta(l_\mathbf{c}, l_\mathbf{e})), \tag{3.2}$$

where function δ is defined as follows:

$$\delta(l_\mathbf{c}, l_\mathbf{e}) = \frac{l_\mathbf{e} - e l_\mathbf{c}}{\sqrt{l_\mathbf{c} \sigma^2}}, \tag{3.3}$$

where $e = E(r) = E(\frac{l_\mathbf{e}}{l_\mathbf{c}})$ is the mean number of English characters generated by each Czech character, and $\sigma = D(\frac{l_\mathbf{e}}{l_\mathbf{c}})$ is the variance. After applying Bayes' Rule and with the assumption that δ is normally distributed:

$$\arg\max_{\mathcal{A}} \prod_{(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}) \in \mathcal{A}} \mathbf{P}(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}|\delta(l_\mathbf{c}, l_\mathbf{e})) \approx$$

Type of	Computer	Oriented	Reader's	Digest	Canadian	Hansards
Alignment	# sent.	$\mathbf{P}(L_c \rightleftharpoons L_e)$	# sent.	$\mathbf{P}(L_c \rightleftharpoons L_e)$	# sent.	$\mathbf{P}(L_f \rightleftharpoons L_e)$
1–1	109	0.90	64	0.69	1167	0.89
1–0, 0–1	3	0.03	3	0.03	13	0.01
1–2, 2–1	7	0.06	24	0.26	117	0.09
2–2	1	0.01	2	0.02	15	0.01
total	120	1.00	93	1.00	1312	1.00

Table 3.1: Sentence alignment type distribution on a hand-annotated sample.

$$\approx \arg\min_{\mathcal{A}} \sum_{(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}) \in \mathcal{A}} -\log \frac{\mathbf{P}(\delta(l_\mathbf{c}, l_\mathbf{e}) | l_\mathbf{c} \rightleftharpoons l_\mathbf{e}) \mathbf{P}(L_\mathbf{c} \rightleftharpoons L_\mathbf{e})}{\mathbf{P}(\delta(l_\mathbf{c}, l_\mathbf{e}))} \approx$$

$$\approx \arg\min_{\mathcal{A}} \sum_{(L_\mathbf{c} \rightleftharpoons L_\mathbf{e}) \in \mathcal{A}} -\log \left(\frac{2(1 - \frac{1}{\sqrt{2\Pi}} \int_{-\infty}^{|\sigma|} e^{\frac{-z^2}{2}} dz) \mathbf{P}(L_\mathbf{c} \rightleftharpoons L_\mathbf{e})}{\mathbf{P}(\delta(l_\mathbf{c}, l_\mathbf{e}))} \right). \quad (3.4)$$

Parameters e, σ^2 and $\mathbf{P}(L_\mathbf{c} \rightleftharpoons L_\mathbf{e})$ are estimated from a sample of hand-aligned sentences. $\mathbf{P}(\delta(l_\mathbf{c}, l_\mathbf{e}))$ is constant and as such does not influence the result of minimization. Table 3.1 compares parameters of three different corpora. Although the *Canadian Hansards* and the *Czech-English Computer Oriented* corpora have very similar distribution of categories of alignment, the distribution differs substantially on the *Reader's Digest* corpus.

The best alignment is now easy to find using a dynamic programming procedure. In the first step, the algorithm aligns paragraphs in matching articles, and in the second step, it aligns sentences in matching paragraphs.

Table 3.2 summarizes results of the automatic paragraph and sentence alignment. The accuracy was 96 correctly aligned pairs on the *Computer Oriented* corpus and 85 pairs on the *Reader's Digest* corpus in a randomly selected sample of 100 pairs of sentences in each corpus. The accuracy was verified manually.

3.1.2 Reader's Digest Corpus

We had access to a Czech/English corpus which is a parallel text of articles from the Reader's Digest, 1993-1996. The Czech part is a translation of the English one. The Reader's Digest corpus consists of 53,000 sentence pairs from 450 articles. Sentence pairs were aligned automatically by the [Gale and Church, 1993] algorithm, but this alignment was not sufficiently good. The corpus was realigned using SIMR/GSA [Melamed, 1996]. With language-pair-specific parameter settings learned from a small amount of word-aligned

	Computer Oriented	Reader's Digest
# words (English)	1245780	959583
# words (Czech)	1089813	860757
# paragraphs (English)	88790	19567
# paragraphs (Czech)	88790	24874
# sentences (English)	120743	70872
# sentences (Czech)	121295	67856
# aligned sentences	119886	67436
types of alignment:		
1–1	117450 (98%)	37039 (57%)
0–1 (En/Cz)	73 (0%)	5311 (8%)
1–0 (En/Cz)	36 (0%)	4454 (7%)
1–2 (En/Cz)	1397 (1%)	9501 (15%)
2–1 (En/Cz)	882 (1%)	7342 (11%)
2–2	48 (0%)	1089 (2%)
accuracy:		
(on 100 samples)	**96%**	**85%**

Table 3.2: Results of automatic paragraph and sentence alignments.

Whenever young Les Polsfuss was sick, his mother put him on the parlor couch in their home in Waukesha, Wis. There Les could hear boxcars from the Chicago-St. Paul line rumbling up and down a nearby siding. Listening to the trains one morning at age five, he noticed that when the sound reached a certain pitch, it made the window vibrate. That's strange, he thought. Feeling the glass, he discovered he could dampen the vibration but couldn't make it stop. Only when the train's speed and pitch changed did the windowpane become silent.

Když Les Polsfuss jako malý kluk stonal, ustlala mu vždycky maminka na gauči v obývacím pokoji jejich domu ve wisconsinské Waukeshe. Tady naslouchal dunění nákladních vagonů jezdících po nedaleké železniční odbočce. Bylo mu asi pět, když jedno ráno opět zaslechl projíždět vlak a při tom si všiml, že pokaždé, když zvuk vagonů dosáhne určité výšky, okenní tabulky se rozdrnčí. To je zvláštní, pomyslel si. Sáhl na sklo a zjistil, že chvění může sice zmírnit, ale že ho nezastaví úplně. Drnčení přestalo, jen když se rychlost vlaku a s ním i výše tónu změnila.

Figure 3.1: Sample text of *Reader's Digest* corpus

Help
Digital Certificate Manager
Your web browser will display several windows to help you complete the installation of the certificate. A certificate for your Certificate Authority was created and stored in the default Certificate Authority key ring file. Clients must install the certificate to make use of the security provided by the certificate. To install this certificate on your browser, click the following link. Clients must install the updated certificate to make use of the security provided by the certificate.

Nápověda
Správce digitálního certifikátu
Váš prohledávací program sítě zobrazí několik oken, která vám pomohou dokončit instalaci certifikátu. Certifikát pro vašeho Ověřovatele oprávnění byl vytvořen a uložen v předvoleném souboru klíčového řetězce Ověřovatele oprávnění. Klienti musí certifikát instalovat, aby mohli využívat zabezpečení poskytované certifikátem. Pro instalaci tohoto certifikátu na váš prohledávací program klepněte na následující propojení. Klienti musí instalovat aktualizovaný certifikát, aby mohli využívat zabezpečení poskytované certifikátem.

Figure 3.2: Sample text of *Computer Oriented* corpus

data, the SIMR performance can be substantially improved; however, in this experiment we simply adopted the French/English settings.

There was also a lot of manual work to do on this corpus. Every issue of this magazine contains only 30-60% of articles translated from English to the local language. We had to search the English version to find the corresponding articles that are in the Czech version. The translations in the Reader's Digest are mostly very liberal. They include many constructions with direct speech. Articles with culture-specific facts have been excluded.

We show a sample of parallel text in Figure 3.1. The *Reader's Digest* corpus is also included in the LDC distribution of PCEDT, version 1.0.

3.1.3 Computer Oriented Corpus

We also experimented with a technically-oriented Czech/English corpus from IBM. This is a huge and very good source of Czech/English parallel data, but for a very specific domain. This corpus consists of operating system messages and operating system guides. These are products of localization and translation of software from English to Czech. The translations are very literal and precise, and mostly translated sentence by sentence. You may see a sample of *Computer Oriented* corpora in Figure 3.2.

3.2 Prague Czech-English Dependency Treebank

The *Prague Czech-English Dependency Treebank* (**PCEDT**) is a project to create a Czech-English syntactically annotated parallel corpus motivated by research in the field of machine translation. Parallel data is needed for designing, training, and evaluation of both statistical and rule-based machine translation systems.

The PCEDT, version 1.0 was released by the *Linguistic Data Consortium*[2] (LDC) in 2004 [Cuřín et al., 2004].

When starting the PCEDT project, we decided to translate and annotate an existing syntactically annotated monolingual corpus, rather than to annotate both languages of an already existing parallel corpus of raw texts, since the latter option would have been more money and time consuming. The choice of the Penn Treebank as the source corpus was pragmatically motivated: firstly it is a widely recognized linguistic resource, and secondly the translators were native speakers of Czech, capable of high quality translation into their native language.

Since Czech is a language with a relatively high degree of word-order freedom, and its sentences contain certain syntactic phenomena, such as discontinuous constituents (non-projective constructions), which cannot be straightforwardly handled using the annotation scheme of Penn Treebank [Linguistic Data Consortium, 1999], based on phrase-structure trees, we decided to adopt for the PCEDT the dependency-based annotation scheme of the Prague Dependency Treebank – PDT [Linguistic Data Consortium, 2001].

In PCEDT version 1.0, about half of the Penn Treebank has been translated (21,628 sentences). The project aims to translate the whole Wall Street Journal part of the Penn Treebank.

The Prague Czech-English Dependency Treebank raised the interest of the linguistic community in several research areas. The methods and algorithms on the automatic process of transformation of the Penn Treebank annotation into dependency representations (both, analytical and tectogrammatical) were published in [Čmejrek et al., 2004a].

The exploitation of the PCEDT for the Machine Translation task was presented in [Čmejrek et al., 2004b, Čmejrek et al., 2005]. The general issues on building a parallel bilingual syntactically annotated corpus were described in [Čmejrek et al., 2003b, Cuřín et al., 2005].

A sample of Wall Street Journal text and its Czech translation is in Figure 3.3.

[2] `http://www.ldc.upenn.edu/`

Japan's production of cars, trucks and buses in September fell 4.1% from a year ago to 1,120,317 units because of a slip in exports, the Japan Automobile Manufacturers' Association said. Domestic demand continues to grow, but its contribution to higher production was sapped in September by the estimated 2% fall in imports, accompanied by a growing tendency for Japanese manufacturers to build vehicles overseas, according to the association. The association said domestic demand grew 8.8% in September. Demand has been growing consistently under the encouragement of pro-consumption government policies, an association spokesman said.

Výroba osobních automobilů, nákladních automobilů a autobusů v Japonsku za září poklesla oproti loňskému roku o 4,1% na 1 120 317 jednotek v důsledku snížení exportu, uvedla Asociace výrobců japonských automobilů. Domácí poptávka i nadále stoupá, avšak její přispění k vyšší produkci bylo v září omezeno odhadovaným 2% poklesem importu spolu s rostoucí tendencí japonských producentů vyrábět vozidla v zahraničí, jak uvedla asociace. Asociace uvedla, že domácí poptávka v září stoupla o 8,8%. Poptávka trvale stoupá za podpory prospotřebitelské vládní politiky, řekl mluvčí asociace.

Figure 3.3: A sample text of the Prague Czech-English Dependency Treebank.

3.2.1 The Prague Dependency Treebank Annotation

The Prague Dependency Treebank (PDT) is an project for manual annotation of substantial amount of Czech-language data with linguistically rich information ranging from morphology through syntax to semantics and pragmatics.

PDT version 1.0 contains manual annotation of morphology and (surface) syntax, sequel version 2.0 adds the underlying syntax and semantics, topic/focus, coreference and lexical semantics based on a valency dictionary to the surface syntax and morphology. Three main groups ("layers") of annotation are used:

- the **morphological layer**, where lemmas and tags are being annotated based on their context;
- the **analytical layer**, which roughly corresponds to the surface syntax of the sentence,
- the **tectogrammatical layer**, or linguistic meaning of the sentence in its context.

The PDT 2.0 contains Czech texts with interlinked morphological (2 million words), analytical (1.5 million words) and tectogrammatical annotation

(800 thousand words).

Dependency trees, representing the sentence structure as concentrated around the verb and its valency, are used for the analytical and tectogrammatical levels. The related linguistic theory was proposed by Functional Generative Description [Sgall et al., 1986].

The Morphological Layer

The annotation of Czech at the morphological layer is an unstructured classification of the individual tokens (words and punctuation) of the utterance into morphological classes (morphological tags) and lemmas. Since Czech is a highly inflective language, the tagset size used is 4257, with about 1100 different tags actually appearing in the PDT.

There are 13 categories used for morphological annotation of Czech: Part of speech, Detailed part of speech, Gender, Number, Case, Possessor's Gender and Number, Person, Tense, Voice, Degree of Comparison, Negation and Variant. Complete description of the morphological annotation can be found in [Hana et al., 2005].

For English, we adopted the Penn Treebank POS annotation.

The Analytical Layer

At the analytical layer two attributes are being annotated:

- (surface) sentence structure,
- analytical function.

A rooted dependency tree is being built for every sentence as a result of the annotation. Every item (token) from the morphological layer becomes exactly one node in the tree, and no nodes (except for the single "technical" root of the tree) are added. Analytical functions, despite being kept at nodes, are in fact names of the dependency relations between a dependent (child) node and its governor (parent) node.

Coordination and apposition is handled using "technical" dependencies: the conjunction is the head and the members are its "dependent" nodes. Common modifiers of the coordinated structure are also dependents of the coordinating conjunction, but they are not marked as coordinated structure members. This additional "coordinated structure member" markup [`_Co`, `_Ap`] gives an added flexibility for handling such constructions.

Ellipsis is not annotated at this level (no traces, no empty nodes etc.), but a special analytical function [`ExD`] is used at nodes that are lacking their governor, even though they (technically) do have a governor node in the annotation.

There are 24 analytical functions used, such as `Sb` (Subject), `Obj` (Object, regardless of whether direct, indirect, etc.), `Adv` (Adverbial, regardless of type), `Pred,Pnom` (Predicate / Nominal part of a predicate for the (verbal) root of a sentence), `Atr` (Attribute in noun phrases), `Atv,AtvV` (Verbal attribute / Complement), `AuxV` (auxiliary verb - similarly for many other auxiliary-type words, such as prepositions [`AuxP`], subordinate conjunctions [`AuxC`], etc.), `Coord,Apos` (coordination/apposition "head"), `Par` (Parenthesis head), etc.

For details about analytical annotation scheme refer to [Hajičová et al., 1999], or see samples in Figures 3.6 and 5.5.

The Tectogrammatical Layer

The tectogrammatical layer is the most elaborated, complicated but also the most theoretically based layer of syntactico-semantic (or "deep syntactic") representation. For the purposes of the annotation of the PCEDT, we will sketch only the core components of the tectogrammatical annotation.

The tectogrammatical layer goes beyond the surface structure of the sentence, replacing notions such as "subject" and "object" by notions like "actor" [`ACT`], "patient" [`PAT`], "addressee" [`ADDR`] etc, but the representation still relies on the language structure itself rather than on world knowledge. The nodes in the tectogrammatical tree are autosemantic (content) words only. Dependencies between nodes represent the relations between the (autosemantic) words in a sentence. The dependencies are labeled by functors which describe the dependency relations. Every sentence is thus represented as a dependency tree, the nodes of which are autosemantic words, and the (labeled) edges name the dependencies between a dependent and its governor. Coordination and apposition is handled in the same way as on the analytical level.

Many nodes found at the morphological and analytical layers disappear (such as function words, prepositions, subordinate conjunctions, etc.). The information carried by the deleted nodes is not lost, of course: the relevant attributes of the autosemantic nodes they belong to now contain enough information (at least theoretically) to reconstruct them.

Ellipsis is also being resolved at this layer. Insertion of (surface-)deleted nodes is driven by the notion of *valency* and completeness: if a word is deemed to be used in a context in which some of its valency frames applies, then all the frame's obligatory slots are "filled" (using regular dependency relations between nodes) by either existing nodes or by newly created nodes, and these nodes are annotated accordingly.

Further description of the tectogrammatical annotation scheme can be found in [Böhmová et al., 2005], see also tectogrammatical trees in Figures 3.9, 3.6, and 5.6.

3.2.2 Schema of PCEDT Annotations

Figure 3.4 shows schema of data in the parallel Czech-English treebank with dependency annotation. Full arrows denote manual procedures and dotted arrows automatic conversions.

These are the steps performed to create the PCEDT (the step number corresponds to arrow labels in Figure 3.4):

1. Besides the syntactical annotation, the Penn Treebank also contains part of speech tags. Plain text format can be easily derived from these.

2. Translation into Czech was done by human translators. They were asked to translate the text sentence by sentence where possible and to preserve the sentence ID references to ease the sentence alignment procedure.

3. For the automatic annotation of Czech parts into analytical and tectogrammatical representations we used the tools and resources available at IFAL, as explained in Section 3.2.3.

4. The automatic process of transformation of the Penn Treebank annotation of English into both analytical and tectogrammatical representations is described in Section 3.2.4.

5. As a "gold standard" for the evaluation of various experiments and for the automatic transformations itself, a subset of sentences were manually annotated on the tectogrammatical level, see Section 3.2.5.

6. For evaluation purposes, the same subset of sentences was retranslated back from Czech into English by 4 different translator offices, see Section 3.2.6.

The corpus was divided into three parts: *training*, *development*, and *evaluation*. Development and evaluation parts contains together 515 sentences. The same set of sentences were manually annotated (step 5) and retranslated (step 6).

The summary on the data included on PCEDT can be found in Table 3.3.

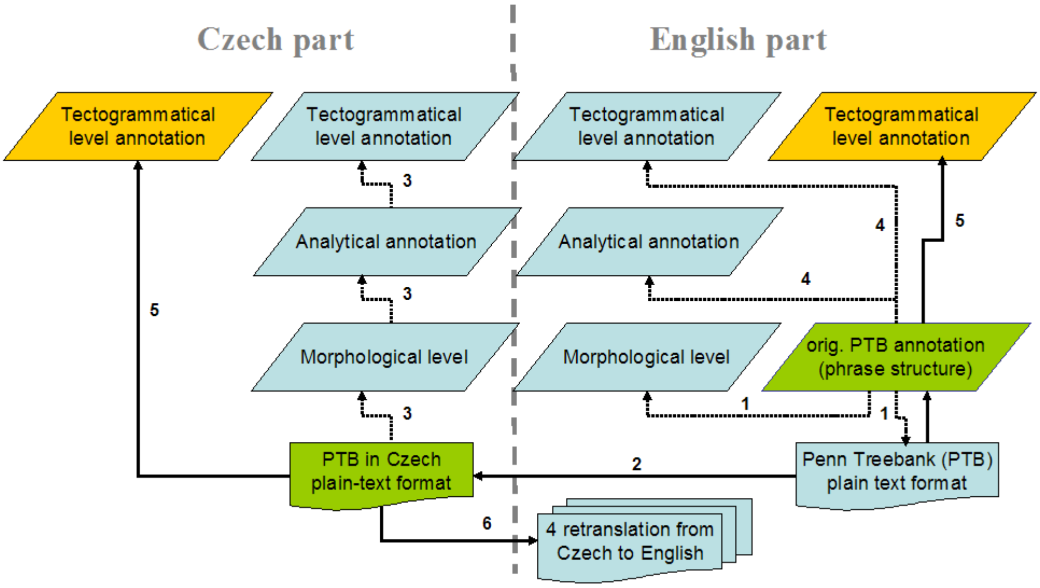
Czech part
English part
Tectogrammatical level annotation
Tectogrammatical level annotation
Tectogrammatical level annotation
Tectogrammatical level annotation
Analytical annotation
Analytical annotation
Morphological level
Morphological level
orig. PTB annotation (phrase structure)
PTB in Czech plain-text format
Penn Treebank (PTB) plain text format
4 retranslation from Czech to English
1
2
3
4
5
6

Figure 3.4: Schema of data resources of Czech-English dependency treebank included on PCEDT. Full arrows denote manual procedures and dotted arrows automatic conversions.

3.2.3 Automatic Annotation of Czech

Morphological Tagging and Lemmatization of Czech

The Czech translations of the Penn Treebank sentences were automatically tokenized and morphologically tagged. Each word form was assigned a base form (lemma) using the tagging tools by [Hajič and Hladká, 1998].

First, morphological analysis assigns to each word in the sentence a list of possible tags and lemmas. Then, based on the context, tags are disambiguated by a maximum-entropy tagger, after which a unique lemma is selected. A sample PCEDT sentence output of this annotation procedure is shown in Figure 3.5. Details about Czech morphological tags can be found in [Zeman et al., 2005]. The current performance of the tagger is higher than 94% in correctly assigned and disambiguated tags [Hajič, 2004].

Analytical Annotation of Czech

The Czech analytical parsing was done by a statistical dependency parser for Czech, either Collins' parser [Hajič et al., 1998] or Charniak's parser [Charniak, 1999]. Both parsers were adapted to dependency grammar and trained on the Prague Dependency Treebank [Linguistic Data Consortium, 2001]. Success rates for the parsers are about 80% for Collins' parser and

Description of Data	*Size*
PTB Corpus: English part (# sentences)	
– manually annotated on tectogrammatical level	1,257
– automatically transformed into analytical & tectogrammatical levels	49,208
– retranslated by 4 different human translators for the purposes of quantitative evaluation	515
PTB Corpus: Czech part (# sentences)	
– manually annotated on tectogrammatical level	515
– automatically parsed into analytical & tectogrammatical levels	21,628
Reader's Digest corpus (# aligned segments)	58,656
Czech monolingual corpus (# sentences)	2,385,000
Dictionaries (# entry-translation pairs)	
Czech-English probabilistic dictionary	46,150
Czech-English dictionary of word forms	496,673
GNU/FDL English-Czech dictionary	115,929

Table 3.3: Data set sizes

form	*lemma*	*rich morph. tag*	
`Založení`	`založení`	`NNNS1-----A----`	*noun, neuter, sing., nominative*
`první`	`první`	`CrFS2----------`	*ordinal num., fem., sing., gen.*
`kanceláře`	`kancelář`	`NNFS2-----A----`	*noun, feminine, sing., genitive*
`na`	`na`	`RR--6----------`	*preposition, requiring locative*
`území`	`území`	`NNNS6-----A----`	*noun, neuter, sing., locative*
`Varšavské`	`varšavský`	`AAFS2----1A----`	*adjective, fem., sing., genitive*
`smlouvy`	`smlouva`	`NNFS2-----A----`	*noun, feminine, sing., genitive*
`dokazuje`	`dokazovat`	`VB-S---3P-AA---`	*verb, present, 3rd person, sing.*
`rozsah`	`rozsah`	`NNIS4-----A----`	*noun, inanim. masc., sing., acc.*
`některých`	`některý`	`PZXP2----------`	*pronoun, plural, genitive*
`změn`	`změna`	`NNFP2-----A----`	*noun, feminine, plural, genitive*
`ve`	`v`	`RV--6----------`	*preposition, requiring locative*
`Východní`	`východní`	`AAFS6----1A----`	*adjective, fem., sing., locative*
`Evropě`	`Evropa`	`NNFS6-----A----`	*noun, feminine, sing., locative*
`.`	`.`	`Z:-------------`	*punctuation*

Figure 3.5: Disambiguated output tags from the morphological analysis of the sample sentence "*Založení první kanceláře na území Varšavské smlouvy dokazuje rozsah některých změn ve Východní Evropě.*".

85% for Charniak's parser. These figures represent the number of correctly assigned dependencies between governing and dependent nodes in the test corpus. In fact, the number of correctly parsed sentences is much lower. Output of these parsers is a dependency tree structure, to which we have to assign analytical functions to obtain the analytical annotation. We used a module based on a C4.5 classifier implemented by Zdeněk Žabokrtský for the automatic analytical function assignment.

Tectogrammatical Annotation of Czech

When building a tectogrammatical structure, the analytical tree structure is converted into the tectogrammatical one. These transformations are described by linguistic rules [Böhmová, 2001]. Then, tectogrammatical functors are assigned by a C4.5 classifier. Reported results are about 77% of correctly assigned functors [Žabokrtský et al., 2002].

See trees in Figure 3.6 for a comparison of automatic analytical and tectogrammatical annotations.

3.2.4 Automatic Transformation of Penn Treebank Annotation

This section briefly describes the automatic conversion of Penn Treebank annotation into analytical representation, for more details refer to [Hajič et al., 2002].

The general transformation algorithm from phrase-tree topology into dependency one works as follows:

- Terminal nodes of the phrase tree are converted to nodes of the dependency tree.
- Dependencies between nodes are established recursively: The root node of the dependency tree transformed from the head constituent of a phrase becomes the governing node. The root nodes of the dependency trees transformed from the right and left siblings of the head constituent are attached as the left and right children (dependent nodes) of the governing node, respectively.
- Nodes representing traces are removed and their children are reattached to the parent of the trace.

The concept of the head of a phrase is important during the transformation described above. For marking head constituents in each phrase, we used Jason Eisner's scripts. In case of coordination, we consider the rightmost coordinating conjunction (`CC`) to be the head.

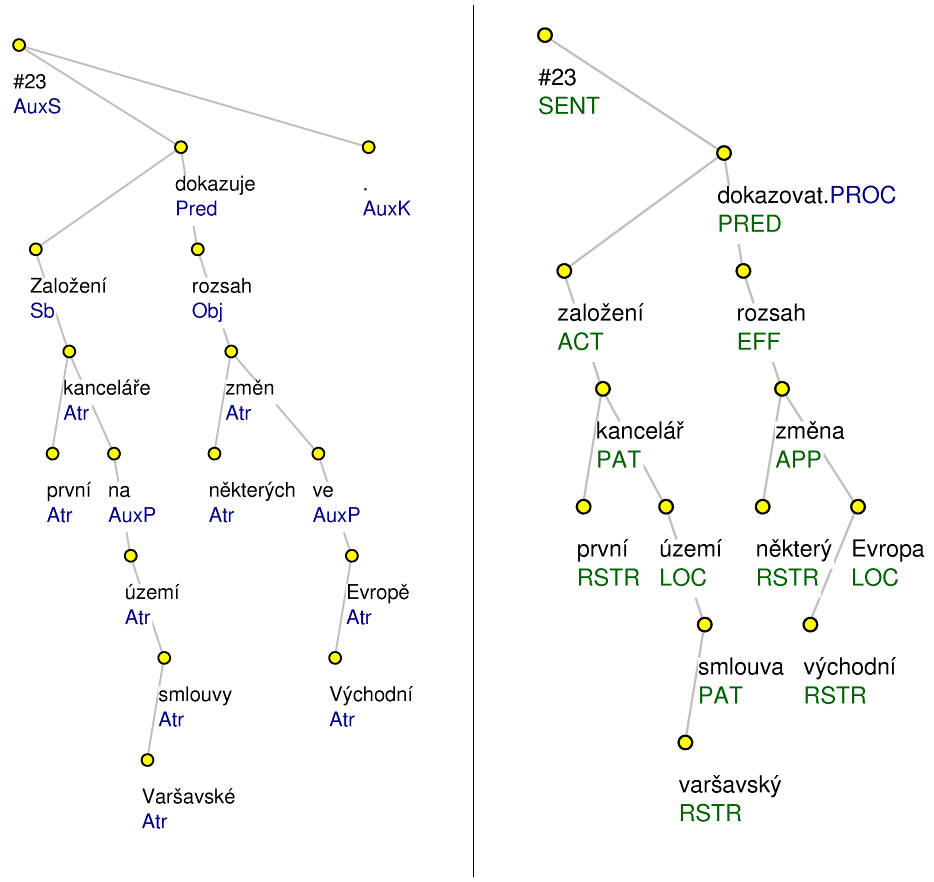

Figure 3.6: Automatic analytical and tectogrammatical annotations for the Czech sentence "*Založení první kanceláře na území Varšavské smlouvy dokazuje rozsah některých změn ve Východní Evropě.*"

The treatment of apposition is a more difficult task, since there is no explicit annotation of this phenomenon in the Penn Treebank; constituents of a noun phrase enclosed in commas or other delimiters (and not containing `CC`) are considered to be in apposition, and the rightmost delimiter becomes the head.

English Analytical Dependency Trees

The information from both the phrase tree and the dependency tree is used for the assignment of analytical functions:

- Some function tags of a phrase tree almost unambiguously correspond to analytical functions in an analytical tree and can therefore be mapped to them: `SBJ` → `Sb`, {`DTV`, `LGS`, `BNF`, `TPC`, `CLR`} → `Obj`, or {`ADV`, `DIR`, `EXT`, `LOC`, `MNR`, `PRP`, `TMP`, `PUT`} → `Adv`.
- For assigning analytical functions to the remaining nodes, we use rules querying their local context (the node, its parent and grandparent) for POS and a phrase marker. For example, the rule `mPOS=MD | pPOS=VB | mAF=AuxV` assigns the analytical function tag `AuxV` to a modal verb headed by a verb.

In the PDT and Penn Treebank annotation schemes, the markup of coordinations, appositions, and prepositional phrases are handled by these steps:

- The analytical function, which was originally assigned to the head of a coordination or apposition is propagated to its child nodes by attaching the suffix `_Co` or `_Ap` to them and the head node gets the analytical function `Coord` or `Apos`, respectively.
- The analytical function originally assigned to a preposition node is propagated to its child and the preposition node is labeled `AuxP`.
- Sentences in the PDT annotation style always contain a technical root node labeled `AuxS`, which is inserted above the original root. The final punctuation mark is moved under this new root.
- The dependency annotation scheme requires lemmatization, i.e. assigning base forms to words. This task is done automatically using the *morpha* tool [Minnen et al., 2001].

You may compare the phrase structure and the derived analytical and tectogrammatical representations of a sample sentence from the Penn Treebank in Figures 3.7 and 3.8.

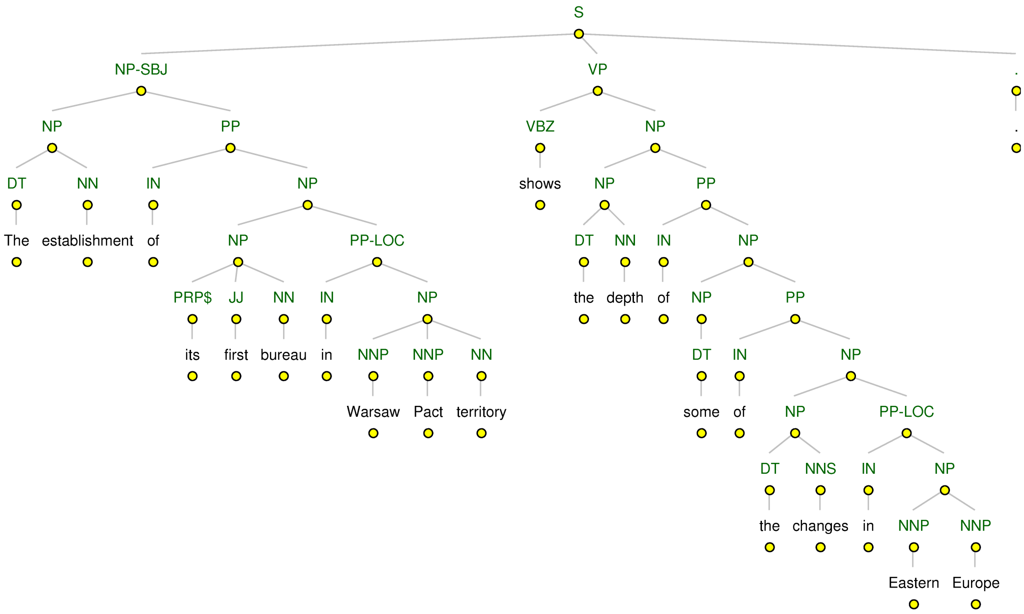

Figure 3.7: Original Penn Treebank annotation for the sentence "*The establishment of its first bureau in Warsaw Pact territory shows the depth of some of the changes in Eastern Europe.*"

English Tectogrammatical Dependency Trees

The transformation of the Penn Treebank phrase trees into their tectogrammatical representation consists of a structural transformation and an assignment of both a tectogrammatical functor and a set of grammatemes to each node.

At the beginning of the structural transformation, an initial dependency tree is created by a general transformation procedure described at the beginning of this section. However, functional (synsemantic) words, such as prepositions, punctuation marks, determiners, subordinating conjunctions, certain particles, auxiliary and modal verbs are handled differently. They are marked as "hidden" and information about them is stored in special attributes of their governing nodes (if they were to head a phrase, the head of the other constituent would become the governing node in the dependency tree).

The well-formedness of a tectogrammatical tree structure requires the valency frames to be complete: apart from the nodes that are realized on the surface, there are several types of "restored" nodes representing the non-realized members of valency frames (cf. the pro-drop property of Czech and

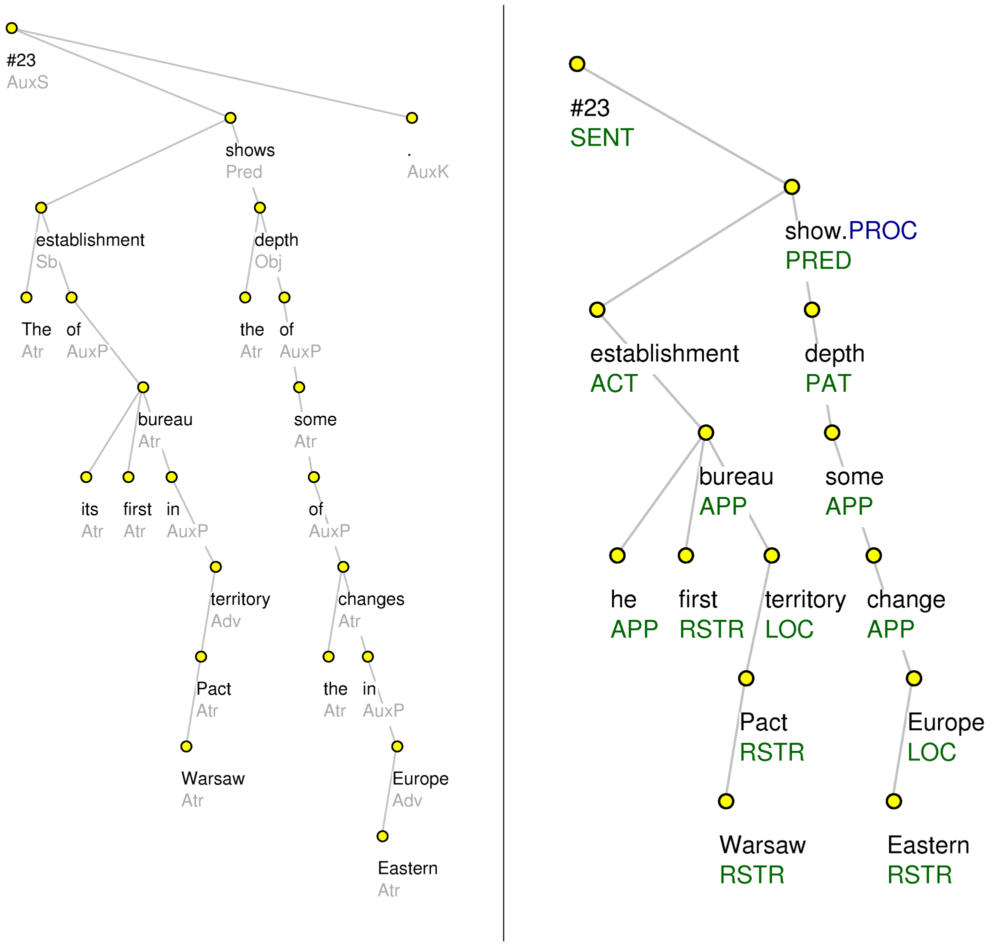

Figure 3.8: Automatic analytical and tectogrammatical annotations for the sentence "*The establishment of its first bureau in Warsaw Pact territory shows the depth of some of the changes in Eastern Europe.*"

verbal condensations using gerunds or infinitives both in Czech and English). For the reconstruction of some of these nodes, we can use traces, which allow us to establish coreferential links and restore general participants in the valency frames.

To assign tectogrammatical functors, we can use rules taking into consideration POS tags (e.g. `PRP` → `APP`), function tags (`JJ` → `RSTR`, `JJR` → `CPR`, etc.) and lemma ("not" → `RHEM`, "both" → `RSTR`).

Morphological grammatemes (e.g. tense, degree of comparison) are assigned to nodes of the tectogrammatical tree, based on the PennTreebank POS tags and reflecting basic morphological properties of English.

The automatic procedure briefly sketched above is described in detail

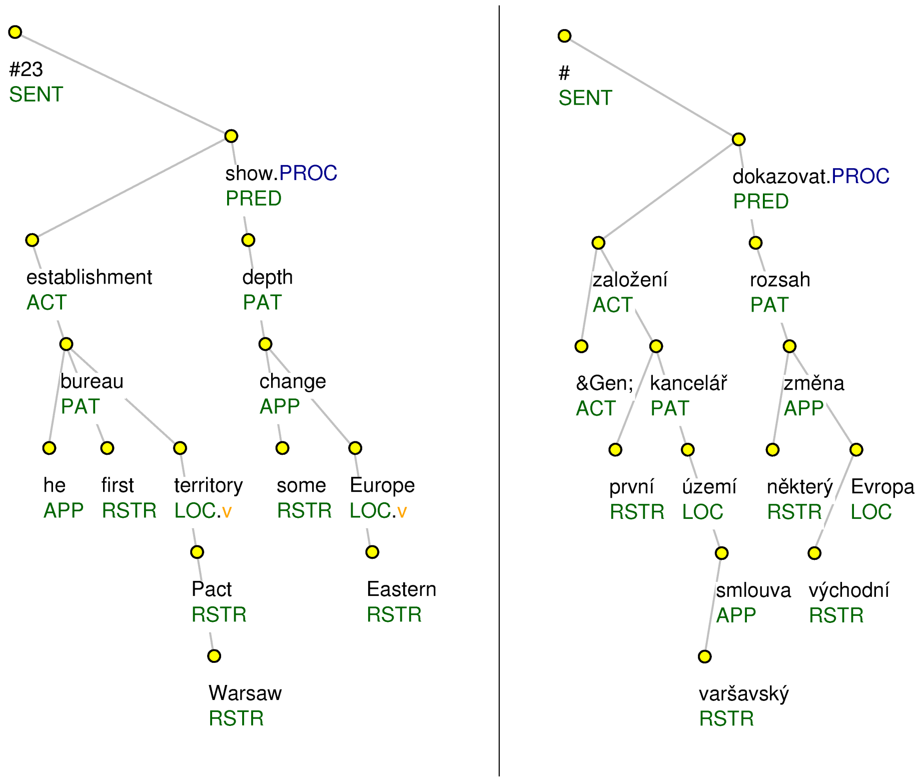

Figure 3.9: Manual tectogrammatical annotation of the sentence ”*The establishment of its first bureau in Warsaw Pact territory shows the depth of some of the changes in Eastern Europe.*” and its Czech counterpart.

in [Žabokrtský and Kučerová, 2002]. The quality of such a transformation, based on the comparison with manually annotated trees, is about 94% of correctly aimed dependencies and 82% of correctly assigned functors.

3.2.5 Manual Tectogrammatical Annotation of Czech and English

Since there were no guidelines for tectogrammatical annotation of English, and to acquire some initial experience before the work on the guidelines began, we made a “gold standard” tectogrammatical annotation of 1,257 sentences. This data is assigned morphological grammatemes (the full set of values), and the nodes were reordered according to the topic-focus-articulation (information structure). The manually annotated sentences comprise the whole development and evaluation test sets (515 sentences). Additionally, the Czech counterpart of the test sets has been manually annotated according to the guidelines for tectogrammatical annotation of Czech.

Original from PTB: *Kaufman & Broad, a home building company, declined to identify the institutional investors.*

Czech translation: *Kaufman & Broad, firma specializující se na bytovou výstavbu, odmítla institucionální investory jmenovat.*

Reference 1: *Kaufman & Broad, a company specializing in housing development, refused to give the names of their corporate investors.*

Reference 2: *Kaufman & Broad, a firm specializing in apartment building, refused to list institutional investors.*

Reference 3: *Kaufman & Broad, a firm specializing in housing construction, refused to name the institutional investors.*

Reference 4: *Residential construction company Kaufman & Broad refused to name the institutional investors.*

Figure 3.10: A sample English sentence from WSJ, its Czech translation, and four reference retranslations into English.

3.2.6 English Retranslation

For the purpose of quantitative evaluation methods for measuring performance of translation systems, such as NIST or BLEU, both development and evaluation test sets were retranslated from Czech into English by 4 different translator offices, two of them from the Czech Republic and two of them from the U.S.A.

This set might also be useful for a linguistic study of the variation between multiple translations. See Figure 3.10 for an example of reference translations of the sentence "*Kaufman & Broad, a home building company, declined to identify the institutional investors.*"

3.2.7 Other Resources

Reader's Digest Corpus

This corpus contains parallel raw text of 450 articles from the Reader's Digest, 1993–1996. Sentence pairs were aligned automatically by Dan Melamed's SIMR/GMA tool. Since the translations in this corpus are relatively free, only 43969 of 54091 aligned segments contain one-to-one sentence alignments. See also Section 3.1.2.

Czech Monolingual Corpus

The electronic text sources have been provided by the Institute of Czech National Corpus. Originally, all data comes from news articles which were published in the daily newspaper Lidové Noviny, years 1994–1995. The total amount of data is more than 39M tokens (words + punctuation) in about 2,385K sentences.

Dictionaries

The Prague Czech-English Dependency Treebank (PCEDT) includes the following translation dictionaries:

- Czech-English probabilistic dictionary, see Section 4.2.1.
- Czech-English dictionary of word forms, see Section 4.2.2.
- Publicly available English-Czech dictionary[3] provided under GNU/FDL (Free documentation license) [Svoboda, 2004].

[3] The version of GNU/FDL dictionary included on PCEDT was downloaded on 12th February 2004 and contains 115,929 entry-translation pairs

Chapter 4

Translation Dictionaries

Some people believe that translation is based on searching for words in a dictionary, but whoever has really tried it has found that it is more difficult to choose the right word among five or ten possibilities for one single entry.

Let us show an example of a backward translation of a Czech word ’*akcie*’ from Figure 4.1. The word ’akcie’ in Czech has an exact meaning and is used just in the financial domain. In the dictionary we can found “only” four English translations: ’*funds*’, ’*share*’, ’*stock*’, and ’*utility*’, but if you translates each of these four English words back into Czech, using all their Czech equivalents, you get 58 different words! Many of them such as for ’*hůl*’ *(transl. stick)* or ’*fiala*’ *(transl. gillyflower)* have nothing to do with the original meaning.

↓ ***akcie***	
funds →	fondy, státní půjčky, peněžní prostředky, základní jmění, dluhopisy, cenné papíry, *akcie*, kapitál, hodnoty, hotovost
share →	podíl, díl, kvóta, *akcie*, radlice pluhu, příděl, účast, čepel, ostří
stock →	původ, kmen, sklad, špalek, kůl, hůl, zásoba, vklad, jistina, límec, fiala, lodyha, stvol, stonek, cenné papíry, dobytek, masový výtažek, rod, pen, pařez, břevno, podpora, podložka, kláda, materiál na skladě, držadlo, kapitál, *akcie*, nákrčník, renty, palice, fond, snop, plemeno
utility →	užitečnost, funkčnost, *akcie*, utilita, prospešnost, štěstí, veřejné služby

Figure 4.1: Backward translations of the word ’akcie’

Probably the most useful outcome of machine translation technology for humans, represented in this case by professional translators, is in using *translation memory* which stores whole sentences translated by the translator or by his colleagues and when the same or a very similar sentence has to be translated again, it is automatically proposed by the translation memory tool. The optimal machine translation would be based on translation of the whole sentences. In fact, the statistical machine translation systems aim exactly at this model, but unfortunately even a small error in the decoding process can result from the human point of view in a totally unstructured or unreadable sentence (in the MT community known as '*translation soup*'). On the other hand, classical or rule-based translation systems can produce translations which on first sight seem to be perfect and syntactically acceptable, but with inadequate or misleading translation of single words or expressions.

The work described in this chapter suggests directions how to improve dictionaries used by humans or in the latter (rule-based) MT systems, see [Cuřín and Čmejrek, 1999, Cuřín and Čmejrek, 2001] for more discussion on this task.

4.1 Filtered and Weighted Dictionary for Transfer

When translating from a source language to a target language, both the human translator and the translation system would like to have a dictionary covering as much source-language words as possible, but on the target-language side they would prefer to have only one choice or a list of translations with a clear description of its semantic meaning or of its expected context. This requirement is unrealistic while translating general texts and for a literary translation of fiction but it could be applied to a restricted-domain translation.

This section explains a process of creating a translation dictionary used in the simple transfer of the Dependency-based Machine Translation system described in Section 5.2.

We expect a translation from Czech to English, we reference the Czech words in the dictionary as *entries* and English words as *translations* below. For the filtering of manual dictionaries we used the following **assumptions**:

1. No filtering on the entry side (Czech) is needed. If the English translation does not fit a particular domain and another entry/translation pair is found in the parallel corpus, we expect that this pair will obtain a higher probability by the GIZA++ training (Section 4.1.2).

2. If there is only one (English) translation for a particular (Czech) entry, this pair remains in the dictionary and the same expectation as in *assumption 1* applies.

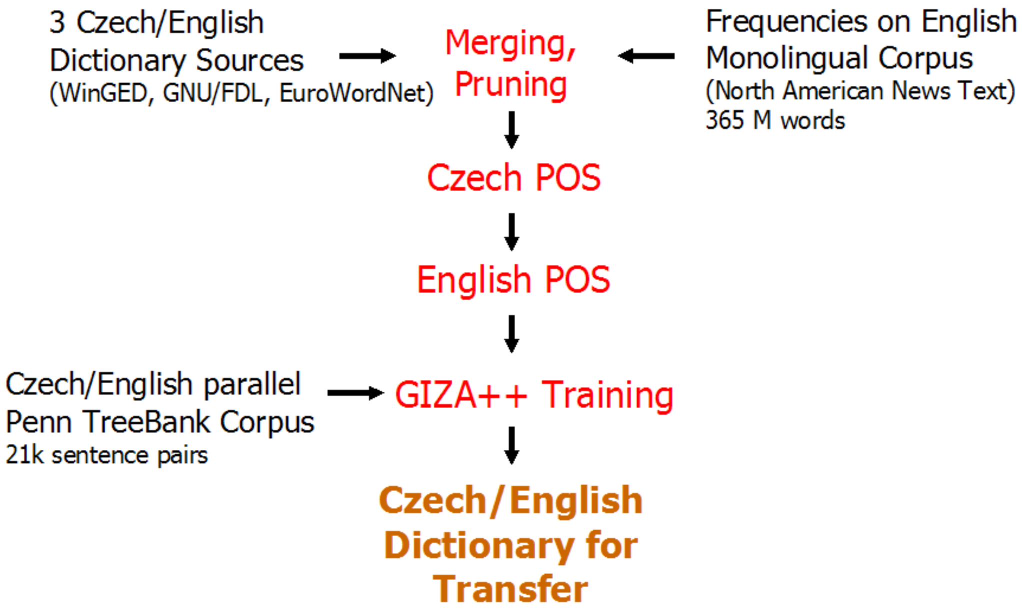

Figure 4.2: Dictionary Filtering and Weighting

3. Translations which were not found in a huge monolingual English corpus of the domain can be removed from the list.

4. If a particular English translation occurs in all input manual dictionaries, it is expected that it better represents the entry's specific meaning than another translation which was found only in one input dictionary.

5. Manual dictionaries have different "importance" weights. For example occurrence of an entry/translation pair in the same synsets of WordNet is more significant than the presence in a regular dictionary. See weights of dictionaries in the Table 4.1

6. In most cases the POS tags of the English translation can be disambiguated by the POS tag of the corresponding Czech entry. This fortunately applies particularly to the Czech-English language pair.

Figure 4.2 show the schema of the transfer dictionary filtering and weighting.

dictionary	*#entries*	*#transl*	*weight*
EuroWordNet	12,052	48,525	**3**
GNU/FDL	12,428	17,462	**2.5**
WinGED	16,296	39,769	**2**
merged	33,028	87,955	—

Table 4.1: Dictionary parameters and weights

4.1.1 Manual Dictionary Filtering

There were three different sources of Czech-English manual dictionaries available, two of them were downloaded from the Web (WinGED[1], GNU/FDL[2]), and one was extracted from Czech and English EuroWordNets[3]. See dictionary parameters in Table 4.1.

For the subsequent use of these dictionaries for a simple transfer from the Czech to the English tectogrammatical trees (see Section 5.2), a relatively huge number of possible translations for each entry[4] had to be filtered. The aim of the filtering is to exclude synonyms from the translation list, i.e. to choose one representative per meaning.

This merged dictionary consisting of entry/translation pairs (Czech entries and English translations in our case) was enriched by the frequencies from large English monolingual corpora and English POS tags obtained from the English morphological analyzer [Ratnaparkhi, 1996] on the translations' part, and by Czech POS tags and lemmas on the entries' part.

We select a few relevant translations (refered to as '*dictionary selected*' below) for each entry taking into account the weight of the source dictionary, the frequencies from the English monolingual corpora, and the correspondence of the Czech and English POS tags, as denoted in *assumptions 1 to 6*.

4.1.2 Scoring Translations Using GIZA++

To make the dictionary sensitive to a specific domain, which is in our case the domain of financial news, we created a probabilistic Czech-English dictionary by running GIZA++ training (translation models 1–4, see Section 2.2) on the training part of the Czech-English parallel corpus from PCEDT extended by the parallel corpus of entry/translation pairs from the merged

[1] http://www.rewin.cz/

[2] http://slovnik.zcu.cz/

[3] http://www.illc.uva.nl/EuroWordNet/

[4] For example for the WinGED dictionary it is 2.44 translations per entry on average, and excluding 1-1 entry/translation pairs even 4.51 translations/entry.

entry *(POS)*			
translation	*POS*	*probability*	*selection*
zesílit (V)			
increase	V	0.327524	DG**F**
reinforce	V	0.280199	DG**F**
amplify	V	0.280198	DG**F**
re-enforce	V	0.0560397	G
reenforce	V	0.0560397	G
výběr (N)			
choice	N	0.404815	DG**F**
selection	N	0.328721	DG**F**
option	N	0.0579416	G
digest	N	0.0547869	G
compilation	N	0.0547869	G
alternative	N	0.0519888	-
sample	N	0.0469601	-
selekce (N)			
selection	N	0.542169	DG**F**
choice	N	0.457831	DG**F**
trhat se (V)			
break	V	0.0853961	G**F**
split	V	0.0825296	G**F**
tear	V	0.080012	DG**F**
rip	V	0.0784138	G
scatter	V	0.07485	G
spread out	V	0.0748499	-
disperse	V	0.0748499	-
rend	V	0.0748498	-
dissipate	V	0.0748498	-
rive	V	0.0748498	-
cleave	V	0.0748498	-
jerk	V	0.0748498	-
twitch	V	0.0748498	-

Figure 4.3: Filtering of the Czech-English dictionary used in the DBMT transfer. In the *selection* column, 'D' means '*dictionary selected*', 'G' means '*GIZA++ selected*', and '**F**' indicates the '*final selection*', i.e. those translation are used in the transfer.

Number of entries	108939
Number of translations	159027
↦ *selected by both selectors* [DGF]	70160
↦ *selected by GIZA++ selector* [GF]	65590
↦ *selected by dict. selector* [DF]	23277
Avg. translations per entry	1.46

Table 4.2: Statistics of the final dictionary for transfer

manual dictionary (i.e. manual dictionaries before filtering). As a result, the entry/translation pairs seen in the parallel corpus of WSJ became more probable.

For entry/translation pairs not seen in the parallel text, the probability distribution among translations is uniform. The translation is '*GIZA++ selected*' if its probability is higher than a threshold, which is in our case set to 0.10 or if there is no translation with this probability, we select the five-best translations.

The *final selection* contains translations selected by both the dictionary and the GIZA++ selectors. In addition, translations not covered by the original dictionary can be included in the final selection if they were newly discovered in the parallel corpus by GIZA++ training and their probability is significant (higher than the most probable translation so far). Also entry/translation pairs with high weight from the manual dictionaries which were not found in the parallel corpus and included with fixed probability 0.01 as a back-off. See sample of the dictionary in Figure 4.3 and statistics in Table 4.2.

The translations from the final selection are used in the transfer module of the Dependency-tree machine translation system (Section 5.2).

4.2 Dictionaries included on PCEDT

4.2.1 Probabilistic Dictionary

The same schema for the dictionary creation and weighting as was described in the previous section was used for building a smaller dictionary which could be included in the PCEDT.

The Czech-English probabilistic dictionary was compiled as the translation of the words occurring in the training part of the PCEDT extended by words that occur more than 100 times in the Czech National Corpus (455M words). For the translation of this set of words we used the same three Czech-English manual dictionaries: WinGED, GNU/FDL, and EuroWordNet. We

f	e	$\mathbf{P}(e\|f)$	$\mathbf{P}(f\|e)$	$c(f)$	$c(e)$
dostavit_se#V	appear#V	0.345995	0.00955128	9369	14833
dostavit_se#V	show_up#V	0.267303	0.49808	9369	11084
dostavník#N	coach#N	1.0	0.0622594	727	6598
dostačit#V	suffice#V	1.0	0.264106	1382	366
dostačující#J	adequate#J	0.503527	0.073797	2377	9968
dostačující#J	sufficient#J	0.496473	0.0866228	2377	10274
dostih#N	race#N	0.693047	0.0541558	7605	30000
dostihnout#V	attain#V	0.17085	0.110701	1970	587
dostihnout#V	catch_up#V	0.256236	0.166027	1970	784
dostihnout#V	hunt_down#V	0.256236	0.24904	1970	591
dostupnost#N	availability#N	0.591496	0.288468	3783	3360
dostupný#J	accessible#J	0.0708691.0	0.49808	16169	2683
dostupný#J	affordable#J	0.0531518	0.74712	16169	1875
dostupný#J	available#J	0.82171	0.457435	16169	67563
dostát#V	comply#V	0.197878	0.153256	3132	5082
dostát#V	keep#V	0.139295	0.00719223	3132	55924

Figure 4.4: Sample of Czech-English probabilistic dictionary.

included only translations that occurred in at least two of the three dictionaries or the frequency of which is significant in the English North American News Text Collection (310M words).

POS tag and lemma were added to each Czech entry. If possible, we selected the same POS for the English translation, otherwise the most frequent one. Again, by training GIZA++ translation model on the training part of the PCEDT extended by the obtained entry-translation pairs, we created a probabilistic Czech-English dictionary more sensitive to the domain of financial news specific for the Wall Street Journal.

The resulting probabilistic dictionary contains 46,150 entry-translation pairs. See sample in Figure 4.4. $c(f)$ is the count of Czech lemma in the Czech National Corpus, $c(e)$ is the count of English words (with disambiguated POS) in the English North American News Text Collection.

4.2.2 Dictionary of Word Forms

Since Czech is highly inflective, the PCEDT also comprises a translation dictionary of word forms containing pairs of Czech and English word forms agreeing in appropriate morphological categories (such as number and person). This dictionary contains 496,673 entry-translation pairs and is suited for experiments in statistical machine translation based on raw parallel texts.

Czech form	*English form*	*comment*
bankéř	banker	*nominative sg.*
bankéře	banker	*genitive + accusative sg.*
bankéře	bankers	*accusative pl.*
bankéřem	banker	*instrumental sg.*
bankéři	banker	*dative + vocative + locative sg.*
bankéři	bankers	*nominative + vocative pl.*
bankéřů	bankers	*genitive pl.*
bankéřům	bankers	*dative pl.*

Figure 4.5: Word-translation pair "bankéř" - "banker" in the dictionary of word forms.

In the sample in Figure 4.5 note that since Czech "bankéři" is ambiguous for singular dative, vocative, or locative, and for plural nominative or vocative it can be translated into English in both singular and plural.

4.3 Terminological Dictionary Extraction from Computer Corpus

This section summarizes the results of an experiment done on the Computer Corpus – parallel text from localization of IBM operating systems and manuals, more details on this task can be found in [Cuřín and Čmejrek, 2001].

The assignment of this work was to create a terminological dictionary of the domain which would help human translators to be consistent in the terminology. In this experiment we simplified the task by limiting the 'terminology' only to noun groups.

We aim at a terminological dictionary, i.e. a dictionary containing also translations consisting of more than one word. For example, the single word "typewriter" (in English) is translated by a noun group consisting of two words "psací stroj" (lit. 'writing machine') in Czech. On the other hand, the English group "construction worker" corresponds to a single Czech word "stavbař". We model a tool which is capable of handling such cases.

The idea is to concatenate words of potential groups into one string, i.e. to consider these constructions as single "words". Identification of groups is based on a simple regular grammar. Grammar rules can be modified by the user. The grammar used in our case identifies noun groups (word sequences which consist of nouns, adjectives and some auxiliary words). Only continuous sequences of words are considered to be noun groups.

Czech is an inflective language with a lot of word forms, and in English

basic form		original forms
integrovaný systém souborů	←	*integrovaný systém souborů*
	←	*integrovaného systému souborů*
	←	*integrovanému systému souborů*
	←	*integrovaný systéme souborů*
	←	*integrovaném systému souborů*
	←	*integrovaným systémem souborů*
integrated file system	←	*the integrated file system*
	←	*integrated file system*
	←	*a integrated file system*

Figure 4.6: Conversion of groups into their basic form

contrastively, one word form corresponds to more POS categories. That is why we decided to proceed as follows:

- Czech groups and words are converted into their basic forms (the basic form means the first case of singular or plural for nouns, adjectives and pronouns, and the infinitive for verbs)
- English nouns and adjectives are distinguished from other POS categories
- definite and indefinite articles are removed from English groups (there is no equivalent for the category of an article in Czech)

For an example, see Figure 4.6.

The tagging of Czech and English texts and the conversion of Czech word forms were done by the BH tagging tools (Hajič, Hladká, 1998).

Marking of potential groups in a sentence is done separately for each language. Noun group identification algorithm works in two passes through the sentence. All possible groups in the sentence are identified in the first pass. During the second pass the algorithm searches for such combination of groups that:

- do not overlap in one combination,
- cover the maximum number of words in the sentence,
- the number of groups in the combination is minimal.

English: The device driver indicates a hardware failure of equipment.

```
&_device_driver_# indicates &_hardware_failure_of_equipment_# .
```

Czech: Ovladač zařízení zjistil technickou závadu přístroje.

```
&_ovladač_zařízení_# zjistit &_technická_závada_přístroje_# .
```

Figure 4.7: Sentence with marked groups

If there is still more than one such combination, one of them is chosen randomly. Parallel sentences with marked groups are shown in Figure 4.7. Concatenated noun groups or nouns (one word groups) are delimited by the symbols `&` and `#`.

Thanks to the fact that the learning of the dictionary is based on probabilistic methods, we have discretion in group identification. Even if some groups are marked incorrectly, they are eliminated by the probabilistic algorithm which handles a big amount of mostly good data.

Once the data is prepared, the translation dictionary training procedure can start.

4.3.1 Significance Filtering and the Evaluation of Results

The training procedure described in the previous section results in a probabilistic dictionary which assigns translation probability to every pair of Czech and English words which have ever been seen together in corresponding sentences. It is necessary to "clean up" the probabilistic dictionary by filtering out most of the translations to produce a useful dictionary. An obvious solution to reduce translations is to set a threshold on probabilities. Absolute thresholds work poorly, we use them only for rough pruning of translations with negligible probabilities.

The principle for significant filtering is to find a combination of just a few filtering criteria that affects the quality of the representative sample of the dictionary in the best way. This combination is used to filter the whole dictionary.

At the beginning we set the dictionary quality indicators and manually mark the quality of translations in the representative sample of the dictionary. We use two obvious quality indicators *Precision* and *Recall*, and a third one, *Share*, defined as follows:

Let $\mathcal{T}$ be a set of all translations in the input dictionary, $\mathcal{G}$ be a set of good translations for corresponding entries (i.e. these translations were marked as good by hand in the representative part of the dictionary), and $\mathcal{S}$ be a set

criterion	description
Thd(p)	Only translations accounting for the top of the threshold p are retained.
MC(n)	Works only for entries with low occurrence. Translations having count higher than n are excluded if they have not been selected as groups. Translation probabilities for each entry are recomputed.
MPr(p)	Translations with the translation probability lower than p are excluded. Translation probabilities for each entry are recomputed.
NG(p)	Works only for entries selected as noun groups. Translations with translation probabilities lower than p are excluded if they haven't been selected as noun groups. Translation probabilities for each entry are recomputed.

Table 4.3: Definition of filtering criteria

of translations which were marked as good by the combination of filtering criteria (i.e. these translation were marked as good by the automatic filtering). We denote the translation probability of a particular translation (x) by $\Pr(x)$. Let $\mathcal{S}^*$ and $\mathcal{G}^*$ be similar sets including only very good translations. Good translations in contrast to very good ones are acceptable only in some context.

$$\mathsf{Precision}(\mathcal{T}) = \frac{card(\mathcal{S} \cap \mathcal{G})}{card(\mathcal{S})} \tag{4.1}$$

$$\mathsf{Share}(\mathcal{T}) = \frac{\sum_{x \in \mathcal{S} \cap \mathcal{G}} \Pr(x)}{\sum_{x \in \mathcal{S}} \Pr(x)} \tag{4.2}$$

$$\mathsf{Recall}^*(\mathcal{T}) = \frac{card(\mathcal{S}^* \cap \mathcal{G}^*)}{card(\mathcal{G}^*)} \tag{4.3}$$

The quality indicator Precision grows with the elimination of bad translations. Share is a weighted Precision which takes into account probabilities assigned to translations by the training procedure. Recall indicates a success in recognizing manually marked good translations by the automatic filtering.

The representative sample of the dictionary was about 4% of all entries.

Filtering criteria used are defined in Table 4.3.

Applying miscellaneous combinations of filtering criteria (changing the order of filtering criteria or criterion thresholds) we observe progress in dictionary quality indicators. An example can be seen in Table 4.4, where we show dictionary quality rates for English-Czech dictionaries (Recall*/ Precision/ Share). On the first line there are the rates for the input dictionary (that is 100% for Recall*). The combination of filtering criteria with balanced

Combination of Criteria	Cz-En dict R*/P/S (v %)	En-CZ dict R*/P/S (v %)
Thd(0.85) ∼ input dictionary	100.0/27.7/58.7	100.0/39.3/66.0
Thd(0.7)	98.4/41.1/66.2	97.3/51.8/72.4
Thd(0.7) → MPr(0.08)	90.1/57.7/73.8	93.8/59.7/74.7
Thd(0.7) → MPr(0.05) → MPr(0.09)	90.1/60.1/74.8	93.8/62.1/76.0
Thd(0.7) → MPr(0.05) → MPr(0.09) → NG(0.3)	89.3/75.0/85.0	93.8/75.7/84.6
MC(1800) → Thd(0.7) → MPr(0.05) → MPr(0.09) → NG(0.3)	91.7/81.9/90.7	95.5/76.8/86.6
Thd(0.7) → MC(1800) → MPr(0.05) → MPr(0.09) → NG(0.3)	93.4/82.5/91.0	94.6/78.4/86.9

Table 4.4: Criteria combination and their influence on dictionary quality indicators

	Recall*	Precision	Share
	Cz-En / En-Cz	Cz-En / En-Cz	Cz-En / En-Cz
All Words	86.8 / 83.8	75.1 / 74.8	86.4 / 84.2
Noun Groups Only	93.4 / 94.6	82.5 / 78.4	91.0 / 86.9

Table 4.5: Quality indicators in the output dictionary

values of dictionary quality indicators is chosen as optimal (the last row in Table 4.4). The whole dictionary is processed by this optimal combination of filtering criteria. Both output dictionaries (Cz–En and En–Cz) contains about 6,000 entries (see quality indicators in Table 4.5).

Figure 4.8 shows example of the filtered dictionary. Entries and translations marked by * were recognized as groups by our noun group extracting tool. Numbers in square brackets are frequency counts in the corpus. Numbers in round brackets are translation probabilities (normalized for each entry). Good translations are underlined.

In Figure 4.8 there is a sample of English-Czech computer oriented lexicon. For instance entries **map** and **map*** or **mark** and **mark*** distinguish verbs and nouns translations. An example of a common error is the translation of the entry **manual IPL***, where the noun group was not recognized in the corresponding Czech sentence.

The results of the dictionary extraction for the computer oriented corpora

manage [177] spravovat (0.47), řídit (0.31), správa* (0.22)

managed [21] řídit (0.36), spravovat (0.27), program (0.18), server/400 (0.18)

management* [37] management*(0.78), řízení* (0.22)

manager's maintenance operating* [10] operating (0.77), podrobnější informace* (0.23)

manager* [76] manager* (1.00)

manager software operating* [13] operating (0.34), nalézt (0.32), SC19 (0.18), program* (0.16)

manages [14] řídit (1.00)

managing [87] řízení* (0.36), správa* (0.27), spravující stroje* (0.22), spravující stroj* (0.16)

managing system* [13] řídicí systém* (1.00)

manual* [105] manual* (0.44), manuál* (0.36), příručka* (0.21)

manually [130] ručně (0.79), manuálně (0.21)

manuals* [11] příručky* (0.57), knihovna* (0.21), vyhledávání informací* (0.21)

manual installation* [13] ruční instalace* (1.00)

manual installation process* [22] proces ruční instalace* (0.44), ruční instalace* (0.41), proces* (0.15)

manual install display* [10] obrazovka manual install* (0.60), objevit (0.40)

manual IPL* [14] IPL (0.54), manuální (0.46)

manual mode* [26] režim manual* (1.00)

manufacturer* [10] výrobce* (0.83), zařízení IBM* (0.17)

many [404] mnoho (0.87), kolik (0.13)

map [12] mapovat (0.51), AS/400 (0.28), datové typy* (0.21)

map* [31] mapa* (0.68), map* (0.32)

mapped [22] mapovat (1.00)

mapping* [45] mapování* (0.45), macintosh (0.30), přiřazení* (0.25)

maps [19] mapy* (0.56), instalační (0.22), jeho (0.22)

maps* [10] mapy* (0.62), aplikace* (0.38)

margins* [13] okraje* (0.85), řádek* (0.15)

mark [19] označit (1.00)

mark* [18] označit (0.54), značka* (0.46)

marked [43] označit (0.83), označený (0.17)

marketing representative* [62] obchodní zástupce* (1.00)

marks [13] uvést (0.40), klíčové slovo* (0.40), uvozovky* (0.20)

mask* [35] maska* (0.59), maska podsítě* (0.41)

master* [13] master* (1.00)

master installation list* [50] hlavní instalační formulář* (1.00)

match [177] odpovídat (0.87), souhlasit (0.13)

match* [46] odpovídat (0.41), odpovídající protějšek* (0.31), shoda* (0.28)

matched [11] odpovídat (0.23), za (0.23), nalezený (0.18), další příkazy* (0.12), splňovat (0.12), vyhovět (0.12)

matches [56] odpovídat (0.85), souhlasit (0.15)

matching* [13] odpovídající* (0.63), odpovídat (0.37)

material* [30] materiál* (1.00)

materials* [11] materiály* (0.60), materiál* (0.40)

matrix [16] matice* (1.00)

max [41] max (0.79), maximálně (0.21)

maximum* [137] maximum* (0.52), maximálně (0.48)

maximum length* [18] maximální délka* (0.72), maximální délka parametru* (0.28)

Figure 4.8: Sample of English–Czech computer oriented dictionary extracted from a parallel corpus

are of unexpectedly high share (weighted precision) rates about 85% and for the terminology dictionary (that contains only noun groups) they are even better: 87%–91%.

Chapter 5

Experiments: Czech/English MT

If you think technology can solve your problems you don't understand technology
and you don't understand your problems.
Bruce Schneier

Since approximately 2002 the state of the art in machine translation has been lead by the statistical approach. We can identify two main impulses which accelerated this. First was the NLP Summer Workshop at JHU in 1999 [Al-Onaizan et al., 1999] which led to the public release of a software toolkit for SMT. The EGYPT toolkit and its descendants helped to start many new SMT research activities and catalyzed the dominance of the statistical approach. The second accelerator was the introduction of a widely accepted automatic evaluation metric – the BLEU score [Papineni et al., 2001] – by IBM in 2001.

The beauty of SMT is in its universality. Creating a translation system for a new language pair "just" requires collecting a couple of thousand parallel sentences for translation model training and as much as possible monolingual resources for language model training. It might be possible to create a baseline translation system in days or weeks depending on the availability of specific language data. But, of course, we can find also disadvantages: the trained statistical model is to some extent a black-box, the designer does not have direct influence on how a specific, let us say syntactic, phenomenon will be translated. For example, suppose we wish to express the preference for adjectives to follow nouns or to precede them in some language. A model that knows about parts of speech needs only one rule to apply this preference, but a generic (language-independent) model has to learn it for every noun-adjective pair in the corpus and just hope that some approximation based on automatically discovered word-classes will be capable of handling unseen pairs properly in most cases.

Adding new training sentences may have negative impact on the translation of a particular sentence even if it results in improvement on the whole test set. Also the word-to-word or n-gram-to-n-gram based nature of noisy channel models necessarily limits the flexibility and extensibility and makes it difficult to incorporate linguistic knowledge of an individual language.

We limit ourselves only to translation from Czech to English, focusing on exploitation of rich linguistic resources we have available for Czech.

We describe two experimental Czech to English translation systems: in section 5.1, we propose a method for how to improve the results of the SMT system by input language preprocessing using morphological and analytical analysis of Czech. Section 5.2 introduces the translation system based on dependency-trees, particularly on tectogrammatical parsing, discussing the possible advances in the incorporation of statistics automatically retrieved from parallel corpus.

5.1 Improving SMT by Linguistic Preprocessing

Let us start with a few observations about Czech and English language specifics.

Czech, as a Slavic language, is a highly inflectional and almost free word-order language. Most of the functions expressed in Czech by endings (inflection) are rendered by English word order and some function words.

For example, most Czech nouns or personal pronouns can form singular and plural forms in 7 cases. Most adjectives can form 4 genders, both numbers, 7 cases, 3 degrees of comparison, and can be either of positive or negative polarity (giving 336 possibilities for each adjective). In the PCEDT corpus there are 53,090 unique word forms in the Czech part against 27,291 forms in English. On the other hand, there are 10% more running words in English than in Czech in the corpus. Table 5.1 shows that while Czech has only 7.46 occurrences of one word form on average, medial English word form can be found more than twice as often in the English part of the PCEDT.

Czech is a pro-drop language[1]. This means that the subject pronoun (I, he, they) has usually a zero form. There are no definite and indefinite articles in Czech. English preposition equivalents can also be part of a Czech noun or pronoun inflection.

	English	Czech
running words	429013	396281
unique word forms	27291	53090
running words per unique form	15.72	7.46

Table 5.1: English vs. Czech Statistics on PCEDT

All these features create problems in automatic translation.

5.1.1 Baseline

The idea of linguistic preprocessing of Czech for the purpose of SMT was introduced by the author during the already mentioned 1999 NLP Summer Workshop at JHU [Cuřín and Peterek, 2000]. The training program for IBM translation models (introduced in Section 2.2) parameters called GIZA (later extended to GIZA++) was developed during the workshop Czech and English was among the first tested language pairs. We also did experiments with translations from Czech to English using the Alignment Templates system from University of Aachen [Och et al., 1999].

[1] A **pro-drop language** (from “pronoun-dropping”) is a language where pronouns can be deleted when they are in some sense pragmatically inferable.

As a baseline can be seen building of SMT system on a parallel corpus of without any special processing beside tokenization[2], both translation & evaluation systems usually ignore capitalization by converting the corpus to lowercase.

The next natural step in the case of translation from highly inflectional languages is stemming or lemmatization. By stemming we mean a simple preprocessing of a word by cutting off its inflection (ending) or even more simply taking n-first characters from each word. We consider lemmatization as part of the morphological analysis of Czech, lemma is a base form of a particular class of inflection. For example for nouns it is the nominative singular, for adjectives it is the nominative singular masculine, and for verbs the infinitive form. See the tokenized, stemmed and lemmatized version of our sample sentence "*Založení první kanceláře na území Varšavské smlouvy dokazuje rozsah některých změn ve Východní Evropě.*":

Tokenized: *založení první kanceláře na území varšavské smlouvy dokazuje rozsah některých změn ve východní evropě .*

Stemmed (4 chars): *zalo prvn kanc na územ varš smlo doka rozs někt změn ve vých evro .*

Lemmatized: *založení první kancelář na území varšavský smlouva dokazovat rozsah některý změna v východní Evropa .*

Even in the lemmatized version of the input we discard information about number, tense, gender, and other features. But, are not these attributes necessary to produce a useful translation? Asking this question we tried to investigate possibilities of preprocessing the Czech input to the form that would help word-to-word based translation models to produce better results. Below, such a form of Czech input is called *Czech'*.

5.1.2 *Czech'* – Czech closer to English

First, because of the disproportion between the average number of words in Czech and in English we introduce the idea of adding *artificial words* into Czech to places where we expect its English counterpart to appear. These artificial words correspond to pro-dropped proverbs, articles, and prepositions which are in Czech part of inflection.

An example of a Czech sentence with artificial words (in brackets) is given in Figure 5.1. The corresponding words in both languages are coindexed.

[2] Tokenization puts spaces (word-separators) between words and punctuation characters

I$_1$ am$_2$ convinced$_3$ that$_4$ *[]*$_5$ team$_6$ work$_7$ is$_8$ the$_9$ key$_{10}$
for$_{11}$ the$_{12}$ realization$_{13}$ of$_{14}$ ones$_{15}$ dreams$_{16}$

[I]$_1$ jsem$_2$ přesvědčen$_3$, že$_4$ *[the]*$_5$ týmová$_6$ práce$_7$ je$_8$ *[the]*$_9$ klíčem$_{10}$
ke$_{11}$ *[the]*$_{12}$ splnění$_{13}$ *[of]*$_{14}$ *[the]*$_{15}$ snů$_{16}$

Figure 5.1: Addition of artificial words into a Czech sentence

	English	Czech	*Czech’*	*Czech’* (w/o artif.)
running words	429013	396281	677157	404719
unique word forms	27291	53090	33655	33621
running words / form	15.72	7.46	20.12	12.04

Table 5.2: English, Czech, and *Czech’* Statistics on PCEDT

There is an artificial word *[I]* for first person, singular subject, large numbers of artificial articles and an artificial preposition *[of]* corresponding to the Czech genitive. There is a potential over-generation of artificial words as you can see in position 5. This over-generation can be compensated for in the translation or language models.

In the full *Czech’* there is information such as number or tense attached to a particular lemma which is expected to be relevant for English translation. This makes a distinction between singular and plural for nouns, tenses for verbs, grades for adjectives and adverbs, and all sorts of alternatives for pronouns. In Table 5.2 we can see how *Czech’* unique form and running form statistics better correspond to the English ones. The closest last column shows numbers without 34 unique forms of artificial words.

The *Czech’* version of the example sentence with the Czech original and referring English translation is here:

Original: *Založení první kanceláře na území Varšavské smlouvy dokazuje rozsah některých změn ve Východní Evropě .*

English reference: *The establishment of its first bureau in Warsaw Pact territory shows the depth of some of the changes in Eastern Europe .*

***Czech’*:** *#NtheS; založení-#NS; #Nprep2; #NtheS; první-#C; kancelář-#NS; na-#R; #NtheS; území-#NS; #Nprep2; #NtheS; varšavský-#A; smlouva-#NS; dokazovat-#VBZ; #NtheS; rozsah-#NS; #Pprep2; některý-#PP; #Nprep2; #NtheP; změna-#NP; v-#R; východní-#A; Evropa-#NS; .*

The input to the *Czech'* conversion procedure is the sentence parsed into a analytical tree, as shown in Figure 3.6.

5.1.3 Description of Preprocessing Rules

Let us briefly describe the rules for individual parts of speech. A full list of the rules applied can be found in Appendix A.3. By lemma we mean the basic form of the word in the lemmatized version, the adapted form for the *Czech'* version is referred to as *lemma'*. If the probability is shown in the examples, it refers to $P(e|f)$ or $P(f|e)$ (where e is the English n-gram and f is the *Czech'* n-gram) depending on if *Czech'* or English n-gram is written in the first column. The figures in this chapter are taken from the phrase model automatically built by GIZA++ and Pharaoh on a training part of the PCEDT corpus.

Nouns

Through the lemmatization of nouns, as well as the inflection for different cases, also the information about the number (singular/plural) is discarded, therefore we introduce different *lemma'* for singular and for plural, see Rule **N1** in the rules below:

N1 different *lemma'* for singular and plural

N2 if the noun does not govern a pronoun or cardinal numeral, the artificial article is added before the noun group (group of nouns and adjectives or ordinal numerals)

N3 if the noun is in the genitive, dative, locative or instrumental case and it is not governed by a preposition in the parse tree, the artificial preposition is added ahead the noun group (in prior of already inserted artificial article from the previous rule)

N4 if the noun is recognized as a proper noun, use lemma

The example for Rule **N1** shows the ideal case where for the *lemma' autor-#NP;* English plural forms and for *lemma' autor-#NS;* words in the singular were selected, the information about the number of the Czech word is taken from automatic morphological analysis (POS tagging and lematization).

```
autor-#NP; | authors | 0.333333
autor-#NP; | writers | 0.666667
autor-#NS; | author of | 0.111111
```

```
autor-#NS; | author | 0.666667
autor-#NS; | writer | 0.222222
```

Rule **N2** also uses information from the analytical analysis, the artificial word for the article is inserted only if the observed noun does not govern a pronoun or cardinal numeral. Let us look at the most probable pairs for artificial articles. The results listed below seems promising, the artificial article for the singular[3] (*&#NtheS;*) is with high probability aligned to English *the* or *a*, the possibility of how to determine if a definite or indefinite article has to be added is investigated in Section 5.2. This selection is based on contextual boundness in the tectogrammatical representation of the sentence, in our case the final selection of the "proper" article is passed to the language model. The artificial article may not be inserted if the noun is in a list of uncountable nouns.

```
&#NtheS; | the | 0.560819     &#NtheP; | the | 0.810066
&#NtheS; | a | 0.280108       &#NtheP; | its | 0.016439
&#NtheS; | 's | 0.0475481     &#NtheP; | their | 0.00834598
&#NtheS; | an | 0.0360514     &#NtheP; | some | 0.00278199
&#NtheS; | its | 0.00769995   &#NtheP; | other | 0.00227618
```

For the plural artificial article the competition is won by the English word *the*, other proposed translations with lower probability are also good candidates. The counterpart of the plural definite article is often not present in the English sentence, we have again to rely on the language model strength.

Rule **N3** adds an artificial preposition in front of a noun phrase groups only if a Czech preposition was not present in the original sentence. For the nominative, accusative, and vocative no artificial prepositions are inserted because we do not expect any corresponding preposition in the English sentence.

Figure 5.2 shows probabilities (first 4 cases in order) trained for the following artificial prepositions: genitive (*&#Nprep2;*), dative (*&#Nprep3;*), locative (*&#Nprep6;*), and instrumental (*&#Nprep7;*) and reverse probabilities for English prepositions *of, to, with*, and *by*. We can see strong correlation between the genitive case and the preposition *of* (ex. "*skupina studentů*" ↔ "*group of students*"), between the dative case and the preposition *to* (ex. "*dal dárek Marii*" ↔ "*[he] gave [a] present to Maria*"), and between the instrumental and the preposition *by* (ex. "*jel autobusem*" ↔ "*[he] went by bus*"). The alignment of *&#Nprep7;* and *with* is relatively weak because for the Czech instrumental case the preposition *s* (*s-#R;* in the Figure) is

[3] Singular or plural artificial articles were chosen according to the number of the noun.

```
    f   |   e   |  P(e|f)        e   |   f   |  P(f|e)
&#Nprep2; | of | 0.826121     of | &#Nprep2; | 0.590431
&#Nprep2; | to | 0.0593187    of | &#Nprep2; &#NtheS; | 0.13874
&#Nprep2; | for | 0.0547152   of | &#Nprep2; &#NtheP; | 0.11656
&#Nprep2; | by | 0.0230172    of | z-#r; | 0.0477533
&#Nprep3; | to | 0.819403     to | &#Vto; | 0.270468
&#Nprep3; | for | 0.0626866   to | na-#R; | 0.194758
&#Nprep3; | with | 0.016417   to | &#Nprep3; | 0.0573308
&#Nprep3; | the | 0.0074626   to | &#Nprep2; | 0.0470969
&#Nprep6; | of | 0.411371     with | s-#R; | 0.600211
&#Nprep6; | in | 0.183946     with | s-#R; &#Nprep7; | 0.051824
&#Nprep6; | on | 0.0769231    with | u-#R; | 0.0275680
&#Nprep6; | at | 0.0434783    with | u-#R; &#Nprep2; | 0.010576
&#Nprep7; | by | 0.663763     by | &#Nprep7; | 0.385238
&#Nprep7; | with | 0.085365   by | &#Nprep7; &#NtheS; | 0.10465
&#Nprep7; | to | 0.0766551    by | &#Nprep2; | 0.0884732
&#Nprep7; | as | 0.03223       by | &#Nprep7; &#NtheP; | 0.06218
```

Figure 5.2: Trained probabilities for prepositions.

used very often (ex. "*s radostí*" ↔ "*with pleasure*"). In the reverse table, the preposition "*s*" is also the most expected translation of *with*. The pair *with* ↔ "*s-#R; &#Nprep7;*" probably occurs when the Czech noun is by mistake not governed by the preposition "*s*" in the parse tree. The English word *to* can be either a part of the infinitive or a preposition. In the former case it will be aligned to the artificial word *&Vto;* which is more often then co-occurrence with Czech preposition "*na*".

Nouns which were recognized as proper nouns (Rule **N4**) are printed in the nominative case without any additional suffixes or artificial words. We consider nouns which have the first letter capitalized in both the original text and the lemmatized text to be proper nouns.

Verbs

The Czech and English systems of verbs are different. Czech distinguishes three verb tenses: past, present, and future, in English there are, besides these three basic tenses, three additional forms for each tense: progressive, perfect, and perfect progressive, which means twelve different verb tenses. In Czech, distinguished forms (inflections) are used for the grammatic person and gender. Czech also divides verbs according to their aspect - perfective[4] of imperfective[5], different aspects also have different lemmas. A list of the rules applied to verbs during the transformation to *Czech*' follows. A more detailed but rather technical description can be found again in Appendix A.3.

V1 different *lemma'* for different tenses, special *lemma'* for present tense, 3rd person, singular.

V2 if the verb is not governing a nominative noun, an artificial subject is added (artificial subjects differ for person, gender and number depending on the form of the verb)

V3 special handling for the auxiliary verb "*být*" (*to be*)

V4 adding an artificial word for negative verbs, infinitives and conditionals

Through rule **V1** different inflections for person and gender are discarded, we add a special ending according to the tense assigned to the Czech verb by a morphological tagging. We use the ending *-#VB;* for the present tense, *-#VBD;* for the past tense and *-#VBF;* for the future tense. If a verb is annotated as present, singular, 3rd case (ex. "*he works*"), the ending *-#VBZ;* is appended. See automatic alignments and probabilities for word "*říkat*" (*to say*):

```
říkat-#VB; | say | 0.55
říkat-#VB; | said | 0.25
říkat-#VB; | call | 0.1
říkat-#VBD; | said | 0.25
říkat-#VBD; | saying | 0.125
říkat-#VBD; | spoke | 0.125
říkat-#VBD; | telling | 0.125
říkat-#VBD; | tells | 0.125
říkat-#VBZ; | says | 0.821918
říkat-#VBZ; | said | 0.116438
```

Rule **V2** is applied because Czech is a pro-drop language where personal pronouns are usually omitted (they can be "pragmatically inferred" from verb inflection). The presence of a personal pronoun in Czech indicates emphasis,

[4] A **perfective verb** describes a one-time, finished action that took place in a certain time slot (e.g. *vypít* - to drink up, to finish drinking)

[5] A **imperfective verb** describes an unfinished or repeated action of a non-specified duration (e.g. *pít* - to drink, to be drinking)

```
být-#VBZ; | 's | 0.551948
být-#VBZ; | is | 0.15368
být-#VBZ; | was | 0.112554
být-#VBZ; | are | 0.0411255

být-#VB; | be | 0.487654
být-#VB; | are | 0.141975
být-#VB; | 're | 0.0946502
být-#VB; | is | 0.0679012

být-#VBD; | been | 0.394737
být-#VBD; | was | 0.204678
být-#VBD; | be | 0.19883
být-#VBD; | were | 0.111111
```

```
být-#VBF; | be | 0.348
být-#VBF; | going | 0.124
být-#VBF; | would | 0.088
být-#VBF; | will | 0.08
být-#VBF; | to be | 0.048
být-#VBF; | will be | 0.044
být-#VBF; | going to be | 0.036

být-#VC; | would | 0.728889
být-#VC; | though | 0.0177778
```

Figure 5.3: Trained $P(e|f)$ probabilities for the verb "*být*" (*to be*).

for example the sentence "*Přijde pozdě!*" can be translated as "*He will be late.*", adding the pronoun "*on*" (*he*), i.e. the sentence "*On přijde pozdě.*" would be rather translated as "*That's him, who will be late.*".

The verb "*to be*" (Rule **V3**) has special status in almost all languages, in Czech it can be part of the future tense (ex. "*budu pracovat*", transl. "*I will work*"), the past tense (ex. "*pracoval jsem*") or the conditional (ex. "*pracoval bych*"). See details in the Appendix and probabilities in Figure 5.3.

Personal Pronouns

As we mentioned in the previous subsection about verbs, the use of personal pronouns as a subject in Czech is not mandatory as in English. Subject pronouns in Czech are in the nominative case, for other cases, such as for the dative, the Czech pronoun is present, but it may be useful to add an artificial preposition as we did for nouns. An example sentence might be "*Dal to jí*":

```
Czech:   Dal to jí .
Czech':  &#Vsub3SM; dát-#VBD; on-#P-S; &#Pprep3; on-#P-S-F; .
English: He gave it to her .
```

Where *&#Vsub3SM;* denotes the artificial subject of the third person singular masculine, *dát-#VBD;* the verb *give* in the past tense, *on-#P-S;* the pronoun *he* in the singular, other gender (neither masculine nor feminine), *&#Pprep3;* the artificial preposition for the dative, and *on-#P-SF;* the pronoun *he* in the singular, feminine.

There are the rules:

PP1 different lemma for singular and plural

PP2 for third person singular, there is a different lemma for masculine animate, feminine and others (*he, she, it*)

PP3 if the pronoun is in the genitive, dative, locative or instrumental case, and it is not governed by a preposition in the parse tree an artificial preposition is added

Other Pronouns

The set of rules applied differs depending on the type of pronoun. The following types are distinguished: Relative Pronouns, Relative Possessive Pronouns, Reflexive Pronouns, Pointing Pronouns Interrogative Pronouns, Infinitive Pronouns, Negative & Undetermined Pronouns. A full list of the rules for pronouns can be find in the Appendix.

Adjectives and Adverbs

For both adjectives and adverbs we apply the following two rules:

A1/D1 artificial word(s) for second (*more*) and third grade (*the most*)

A2/D2 artificial word for a negative adjective or adverb

See table for the most probable pairs for n-gram "*&#Dthe; &#D3;*":

```
&#Dthe; &#D3; | the most | 0.259259
&#Dthe; &#D3; | most | 0.222222
&#Dthe; &#D3; | the best | 0.111111
&#Dthe; &#D3; | the lowest | 0.0740741
&#Dthe; &#D3; | are most | 0.0740741
&#Dthe; &#D3; | the hardest | 0.037037
```

5.1.4 SMT System Used for the Experiments

The *Czech'* preprocessing module can be utilized on arbitrary Czech input parsed into the analytical representation, the output of this module is a text-file with one sentence per line. Any SMT system capable of training and performing the translation on a sentence aligned text can be used for the experiments.

During the development of the system we used the Giza++ [Och, 2002] module for IBM Model 4 training and the ISI Decoder [Germann et al., 2001] for decoding. In Appendix A.2 we briefly introduce the SMT Quick Run package, a package of scripts and instructions for building a statistical machine translation system on your own parallel corpora. It also includes a sample of building a Czech-English translation system on parallel text from the PCEDT corpus.

For the final evaluation we deployed the Pharaoh [Koehn et al., 2003] system for phrase model training and decoding with Giza++ used for the translation model training (IBM model 3 is sufficient in this case).

Both systems ISI Decoder and Pharaoh use the SriLm language modeling toolkit [Stolcke, 2002] for the language model training, the trained model can also be accessed through the SriLm interface. The English language model is trained on the training part of the PCEDT, two additional years (1995 and 1996) of the Wall Street Journal archive, the English part of the Reader's Digest corpus and a part of the English North American News Text Collection, all together more than 2,300,000 sentences (cca 100,000,000 tokens).

Evaluation of the linguistic preprocessing on the overall performance of the SMT system is discussed in Section 6.2.

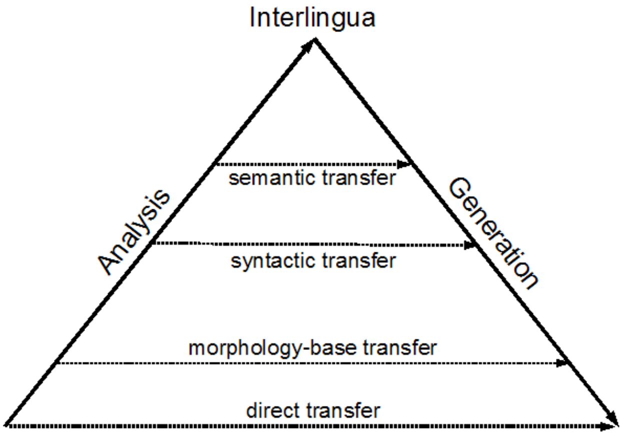

Figure 5.4: Classical Machine Translation Schema - Vauquois' Triangle

5.2 Dependency-based Machine Translation

In this chapter we briefly describe the Dependency-based Machine Translation (**DBMT**) system proposed and implemented at IFAL by the author, Martin Čmejrek, Jiří Havelka, and Jan Hajič. We focus on the transfer module which uses the dictionaries introduced in Chapter 4. Further details and discussions can be found in [Cuřín et al., 2002, Čmejrek et al., 2003a]. This system was also used as a base scheme by the "Generation in the context of MT" group at the NLP Workshop at CLSP, Johns Hopkins University at 2002 [Hajič et al., 2002].

The DBMT exploits the idea of the classical analysis–transfer–generation architecture usually presented in the so-called Vauquois' triangle [Vauquois, 1975] as shown in Figure 5.4. The analysis goes from the input language sentence through morphological, syntactic, and semantic annotation to *interlingua*, e.g. common representation of the meaning across all languages. Then the generation process follows the same stages when creating the output sentence. It is commonly agreed that the closer the transfer procedure is to interlingua, the more straightforward it is. But in special cases, such as for close language pairs, the deep analysis of the input is not necessary to obtain reasonable results.

For DBMT, we decided to use the level of tectogrammatical analysis for the transfer. Figures 5.5 and 5.6 show analytical and tectogrammatical analysis of the sentence *"According to his opinion UAL's executives were misinformed about the financing of the original transaction"* and its Czech translation. While, the analytical trees for Czech and English are quite different, the tectogrammatical representation is, with one exception when the

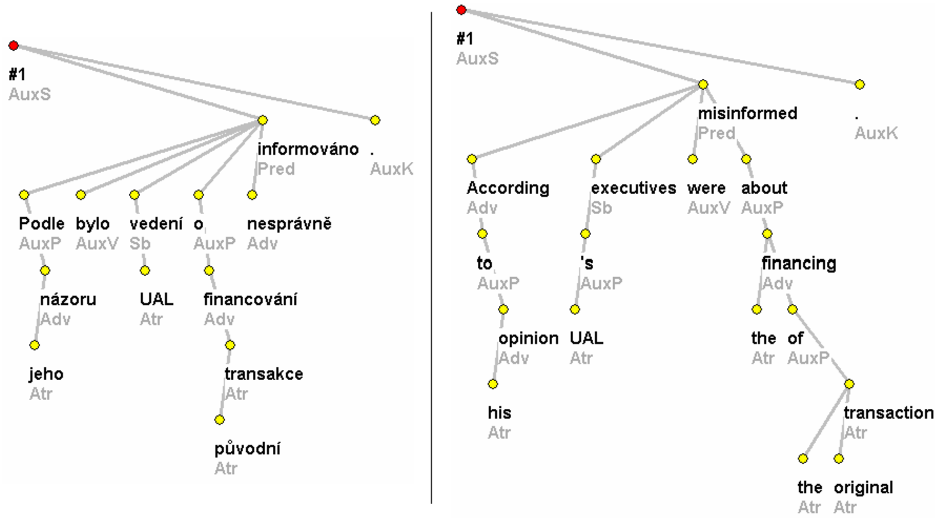

Figure 5.5: Comparison of the analytical tree of the English sentence *"According to his opinion UAL's executives were misinformed about the financing of the original transaction"* and its Czech Translation.

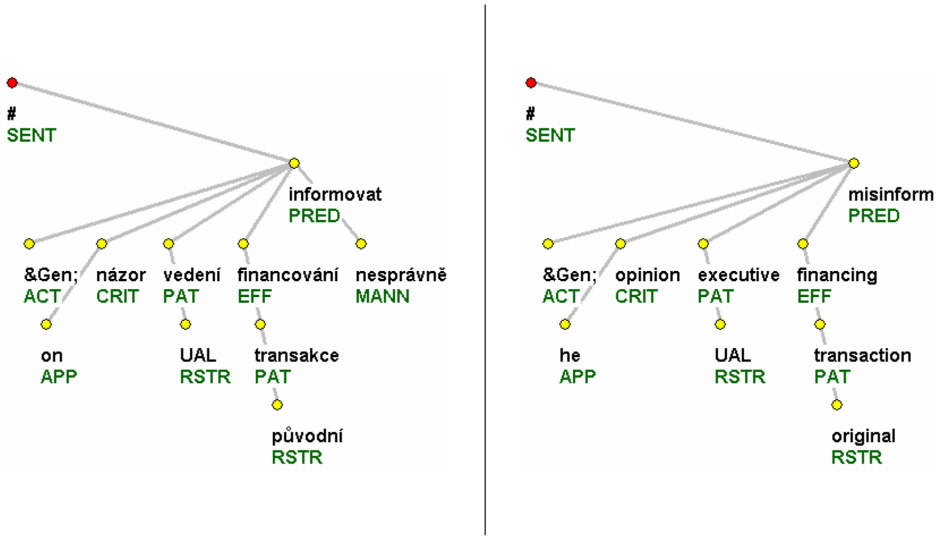

Figure 5.6: Comparison of the tectogrammatical tree for the same sentence pair.

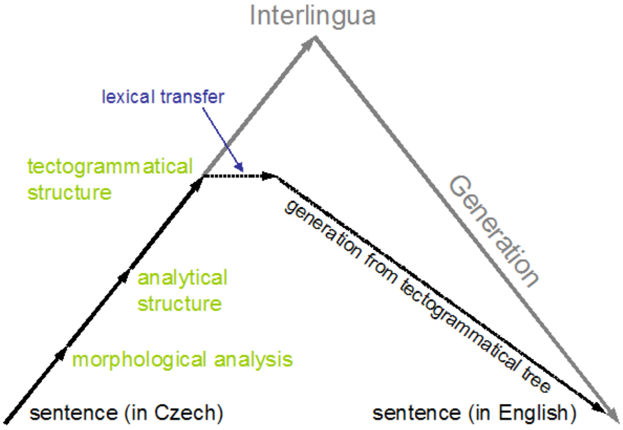

Figure 5.7: MT Schema for DBMT System

English "*misinform*" is translated as two Czech words "*nesprávně informovat*", almost identical. This leads us to the idea that a relatively simple, lexical-base replacement might produce acceptable results for the transfer procedure. Note that this approach is a simplification because in fact the labels of the tectogrammatical nodes (= *tectogrammatical lemmas*) represent the meaning encoded in a particular tectogrammatical node. In our case the transfer lexicon was derived from a lemmatized parallel corpora with small adjustments to fit the purpose, such as considering the Czech verb and the preceding or following reflexive pronoun "*se*" as one lemma, or concatenating English phrasal verbs for the similar reason.

Figure 5.7 gives the overview of DBMT schema.

5.2.1 The DBMT Processing Pipeline

Analysis

In the analysis step, we carry out morphological analysis, parsing into an analytical representation, and a conversion into a tectogrammatical representation. There is still a certain gap between what we obtain by an automatic procedure and between what the theory says about the tectogrammatical representation. For example, we do not handle coreferences, topic-focus articulation and some other phenomena.

The Czech sentence is automatically tokenized, morphologically tagged, and each word form is assigned a *lemma*, i.e. a basic form, using Hajič and Hladká [Hajič and Hladká, 1998] tagging tools.

The analytical representation of Czech is obtained in two steps: the input is parsed by a statistical dependency parser – either Collins' parser [Hajič et al., 1998], or Charniak's parser [Charniak, 1999], and then a C4.5 module for automatic assignment of *analytical functions*.

The analytical structure is converted into the tectogrammatical structure. During the transformation only auto-semantic nodes are preserved. The functional nodes, such as prepositions and subordinating conjunctions, are removed from the tree. For verbs, the procedure searches for their arguments and if some inner participant (for example `ACTor` or `PATient`) or some obligatory argument (according to a valency dictionary) is not present in the analytical structure, then this position is filled by some default value. Furthermore, all the nodes of a complex verb form are collapsed into one node, and attributes specifying sentence and verb modality, aspect are set. These transformations are described by linguistic rules [Böhmová, 2001]. Then, *tectogrammatical functors*, which were not assigned by the transformation procedure, are subsequently added by a C4.5 classifier [Žabokrtský et al., 2002].

The analysis part corresponds to the steps applied to the Czech part of the Prague Czech-English Dependency Treebank (PCEDT). The reported results are for a fully automatic processing of the Czech input using the Charniak's parser for building the analytical structure.

Transfer

The transfer procedure consists of a lexical replacement of the tectogrammatical base-form attribute (*trlemma*) of autosemantic nodes by its English equivalents found in the Czech-English transfer dictionary. The current implementation of the system is capable of handling 1–1 and 1–2 translations and, as already been mentioned, there are also special rules for reflexive and phrasal verbs.

While translating, the system uses either only one – the most probable – translation from the transfer dictionary or for multiple translation possibilities a special output structure (a pack-tree format) capable of splitting the tectogrammatical nodes is created, see Section 5.2.2.

See Section 6.3 for further details about the transfer dictionaries used and for discussion on their influence on performance.

Generation

When generating from the tectogrammatical representation, two kinds of operations have to be performed: lexical insertions and transformations modifying word order.

In each of the following five steps, the whole tectogrammatical tree is traversed and rules pertaining to the particular group are applied:

- **Determining contextual boundness**: For the reordering of constituents in the English counterpart of the Czech sentence, and for determining the definiteness of noun phrases in English, we make use of the fact that Czech is a language with a relatively high degree of word order freedom and it mainly uses the left to right ordering to express the information structure.

  Verb complements and adjuncts have to be rearranged in order to conform with the constraints of English grammar, according to the sentence modality.

- **Generation of verb forms**: The active or passive voice, tense, mood, person are determined according to or taken over from the semantic representation of the sentence.

  Auxiliary verbs needed to create a complex verb form are inserted.

- **Insertion of prepositions and articles**: The correspondence between tectogrammatical functors and auxiliary words is a complex task. The rules describing surface realization by a preposition take into consideration the tectogrammatical functor, the original Czech preposition, and the English lexical word.

  The task of generating articles in English is not easy due to the absence of articles in Czech. We use information about the contextual boundness of a noun phrase to make the decision. The definite article is inserted when the noun phrase is either contextually bound, postmodified, or is premodified by a superlative adjective or ordinal numeral. Otherwise, the indefinite article is used. An article is not inserted for uncountable or proper nouns.

  A packed-tree format is used to represent multiple variants of insertions of both prepositions and articles.

- **Morphology**: A simple module generates the surface word form from the lemma and its morphological tag. It searches through the table of triples [word form, morphological tag, lemma] for the word form corresponding to the given lemma and morphological tag. Also, the appropriate form of the indefinite article is selected according to the word immediately following.

5.2.2 Language Model Rescoring

Being aware of the fact that choosing just one English translation representative for a particular Czech tectogrammatical lemma is an oversimplification, we have introduced a special structure capable of storing multiple translation possibilities in the tectogrammatical node. This structure is based on a pack-tree representation similar to the one introduced by [Langkilde, 2000].

Packed-tree representation is used for storing multiple translations found in the transfer dictionary and for possible variants of prepositions and articles inserted during the generation process. Because the number of potential sentence hypotheses stored in a pack-tree grows exponentially with the sentence length, we have limited the number of translation variants by cutting off translations with a probability lower than a certain threshold.

The process of Czech input sentence processing is the same as described in the previous section, except that during the transfer and generation multiple variants may be added into the pack-tree structure.

After the generation phase is finished, the packed-tree representation is unwrapped into a list of translation hypotheses. This list of sentences is scored by an n-gram language model for English and the sentence with the highest score is selected as the result of the translation process.

The trigram language model with Good-Turing discounting and Katz back-off for smoothing was trained by the SriLm language modeling toolkit [Stolcke, 2002] on a 52 million-word monolingual corpus of the Wall Street Journal from 1995 and 1996[6] [Linguistic Data Consortium, 1995].

The performance of the DBMT system with regard to the deployed transfer dictionary and to the method of finding the translation equivalents is discussed in Section 6.3.

[6] Wall Street Journal articles included in the Penn Treebank and consequently in the PCEDT are from other years.

Chapter 6

Evaluation and Results

6.1 Evaluation Metric

By virtue of the natural language translation, the evaluation of translation quality is a very difficult task. Human evaluations usually weigh not only the adequacy of translation, but also many other aspects such as fluency, fidelity, syntactic correctness and stylistic purity. We can expect that such evaluation to be expensive and slow. This was a big problem for machine translation system developers who needed to have effective, cheap, and fast measures to evaluate their daily progress. The automatic MT evaluation metrics which became widely recognized and used was a Bleu score proposed by MT group from IBM Research in [Papineni et al., 2001].

6.1.1 BLEU Score Metric

The name of the metrics - Bleu - is an abbreviation for BiLingual Evaluation Understudy.

This automatic evaluation is based on the comparison of the output from a tested system with at least one reference translation. The basic idea is to measure the number of matching n-grams between the output and the reference or references. A proposal for using multiple reference translations tries to reflect the nature of translation, i.e. that for a particular sentence in the source language there are potentially many perfect or reasonable translations in the target language. In literature the usual number of available reference translations is four. The Bleu score is commonly counted on the whole tested set of documents, but of course it is also possible to obtain scores only for a document or sentence level.

The authors show a distinctive correlation between the automatically counted scores and the human evaluations, see [Papineni et al., 2001] for details. They compare the automatic metric with both monolingual and bilingual human judgments on the machine translation output.

Formally, the Bleu score is computed as follows:

Let p_n is a precision[1] of n-grams, N is a maximal considered n-gram (N is usually set to 4), w_n is a positive weight of n-gram score for a particular n[2], r and c are lengths in words of reference and candidate (evaluated) translation, respectively. BP is called *brevity penalty* factor. Then,

$$\begin{aligned}\textsc{Bleu} &= \mathrm{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right) \\ \mathrm{BP} &= \min(e^{1-r/c}, 1)\end{aligned}$$

The ranking is more apparent in the log-scale:

$$\log \textsc{Bleu} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log(p_n)$$

The *brevity penalty* penalizes candidates shorter than their reference translations. This is to avoid cheating by including only the highly ranking n-grams in the translation. Candidate translations longer than their references are implicitly penalized by the modified n-gram precision measure, it is not needed to penalize them again.

For the evaluation of the machine translation systems in this work, we used the latest script (version 11b, released on May 20th, 2004) available on the NIST official website (`http://www.nist.gov/speech/tests/mt/`).

6.1.2 Evaluation of Our Results

The test translations and the evaluations are performed on the development and evaluation sets from the PCEDT. We use five reference translations: four English retranslations and the original text from the Penn Treebank corpus. Figure 3.10 shows the sample sentence with reference translations from the PCEDT corpus.

As a limit of the [Papineni et al., 2001] for our systems, we can see an average score of each of the English retranslations against the rest of the references. The average score of human translation professionals is 0.5560.

[1] Precision is expressed as a fraction of units (n-grams) correctly identified by the algorithm over the total number of units.

[2] By default, the n-gram weight is uniformly set to $1/N$.

	DevTest	*EvalTest*	
PHARAOH *on raw text from PCEDT*	0.3592	0.3291	**(A)**
PHARAOH *on lemmatized text from PCEDT*	0.3747	0.3458	**(B)**
PHARAOH *on PCEDT with Czech' preprocessing*	0.3858	0.3650	**(C)**

Table 6.1: BLEU scores of translations from the PHARAOH system

6.2 Evaluation of the SMT System with Linguistic Preprocessing

In this section, we compare the quality of automatic translations from Czech to English depending on the preprocessing method of the Czech input.

We use the PHARAOAH SMT system as described in Section 5.1.4 with the following stages of linguistic preprocessing:

(A) No preprocessing, i.e. using the raw text for both sides, Czech and English.

(B) Lemmatization of the Czech part of the corpus and of the input to the translation system.

(C) Applying the *Czech'* preprocessing module on the Czech part of the corpus and on the input to the translation system.

A system trained on the corpus without any preprocessing (A) is a baseline. Note, that it is not an easy task to beat this baseline because it uses the state-of-art modules for decoding and for language and translation models training.

The next step is the lemmatization of the Czech part of the corpus (B); this preprocessing reduces the disproportion in the number of unique word forms between Czech and English, refer to details in Section 5.1.1.

From the final results listed in Table 6.1 we can see the positive influence of more advanced model of the linguistic preprocessing – the *Czech'* module (C). The SMT system trained on the Czech' corpus gained 3.5% of BLEU on the evaluation test against the baseline system. We can also see the improvement against the lemmatized version of preprocessing. On the developement test set the improvement is less significant, this might be because the parameters for both the translation model and language model training were fine tuned on the baseline (raw-text) system. See sample translations from this system in Appendix C for comparison.

System	1(+)	2	3	4	5(−)	**average**
Raw text – EGYPT *(baseline)*	3.5	6.5	13.5	27.5	15	**3.66667**
Simple lemmatized – EGYPT	3	12.5	23.5	23	4	**3.18939**
Lemmatized' – EGYPT	6	16	16.5	18	9.5	**3.13636**
Commercial 2	7	18.5	21	16.5	3	**2.84848**
Lemmatized'+dict – ALTEMP	13	18	14	17	3.5	**2.69466**
Commercial 1	11	19.5	18	14.5	3	**2.68182**

Table 6.2: Human evaluation of Czech/English Translation

6.2.1 Human Evaluations from WS'99

For variety we include the evaluation section from the NLP workshop in 1999 [Al-Onaizan et al., 1999], i.e. from the time when the BLEU metrics and the PCEDT corpus were non-existent.

The reported results are for the Reader's Digest corpus (mentioned in Section 3.1.2). The system for *Lemmatized'* preprocessing is an antecedent of our *Czech'* module. The results are also compared with two Czech commercial translation systems: PC Translator 98 and SKIK v. 4.0.

We carried out a human evaluation of translations to observe the progress obtained by each level of preprocessing the Czech input. The tool for the human evaluation displays the original sentence (in Czech) and translations from different translation systems. Translations are shuffled for each original sentence. Judges assign marks from 1 to 5 to each translation. Mark 1 is the best, mark 5 is the worst translation.

In our particular case the evaluation was done by two evaluators on 66 randomly chosen sentences from the test data. Results are in the Table 6.2. The average counts of the assigned marks are in the columns. Rows correspond to translation systems. The average value of marks assigned to each translation system is in the last column in the table.

We can observe the progress of quality of translation obtained by the EGYPT toolkit from the baseline to the simple lemmatized version and to the extended lemmatized version of Czech input (*Lemmatized'*) in comparison with the two commercial systems and the Alignment Templates system (ALTEMP). Results on Czech/English translation using the Alignment Templates system are better than one of the commercial systems and almost the same as the second one.

6.3 Evaluation of the Dependency-based MT System

In this section, we discuss the influence of translation dictionary utilized for the transfer on the overall performance of the system. Dictionaries described in Chapter 4 that are used:

- Filtered and Weighted Dictionary for Transfer (Section 4.1)
- Probabilistic Dictionary from PCEDT (Section 4.2.1)

Results of the following translation system configurations are being compared:

(a) DBMT system using the most frequent English translation from PCEDT Probabilistic Dictionary in the lexical transfer.

(b) DBMT system using the most probable English translation with respect to the PCEDT parallel corpus from PCEDT Probabilistic Dictionary.

(c) DBMT system using the most probable English translation from the full transfer dictionary specially filtered and weighted for the purpose of DBMT transfer[3].

(d) DBMT system using the full transfer dictionary specially filtered and weighted on PCEDT. The list of translation hypotheses is generated from the pack-tree representation. The best output sentence is selected by the English language model.

As a baseline we used the transfer dictionary without parallel corpus weighting and with only one translation of a particular Czech tectogrammatical lemma. For a baseline configuration (a) we choose the translation with the highest frequency in the English monolingual corpus. Monolingual frequencies are included in the Probabilistic Dictionary from PCEDT. We use this source directly because it makes the influence of deploying weighted dictionary (b) comparable.

Other two configurations (c,d) make use of a better translation dictionary, system (d) even more incorporates language model rescoring.

For example word "*obchodování*" in the sentence "*DIG Acquistition Corp., akviziční prostředník newjerseyského investora, oznámil, že na konci včerejšího* <u>*obchodování*</u> *bylo odkoupeno 560 839 akcií.*" (reference translation: "*DIG Acquistition Corp., an acquisition broker of the New Jersey investor, announced*

[3] Note that the size (in number of entry/translation pairs) of this transfer dictionary is more than three times bigger than the size of the Probabilistic Dictionary.

	DevTest	*EvalTest*	
DBMT, PCEDT ProbDict, most frequent En tr.	0.0968	0.0798	**(a)**
DBMT, PCEDT ProbDict, weighted on PCEDT	0.1136	0.0995	**(b)**
DBMT, transfer dictionary, weighted on PCEDT	0.1916	0.1705	**(c)**
DBMT, transfer dictionary, LM rescoring	0.1921	0.1705	**(d)**

Table 6.3: BLEU score for different dictionaries used for transfer in DBMT.

that 560,839 shares were purchased at the end of yesterday's trading.") was translated differently for each dictionary:

```
Entry: obchodování
 -> most frequent transl. in the PCEDT dictionary (a): traffic
 -> most probable transl. in the PCEDT dictionary (b): dealing
 -> most probable transl. in the transfer dictionary (c,d): trading

Entry "obchodování" in the PCEDT dictionary:
  f              e           P(e|f)      c(e)
  obchodování#N  dealing#N   0.24904      1043
  obchodování#N  traffic#N   0.0311563   17462

Entry "obchodování" in the transfer dictionary:
  f              e           P(e|f)
  obchodování#N  trading#N   0.827002
  obchodování#N  trade#V     0.0290892
  obchodování#N  session#N   0.0277124
  obchodování#N  business#N  0.0139494
```

Table 6.3 compares BLEU scores for different DBMT configurations. We can see the positive influence of the dictionary weighting procedure on the system performance, even though the resulting BLEU scores for configurations **(a)** and **(b)** are too low. This is because the coverage of the PCEDT probabilistic dictionary is not sufficient.

More promising results (twice the baseline score) are achieved with a fully fledged transfer dictionary. The incorporation of multiple hypothesis format and subsequent language model rescoring did not have the impact on the results as we would expect. We assume that the variance would be bigger for more generic domains than a domain of financial news dominating the Penn Treebank corpus.

Still, the DBMT system is out-performed by the state-of-art statistical machine translation systems described Section 6.2.

Chapter 7

Conclusions

Let us briefly summarize the results achieved in this work:

- We have been in touch with the statistical machine translation community since the important NLP workshop in 1999 where the first publicly available tool for building statistical translation models – Giza – was released. Czech was among the first languages exercised by this system.

- We have proposed a module for linguistic preprocessing of Czech with the intention to help the performance of a statistical machine translation systems translating from Czech. We continued improving the system to be able to add its contribution to the state-of-art machine translation systems, such as phrase-based machine translation.

- We have explored alternative techniques of linguistic preprocessing together with analysis of the influence of particular features of the Czech language.

- We have proposed and implemented an algorithm for the automatic extraction of translation dictionaries from parallel texts, with subsequent filtering and weighting for two main purposes. First, to provide human translators with readable dictionary which can help them to be consistent in terminology with their colleagues. Second, to build a weighted dictionary for subsequent use in rule-based translation systems.

- We have suggested and deployed methods for merging multiple manual dictionaries with the dictionary automatically built from the parallel corpus and have prepared such a dictionary for use in the machine translation system.

- We have participated in designing and implementing a translation tool based on tectogrammatical dependency structure – the Dependency-base machine translation system. The main contribution was in the transfer procedure. We have shown the importance of the quality of the translation dictionary intended for the transfer.

- We have created first syntactically annotated parallel corpus based on the dependency representation – the Prague Czech-English Dependency Treebank. This corpus was publicly released as a data collection by the renowned institution – the Linguistic Data Consortium. Apart from the machine translation community the corpus has also been of interest to researches in other linguistic disciplines such as syntactic parsing and studies on semantic representation. The success of this activity can be seen in the six publications accepted at various international conferences and workshops.

Appendix A

SMT System

A.1 IBM Translation Models Training

The translation model consists of the following probability tables:

- **t-table** ... word-translation table [two-dimensional, (# of English words) × (# of foreign words): $t(f_j|e_i)$], used in all models
- **a-table** ... alignment table [four-dimensional: $a(i|j,l,m)$, (maximum sentence length)^4], used in Model 2
- **d-table** ... distortion table [four-dimensional: $d(j|i,l,m)$, (maximum sentence length)^4], used in Model 3
- **n-table** ... fertility table [two-dimensional, (# of English words) × (maximum fertility allowed)], used in Model 3
- p_0, p_1 ... NULL-word insertion parameters, used in Model 3

where

- i, j are indexes of English/foreign words
- l, m are lengths of English/foreign sentences
- e_i is an English word on the i-th position in the sentence
- f_j is a foreign word on the j-th position in the sentence

A.1.1 Model 1 and Model 2 training

Model 1 and Model 2 have an efficient EM training algorithm that avoids the necessity of enumerating all alignments:

```
set t, d, n, and p tables uniformly
for several iterations
  set up count tables tc, dc, nc, and pc with zero entries
```

```
for each sentence pair (e, f) of lengths (l, m)
  for j = 1 to m
    total = 0
    for i = 0 to l
      total += t(fj | ei) * a(i | j,l,m)
    for i = 0 to l
      tc(fj | ei) +=  t(fj | ei) * a(i | j,l,m) / total
      ac(i | j,l,m) +=  t(fj | ei) * a(i | j,l,m) / total
normalize tc and ac to create new tables t and a
```

It is possible to run Model 2 directly on the corpus, but it can be useful to solidify certain word-pair connections by first running a simpler model (Model 1) that only has a single table t. The training algorithm for Model 1 looks similar to the one above. We take the t-table learned by Model 1, together with a uniform a-table, as the starting point for Model 2.

A.1.2 Model 3 training with Viterbi alignment

Here is the basic scheme of Model 3 training, individual steps are described below.

```
(1) set t, d, n, and p tables uniformly or according to model2 tables
(2) for several iterations
(3)   set up count tables tc, dc, nc, and pc with zero entries
(4)   for each sentence pair (e, f) of lengths (l, m)
(5)     find the best Viterbi alignment 'v2Ali' of model2
(6)     do hill climbing for 'v2Ali'; find the best alignment
          'bestAli' derived form 'v2ali' by moves and swaps
(7)     find neighbour alignments of 'bestAli'
(8)     for each neighbour alignment of 'bestAli' of (e, f)
(9)       collect and update counts for tc, dc, ac, nc and pc
(10)  normalize tc, dc, ac, nc, and pc
        to create new tables t, d, a, n, and p
```

Ad step (1)

We use t-table and a-table from Model 2. We get n-table values by the Model 2 to Model 3 transferring procedure. d-table should be set uniformly, p_0 and p_1 are set to the fixed value ($p_0 = 0.95$).

Ad step (2)

Allow the setting of the number of iterations in configuration file. By default use 5 iterations of Model 1, 5 iterations of Model 2, and 5 iterations of

Model3.

Ad step (5); Find the best Model 2 Viterbi alignment

This finds the best Model2 alignment (i.e. no fertilities stuff) in A for the given sentence pair. Its score is returned in A. Its fertility info in $Fert$.

Parameters:

- es ... vector of English words in the sentence
- fs ... vector of French words in the sentence
- Representation of alignment
  - A ... alignment, vector of aligning indexes ($A[j] = i$)
  - $Fert$ ... fertilities of English word ($Fert[i] = n$)
- *best_score* ... the best score found, i.e. $P(a, f|e)$
- t-table, a-table.

Variables:

- i, j ... indexes of English/French words
- l, m ... lengths of English/French sentences
- *best_i* ... the best alignment for given English word
- *temp*, *score*, *ss* ... probabilities

Procedure:

```
best_i = 0 ;
for i = 0 to l
  Fert[i] = 0 ;
for j = 1 to m
  score = 0 ;
  for i = 0 to l
    if ((Fert[i]+1 < MAX_FERTILITY) &&
     ((i == 0 &&  (m >= 2*(Fert[0]+1))) || (i != 0)))
      temp = t(es[i], fs[j]) * a(i, j, l, m) ;
      if (temp > score )
        best_i = i ;
        score = temp ;
  if (score != 0)
    Fert[best_i]++ ;
    A[j] = best_i ;
best_score = count_prob(A, Fert, tTable, fs, es);
```

Where '`count_prob`' proceeds as described bellow and corresponds to $P(a, f|e)$ in terms of [Al-Onaizan et al., 1999], page 7, and to the equation (32) in [Brown et al., 1993].

Ad step (6); hill climbing

Hill climbing is in fact searching for local maxima of $P(a|f, e)$. Let us describe more terminology ([Brown et al., 1993], page 278).

We say that two alignments, a and a', *differ by a move* if there is exactly one value of j for which $a_j \neq a_j{}'$. We say that two alignments *differ by a swap* if $a_j = a_j{}'$ except at two values, j_1, j_2 for which $a_{j_1} = a_{j_2}{}'$. We say that two alignments are *neighbours* if they are identical or differ by a move or by a swap. We denote the set of all neighbours of a by $\mathcal{N}(a)$.

Let $b(a)$ be that neighbor of a for which the likelihood $P(b(a)|f, e)$ is greatest. The sequence of alignments $a, b(a), b^2(a) = b(b(a)), \ldots,$ converges in a finite number of steps to an alignment that we write as $b^\infty(a)$.

We start hill climbing for the best Model 2 Viterbi alignment $V(f|e; 2)$ (the alignment 'v2Ali' found in step (5)) and we find the resulting alignment $b^\infty(V(f|e; 2))$ as an approximation of $V(f|e; 3)$, that is the best Model 3 Viterbi alignment (the alignment 'bestAli' in step (6)). Hill climbing stops if there is no better score of alignment in neighbour alignments.

$P(b(a)|f, e)$ can be counted directly from $P(a|f, e)$ by a simple algorithm described on pages 7 and 8 in [Al-Onaizan et al., 1999].

These tricks make hill climbing fast, and they also allow us to determine alignment weights for collecting counts in the neighbourhood of the pseudo-Viterbi alignment.

Ad step (7); find neighbour alignments of 'bestAli'

Create the set of neighbour alignments of 'bestAli', i.e. $\mathcal{N}(V(f|e; 3)$ by swaps and moves.

Ad steps (8), (9)

Compute $P(a|e, f)$ for all neighbour alignments of 'bestAli' and update fractional count tables (t,d,a,n,p). Denominator

$$'align\_total\_count' = \sum_{a' \in \mathcal{N}(a)} P(a', f|e)$$

Ad step (10)

Normalize tc, dc, ac, nc, and pc to create new tables t, d, a, n, and p.

A.2 SMT Quick Run Package

Statistical Machine Translation Quick Run Package (version 1.2)

This is a brief description of how to build a statistical machine translation system on your parallel corpora as quickly as possible. It includes a sample of building a Czech-English translation system on parallel text from the Prague Czech-English Dependency Treebank (PCEDT) corpus. PCEDT version 1.0 is distributed by Linguistic Data Consortium (LDC), see LDC Catalog.

SMT Quick Run Package is a set of scripts for preparing training data in the appropriate format, for training language and translation models, and for running a decoder in server mode. The SMT Quick Run requires the following software packages:

- CMU Statistical Language Modelling Toolkit (version 2) for language model training
- mkcls for training of word classes
- GIZA++ for training of statistical translation models
- ISI ReWrite Decoder (version 1.0.0a) for decoding (translation)
- NIST MT evaluation kit for BLEU and NIST evaluation

SMT Quick Run Package is available for download at:
`http://ufal.mff.cuni.cz/~curin/SMT_QuickRun/`

Building a Czech-English Statistical Machine Translation System on the PCEDT Corpus

In this sample, the translation goes from Czech to English. It requires installation of the SMT Quick Run package and all executables listed in the Download of Necessary Executables section. Because the statistical machine translation employs the Noisy Channel Model, we use the term source language for English and the term target language for Czech in this case.

In the example below we suppose that the SMT QuickRun package is in your home directory, i.e. `/home/my_home/SMT_QuickRun1.2/`, and PCEDT distribution is in the directory `/data/LDC/PCEDT_CD_1.0`

In the directory `/home/my_home/SMT_QuickRun1.2/PCEDT_Sample/` edit the `Makefile`, set variables `PCEDT_ROOT` and `SMT_QR_ROOT` as indicated:

```
#######################################
# Variables to be set

## Root of PCEDT distribution
#  Such as /data/LDC/PCEDT_CD_1.0
PCEDT_ROOT=/data/LDC/PCEDT_CD_1.0

## Root of SMT Quick Run package
#  Such as /home/my_home/tools/SMT_QuickRun1.2
SMT_QR_ROOT=/home/my_home/SMT_QuickRun1.2
```

For training data preparation run `make` as follows:

```
cd /home/my_home/SMT_QuickRun1.2/PCEDT_Sample/
make prepare_training_data
```

It copies and transforms all training data from PCEDT distribution, continue by running

```
make train_models
```

Language model training takes about 1 minute on Intel(R) Pentium(R) 4 CPU 2.66GHz and `LM_training.log` is its log file. Word classes training and translation models training takes about 25 minutes on the same machine. If the translation models training takes suspiciously less than indicated, check `TM_training.log` for errors. Then a configuration file and a running script for the decoder are created, see log in `prepare_decoder.log` for details.

At the end of the models training process a running script for the decoder is created in the file `Cz2En_start_decoder.sh`. This script runs ISI ReWrite Decoder in server mode and creates script for translations. Type

```
./Cz2En_start_decoder.sh
```

If successful, the decoder enters server mode and indicates a port.

```
entering server mode
Listening on port 1083.
```

Now the system is ready for translations.

The newly created script `PCEDT_Sample/Cz2En_translate.sh` expects a file to be translated on standard input and a resulting translation is written to the standard output.

Use script `Cz2En_kill_decoder.sh` to kill the decoder server after performing all translations.

Hint: The English language model is built on a relatively small corpus of the English side of the PCEDT parallel texts. You can improve the translation quality by adding more English monolingual data for the LM training. See details of data format and location in the next section.

For evaluation of the results against 4 reference translations do:

First prepare test data from the PCEDT distribution by running:

```
make prepare_test_data
```

Perform translations and evaluate results using BLEU and NIST metrics by running:

```
make evaluate_results
```

Enjoy translations!

A.3 *Czech'* Preprocessing Grammar

A.3.1 Nouns

```
Proper Nouns
 if (NOUN is ProperNoun) { # Paris, Jonathan
  lemma
 }
else
 {

1st case (nominative)
 if (NOUN is governing PRONOUN or CARDINAL NUMERAL,
      or NOUN is uncountable) # my house, three beers, coal
  {
   lemma-S  ... sigular
   lemma-P  ... plural
  }
 else # ex. the house, a small house
  {
   &#NtheS; <dependent ADJECTIVE | dependent ORDINAL NUMERAL> lemma-S
              ... sigular
   &#NtheP; <dependent ADJECTIVE | dependent ORDINAL NUMERAL> lemma-P
              ... plural
  }

2nd case (genitive)
 if (NOUN is governing PRONOUN or CARDINAL NUMERAL,
       or NOUN is uncountable)
  {
   if (NOUN is governed by PREPOSITION in the same case)
    {
     lemma-S  ... sigular
     lemma-P  ... plural
    }
   else # ex. of the small house
    {
     &#Nprep2; <dependent ADJECTIVE | dependent ORD. NUM.> lemma-S
              ... sigular
     &#Nprep2; <dependent ADJECTIVE | dependent ORD. NUM.> lemma-P
              ... plural
    }
  }
 else # NOUN is not governing PRONOUN
```

```
  {
   if (NOUN is governed by PREPOSITION in the same case)
    {
     &#NtheS; <dependent ADJECTIVE | dependent ORD. NUM.> lemma-S
              ... sigular
     &#NtheP; <dependent ADJECTIVE | dependent ORD. NUM.> lemma-P
              ... plural
    }
   else # NOUN is not governed by PREPOSITION in the same case
    {
     &#Nprep2; &#NtheS; <dependent ADJ. | dep. ORD. NUM.> lemma-S
              ... sigular
     &#Nprep2; &#NtheP; <dependent ADJ. | dep. ORD. NUM.> lemma-P
              ... plural
    }
  }

3rd case (dative)
 the same as the 2nd case

4th case (accusative)
 the same as the 1st case

5th case (vocative)
 the same as the 1st case

6th case (locative)
 the same as the 2nd case

7th case (instrumental)
 the same as the 2nd case

 }
```

A.3.2 Adjectives

```
1st grade
  lemma

2nd grade [more ..]
  &#A2; lemma

3rd grade [most ..]
```

```
  &Athe; &#A3; lemma

&#Anot; ... negative adjective (insert before lemma)
```

A.3.3 Pronouns

```
Personal Pronouns [PP]
 if((PERSON == 3)&&(NUMBER == sigular)) # he, she, it
  {
   1st, 5th cases # he
     lemma-1S-M  ... singular masculinum
     lemma-1S-F  ... singluar femininum
     lemma-1S  ..... singular others

   2nd case # his
     lemma-2S-M  ... singular masculinum
     lemma-2S-F  ... singluar femininum
     lemma-2S  ..... singular others


   4th case # him
     lemma-S-M  ... singular masculinum
     lemma-S-F  ... singluar femininum
     lemma-S  ..... singular others

   3rd, 6th, 7th cases (*C is the relevant case) # him
    if (PRONOUN is governed by PREPOSITION in the same case)
     {
      lemma-S-M  ... singular masculinum
      lemma-S-F  ... singluar femininum
      lemma-S  ..... singular others
     }
    else # PRONOUN is not governed by PREPOSITION in the same case
     {
      &#Pprep*C; lemma-S-M  ... singular masculinum
      &#Pprep*C; lemma-S-F  ... singluar femininum
      &#Pprep*C; lemma-S  ..... singular others
     }
  }
 else # I, you, we, they
  {
   1st, 5th cases # we
     lemma-1S  ..... singular
     lemma-1P  ..... plural

   2nd case # our
     lemma-2S  ..... singular
     lemma-2P  ..... plural

   4th, case # us
     lemma-S  ..... singular
     lemma-P  ..... plural

   3rd, 6th, 7th cases (*C is the relevant case) # us
    if (PRONOUN is governed by PREPOSITION in the same case)
     {
      lemma-S  ..... singular
      lemma-P  ..... plural
     }
    else # PRONOUN is not governed by PREPOSITION in the same case
     {
      &#Pprep*C; lemma-S  ..... singular
      &#Pprep*C; lemma-P  ..... plural
     }
  }

 exmpl: já, mu, nich # I, to him, about them


Relative Pronouns [P4,PE]
  lemma

 exmpl: což


Relative Pronouns II [P9] (after preposition)
  lemma-S-M  ... singular masculinum
  lemma-S-F  ... singluar femininum
  lemma-S  ..... singular others
  lemma-P  ..... plural

 exmpl: o němž, s níž # about him, with her


Relative Pronouns III [PJ] (no preposition before)
 1st, 4th, 5th cases
```

```
   lemma

 2nd, 3rd, 6th, 7th cases (*C is the relevant case)
    &#Pprep*C; lemma

 exmpl: jehož # whose


Relative Possesive Pronouns [P1]
  lemma-S  ... singular
  lemma-P  ... plural

 exmpl: jehož, jejíž


Possesive Pronouns [P8, PS]
  lemma-S-M  ... singular masculinum
  lemma-S-F  ... singluar femininum
  lemma-S  ..... singular others
  lemma-P  ..... plural

 exmpl: svou, svému ..  # his, her, its, their


Reflexive Pronouns [P7]
  lemma-PREFL

 exmpl: se, si


Reflexive Pronouns II [P6]
  lemma-PREFL2

 exmpl: sebe # self


Pointing Pronouns [PD]
  lemma-S  ... singular
  lemma-P  ... plural

 exmpl: tuto, takové # this, these
```

```
Interrogative Pronouns [PQ, PY]
  lemma

 exmpl: co # what


Interrogative Pronouns II [PK]
 1st, 5th cases # who
   lemma-1

 2nd case # whose
   lemma-2

 4th, case # whom
   lemma

 2nd, 3rd, 6th, 7th cases (*C is the relevant case) # to whom
  if (PRONOUN is governed by PREPOSITION in the same case)
   {
    lemma
   }
  else # PRONOUN is not governed by PREPOSITION in the same case
   {
    &#Pprep*C; lemma
   }

 exmpl: kdo, komu # who, to whom


Infinitive Pronouns [PL]
  lemma

 exmpl: sám, všechen, nikdo # alone, all, nobody


Negative & Undetermined Pronouns [PW, PZ]
 1st, 4th, 5th cases
   lemma-S
   lemma-P

 2nd, 3rd, 6th, 7th cases (C is the relevant case)
  if (PRONOUN is governed by PREPOSITION in the same case)
   {
    lemma-S
    lemma-P
   }
  else # PRONOUN is not governed by PREPOSITION in the same case
   {
    &#PprepC; lemma-S
    &#PprepC; lemma-P
   }

 exmpl: nic, nikdo, někdo # nothing, nobody, somebody
```

A.3.4 Adverbs

```
1st grade
  lemma

2nd grade [more ..]
  &#D2; lemma

3rd grade [most ..]
  &#Dthe; &#D3; lemma

&#Dnot; ... negative adverb (insert before lemma)
```

A.3.5 Verbs

```
Infinitiv [Vf]
 if (VERB is governed by "to be" in VBF) # budu pracovat
  {
   lemma-VB
  }
 else
  {
   &#Vto; lemma-VB
  }


Present Tense [VB------P]
 if (VERB is not governing 1st case node - no subject)
  {
     (*P is relevant PERSON, *N is relevant NUMBER)
   if ((PERSON == 3rd)&&(NUMBER == singular)) # (he) works
    {
     &#Vsub3S; lemma-VBZ
    }
   else # (we) work
    {
     &#Vsub*P*N; lemma-VB
    }
  }
 else # VERB is governing 1st case node - subject
  {
   if ((PERSON == 3rd)&&(NUMBER == singular)) # he works
    {
     lemma-VBZ
    }
   else # we work
    {
     lemma-VB
    }
  }


Future Tense [VB------F]
 if (VERB is not governing 1st case node - no subject)
  {
    (*P is relevant PERSON, *N is relevant NUMBER)
   if (LEMMA == "být")  # auxiliary verb "to be"
    {
     &#Vsub*P*N; být-VBF
    }
   else  # active future tense
    {
     &#Vsub*P*N; být-VBF lemma
    }
  }
 else # VERB is governing 1st case node - subject
  {
   if (LEMMA == "být")  # auxiliary verb "to be"
    {
     být-VBF
    }
   else  # active future tense
    {
     být-VBF lemma
    }
  }


Past Tense [Vp]
 if (VERB is not governing 1st case node - subject)
  {
     (*P is relevant PERSON, *N is relevant NUMBER)
   if ((PERSON == 3rd)&&(NUMBER == singular))
    {
     &#Vsub3SM; &#Vp; lemma-VBD ... masculinum
     &#Vsub3SF; &#Vp; lemma-VBD ... femininum
     &#Vsub3S; &#Vp; lemma-VBD .... other genders
    }
   else
    {
     &#Vsub*P*N; lemma-VBD
     if (VERB is governing verb "být" in VB)
      {
       change být-VB -> &#Vp;
      }
    }
  }
 else # VERB is governing 1st case node - subject
  {
     (*P is relevant PERSON, *N is relevant NUMBER)
   if ((PERSON == 3rd)&&(NUMBER == singular))
    {
     &#Vp; lemma-VBD
    }
   else
    {
     lemma-VBD
     if (VERB is governing verb "být" in VB)
      {
       change být-VB -> &#Vp;
      }
    }
  }
```

```
Imperativ [Vi]
 if (PERSON == 1) # Let's go
  {
   &#Vcomm1S; lemma-VB  ..... singular # Let me go
   &#Vcomm1P; lemma-VB  ..... plural   # Let us go
  }
 else # do it yourself!
  {
   &#Vcomm2; lemma-VB   # Go!
  }

Part of Past Passive Construction [Vs]
 if (VERB is not governing 1st case node - subject)
  {
    &#Vsub3S;  &#Vbe; lemma-VBX
  }
 else # VERB is governing 1st case node - subject
  {
    &#Vbe; lemma-VBX
  }

 exmpl: hotovo # it is done


Conditional [Vc]
    lemma-Vc

 exmpl: šla by # she would go


Gerund and archaic forms [Ve, Vm, Vq, Vt]
  lemma-V[emqt]


&#Vnot; ... negative verb (inserted before all aux words)
```

Appendix B

Samples of Dictionaries

We show the non-edited beginnings of several dictionaries.

B.1 Dictionary of Word Forms from the PCEDT

A	and
a	and
abandon	abandonment
abdikace	abdication
abdikaci	abdication
abdikoval	abdicated
abdikovat	abdicate
abdominální	abdominal
Abeceda	ABC
abeceda	ABC
Abeceda	alphabet
abeceda	alphabet
abecedně	alphabetically
abecední	alphabetic
abecedním	alphabetic
abecedou	ABC
abecedou	alphabet
abecedu	ABC
abecedu	alphabet
abecedy	ABC
abecedy	alphabet
abecedě	ABC
abecedě	alphabet
aberace	aberration
abnormální	abnormal
Abonent	subscriber
abonent	subscriber
abonenti	subscribers
abonentů	subscribers
Absence	absence
absence	absence
Absence	absences
absence	absences
Absenci	absence
absenci	absence
absencí	absence
absencí	absences
absentér	absentee
absentérství	absenteeism
absolutistický	absolute
Absolutně	absolutely
absolutně	absolutely
Absolutní	absolute
absolutní	absolute
Absolutní	essential
absolutní	essential
Absolutní	implicit
absolutní	implicit
Absolutní	positive
absolutní	positive
Absolutní	total
absolutní	total
Absolutní	unconditional
absolutní	unconditional
absolutních	absolute
absolutních	essential
absolutních	implicit
absolutních	positive
absolutních	total
absolutních	unconditional
absolutního	absolute
absolutního	essential
absolutního	implicit
absolutního	positive
absolutního	total
absolutního	unconditional
Absolutním	absolute
absolutním	absolute
Absolutním	essential
absolutním	essential
Absolutním	implicit
absolutním	implicit
Absolutním	positive
absolutním	positive
Absolutním	total
absolutním	total
Absolutním	unconditional
absolutním	unconditional
absolutními	absolute
absolutními	essential
absolutními	implicit
absolutními	positive
absolutními	total
absolutními	unconditional
absolutnímu	absolute
absolutnímu	essential
absolutnímu	implicit
absolutnímu	positive
absolutnímu	total
absolutnímu	unconditional
Absolvent	alumnus
absolvent	alumnus
Absolvent	graduate
absolvent	graduate
absolventa	alumnus
absolventa	graduate

absolventem	alumnus
absolventem	graduate
Absolventi	alumnus
absolventi	alumnus
Absolventi	graduates
absolventi	graduates
absolventy	alumnus
absolventy	graduates
absolventů	alumnus
absolventů	graduates
absolventům	alumnus
absolventům	graduates
Absolvoval	go_through
absolvoval	go_through
Absolvoval	went_through
absolvoval	went_through
Absolvovala	go_through
absolvovala	go_through
Absolvovala	went_through
absolvovala	went_through
Absolvovali	go_through
absolvovali	go_through
Absolvovali	went_through
absolvovali	went_through
absolvovalo	go_through
absolvovalo	went_through
absolvovaly	go_through
absolvovaly	went_through
absolvovat	go_through
absolvování	graduation
absolvuje	go_through
absolvuje	goes_through
absolvuje	going_through
absolvujeme	go_through
absolvujeme	going_through
absolvuji	go_through
absolvuji	going_through
absolvují	go_through
absolvují	going_through
absorboval	absorb
absorbovat	absorb
absorbuje	absorb
absorbuje	absorbing
absorbuje	absorbs
absorbují	absorb
absorbují	absorbing
absorpce	absorption
absorpci	absorption
abstence	abstention
abstinence	abstinence
abstinenci	abstinence
abstinentní	dry
abstinovat	abstain
abstrakce	abstraction
abstrakci	abstraction
abstrakcí	abstraction
abstraktnější	abstract
Abstraktní	abstract
abstraktní	abstract
abstraktních	abstract
abstraktního	abstract
abstraktním	abstract
abstraktními	abstract
abstraktnímu	abstract
absurdit	absurdity
Absurdita	absurdity
absurdita	absurdity
absurditou	absurdity
absurditu	absurdity
absurdity	absurdity
absurditě	absurdity
absurdnost	absurdity
absurdnosti	absurdity
absurdnější	absurd
absurdnější	inept
absurdnější	ludicrous
absurdnější	preposterous
Absurdní	absurd
absurdní	absurd
Absurdní	inept
absurdní	inept
Absurdní	ludicrous
absurdní	ludicrous
Absurdní	preposterous
absurdní	preposterous
absurdních	absurd
absurdních	inept
absurdních	ludicrous
absurdních	preposterous
absurdního	absurd
absurdního	inept
absurdního	ludicrous
absurdního	preposterous
absurdním	absurd
absurdním	inept
absurdním	ludicrous
absurdním	preposterous
absurdními	absurd
absurdními	inept
absurdními	ludicrous
absurdními	preposterous
absurdnímu	absurd
absurdnímu	inept
absurdnímu	ludicrous
absurdnímu	preposterous
ABY	from
Aby	from
aby	from
ABY	so
Aby	so
aby	so
ABY	that
Aby	that
aby	that
ABY	to
Aby	to
aby	to
Abych	from
abych	from
Abych	so
abych	so
Abych	that
abych	that
Abych	to
abych	to
Abychom	from
abychom	from
Abychom	so
abychom	so
Abychom	that
abychom	that
Abychom	to
abychom	to
Abys	from
abys	from
Abys	so
abys	so
Abys	that
abys	that
Abys	to
abys	to
Abyste	from
abyste	from
Abyste	so
abyste	so
Abyste	that
abyste	that

B.2 Terminological Dictionary for Computer Oriented Corpus

B.2.1 Sample of Czech-English dictionary of noun groups

absolutní název [14] absolute path name (1.00)
access/400 [17] access/400 (0.51), icon (0.49)
access [26] access (1.00)
access for [17] client access for (0.67), access for windows (0.33)
active files [12] active files (0.80), check (0.20)
active jobs [10] active jobs (1.00)
adaptéři [13] adapters (1.00)
adaptér [475] adapter (1.00)
adaptér LAN [23] LAN adapter (1.00)
adaptér POWER [15] FEATURE (0.51), ADAPTER (0.49)
adaptér ethernet [206] ethernet adapter (1.00)
adaptér systémové klávesnice [23] system keyboard adapter (1.00)
adaptéry [17] adapters (0.59), adapters (0.41)
additional parameters [16] additional parameters (1.00)
address [25] address (1.00)
administrátoři [15] administrators (0.57), administrator (0.43)
administrátor [69] administrator (1.00)
administrátor systému [13] security administrator (0.69), system administrator (0.31)
administrativa [10] administration (0.70), type (0.30)
administrativa lotus notes [10] lotus notes administration (1.00)
adresář [618] directory (1.00)
adresář CAWIN [11] CAWIN directory (1.00)
adresáře [175] directories (1.00)
adresa [323] address (1.00)
adresa IP [50] IP address (1.00)
adresa paměti [21] bus memory address (0.59), address of bus memory (0.41)
adresa uživatele [12] prompts (1.00)
adresa zařízení [18] device address (1.00)
adresy [118] address book (0.84), addresses (0.16)
adresy IP [11] IP addresses (0.64), classes (0.18), name server (0.18)
agent [15] agent (1.00)
aid [21] aid (1.00)
akce [163] action (0.76), actions (0.24)
aktivace [13] activation (0.33), active (0.22), enable (0.22), activate (0.22)
aktivita [10] activity (0.73), subsystem (0.27)
aktivní funkční klávesy [11] active function keys (0.49), keys (0.26), list (0.25)
aktivní soubor [11] active file (1.00)
aktuální čas [19] current time (0.79), time of day (0.21)
aktuální adresář [31] current directory (1.00)
aktuální datum [11] current date (1.00)
aktuální hodnota [12] current value (1.00)
aktuální hodnoty [16] current values (1.00)
aktuální kódová kombinace [12] digit (0.29), except (0.27), last (0.22), current combination (0.22)
aktuální knihovna [22] current library (1.00)
aktuální kombinace [10] last number of the current combination (0.49), current combination (0.34), digits (0.17)
aktuální obrazovka [13] current display (1.00)
aktuální systém [16] current system (1.00)
aktualizace [136] update (0.71), updates (0.29)
alias [29] alias (1.00)

B.2.2 Sample of English-Czech dictionary of noun groups

ability [44] schopnost (0.68), možnost (0.32)

absolute path name [13] absolutní název (1.00)

access/400 [19] access/400 (0.57), ikona ((0.43)

access [306] přístup (1.00)

access codes [14] přístupové kódy (1.00)

access control list [14] seznam (0.30), seznam přístupových práv (0.28), řízení přístupu (0.21), přístup (0.21)

access error [140] chyba přístupu (1.00)

access path [57] přístupová cesta (1.00)

access path recovery [12] přístupová cesta (0.30), doba obnovy (0.29), doby obnovy (0.29), přístupová cesty (0.12)

access paths [77] přístupové cesty (0.63), přístupová cesta (0.19), přístupové cesta (0.17)

access paths display [11] paths (0.32), access (0.31), rebuild (0.19), of (0.18)

access plan [10] přístupový plán (1.00)

acknowledgement [15] potvrzení (1.00)

action [194] akce (0.77), činnost (0.23)

actions [84] činnost (0.51), akce (0.49)

active file [10] aktivní soubor (1.00)

active files [17] active files (0.67), kontrola aktivních souborů (0.33)

active jobs [15] active jobs (0.63), příkaz work (0.37)

activities [35] činnost (0.70), činnosti (0.30)

activity [21] aktivita (0.40), činnost (0.26), záznam (0.17), činnosti (0.17)

adapter/A [19] portmaster (0.49), adaptér/A (0.25), jeden‘1 (0.13), port (0.13)

adapter [395] adaptér (1.00)

adapter device driver [10] adaptér (0.57), ovladač zařízení (0.43)

adapter hardware [10] adaptér (1.00)

adapter test [11] test adaptér (0.56), portový adaptér (0.22), test adaptéru (0.22)

adapters [13] adaptéry (1.00)

addition [119] kromě (0.63), navíc (0.37)

additional considerations [10] další pokyny (0.82), zahájení instalace (0.18)

additional disk units [12] další diskové jednotky (1.00)

additional hardware [18] hardware (0.52), dodatečný (0.33), dodatečný hardware (0.15)

additional information [100] další informace (1.00)

additional licensed programs [40] další licenční programy (1.00)

additional message information [14] obrazovka additional message information (1.00)

additional parameters [28] additional parameters (0.70), další parametry (0.30)

additional server [13] server (0.51), přídavný server (0.49)

additions [10] dodatky (0.68), vyšší verze (0.32)

address [348] adresa (1.00)

address book [69] adresy (0.69), kniha jmen (0.31)

address book field [10] adresy (0.55), kniha jmen (0.45)

addresses [23] adresy (0.83), hostitelské systémy (0.17)

administration [19] administration (0.67), administrativa (0.33)

administrator [81] administrátor (0.87), správce (0.13)

administrator workstation [11] pracovní stanice administrátora (0.65), pracovní stanice (0.18), podpora (0.16)

administrators [14] administrátoři (1.00)

advance [13] předem (0.42), činnost (0.38), programové vybavení (0.21)

advanced assistance level [10] rozšířená úroveň pomoci (1.00)

advanced function printing [23] rozšířené funkce (0.41), tisk (0.33), advanced function printing (0.26)

advantage [12] využívat (0.38), rozšířené funkce (0.32), výhody (0.30)

agent [18] agent (1.00)

Appendix C

Sample Outputs from Experimental MT Systems

This appendix includes sample translations from experimental translation systems (Sections C.5 – C.8). The original PennTreebank reference (document index 2435) is shown in Section C.1. Section C.2 shows its Czech translation included in the development set of the PCEDT, which was used as an input to all translation systems. Sections C.3 and C.4 display two reference retranslations which were used for the BLEU evaluation.

C.1 Original PTB Reference

*[**1**] Matsushita Electric Industrial Co. of Japan and Siemens AG of West Germany announced they have completed a 100 million-mark ($52.2 million) joint venture to produce electronics parts.*
*[**2**] In the venture's first fiscal year, Siemens will hold 74.9% of the venture and a Matsushita subsidiary, Matsushita Electronic Components Co., 25.1%.*
*[**3**] A basic agreement between the two companies was announced in June.*
*[**4**] The new company is to be called Siemens Matsushita Components G.m.b.H.*
*[**5**] It will have its headquarters in Munich.*
*[**6**] Matsushita's share in the venture will rise to 35% Oct. 1, 1990, and to 50% the following Oct. 1.*
*[**7**] Siemens will retain majority voting rights.*
*[**8**] The parent companies forecast sales for the venture of around 750 million marks for its first fiscal year, Matsushita said.*
*[**9**] Sales are expected to rise to one billion marks after four years.*
*[**10**] The company will have production facilities in West Germany, Austria, France and Spain.*

C.2 Czech Translation from PCEDT (source for all MT systems)

*[**1**] Firmy Matsushita Electric Industrial Co. z Japonska a Siemens AG ze Západního Německa oznámily, že uzavřely joint-venture na výrobu elektronických součástek v hodnotě 100 milionů marek (52,2 milionů $).*
*[**2**] V prvním fiskálním roce tohoto podniku bude Siemens vlastnit 74,9% podniku a dceřiná společnost Matsushity, Matsushita Electronic Components Co., 25,1%.*
*[**3**] Základní dohoda mezi těmito dvěma společnostmi byla oznámena v červnu.*
*[**4**] Nová společnost se má jmenovat Siemens Matsushita Components G.m.b.H.*
*[**5**] Své ústředí bude mít v Mnichově.*
*[**6**] Podíl Matsushity v tomto podniku se zvýší 1. října 1990 na 35% a 1. října následujícího roku na 50%.*
*[**7**] Siemens si ponechá většinová hlasovací práva.*
*[**8**] Mateřské společnosti předpovídají v prvním fiskálním roce tržby podniku okolo 750 milionů marek, uvedli zástupci Matsushity.*
*[**9**] Očekává se, že tržby se po čtyřech letech zvýší na 1 miliardu marek.*
*[**10**] Společnost bude mít výrobní zařízení v Západním Německu, Rakousku, Francii a Španělsku.*

C.3 Reference Retranslation to English from PCEDT (Sample 1)

*[**1**] The firms Matsushita Electric Industrial Co. of Japan and Siemens AG of West Germany announced that they had concluded a joint-venture on the production of electronic components worth 100 million marks (USD 52.2 million).*
*[**2**] During the first fiscal year of that company, Siemens will own 74.9% of that company and Matsushita´s subsidiary company, Matsushita Electronic Components Co., 25.1%.*
*[**3**] A basic agreement between those two companies was announced in June.*
*[**4**] The new company is to be called Siemens Matsushita Components G.m.b.H.*
*[**5**] It is supposed to have its headquarters in Munich.*
*[**6**] On October 1, 1990, the share of Matsushita in that company will increase to 35% and to 50% on October 1 in the following year.*
*[**7**] Siemens will keep all of its major votes.*
*[**8**] The parent companies foresee that the company's revenues will be around 750 million marks during the first fiscal year, mentioned representatives of Matsushita.*
*[**9**] It is expected that after four years revenues will increase to 1 billion marks.*
*[**10**] The company will have production facilities in West Germany, Austria, France and Spain.*

C.4 Reference Retranslation to English from PCEDT (Sample 2)

*[**1**] Matsushita Electric Industrial Co. of Japan and Siemens AG of West Germany announced they've created a joint venture for manufacturing electronic parts in the value of 100 million marks ($52.2 million).*
*[**2**] In the company's first fiscal year Siemens will hold 74.9% of the venture and Matsushita's subsidiary, Matsushita Electronic Components Co. will hold 25.1%.*
*[**3**] The basic agreement between the two companies was announced in June.*
*[**4**] The new company is to be named Siemens Matsushita Components G.m.b.H.*
*[**5**] It will have its headquarters in Munich.*
*[**6**] Matsushita's share in this venture will increase to 35% on October 1, 1990, and to 50% on October 1 of the following year.*
*[**7**] Siemens will retain the majority vote.*
*[**8**] Matsushita's representatives stated the parent companies project the venture's sales to be about 750 million marks in the first fiscal year.*
*[**9**] In four years the sales are expected to increase to 1 billion marks.*
*[**10**] The company will have production facilities in West Germany, Austria, France and Spain.*

C.5 Output from Dependency-based Machine Translation System

An output from the dependency-based machine translation system (with full transfer dictionary and language model rescoring) described in Section 5.2 and evaluated in 6.3 as system **(d)**, $BLEUscore = 0.1921$.

*[**1**] The firms Matsushita Electric Industrial Co. from Japan and Siemens AG from West Germany said a Motors Venture closed the production of electronic elements in the amount the 100 million of marks the 52,2 million.*
*[**2**] At the first fiscal year that company will be Siemens vlasnit 74,9% of company and the filial company Matsushita Matsushita Electronic Components Co., 25,1.*
*[**3**] The institutional agreement in that two companies say at a June.*
*[**4**] The latest company are supposed to name Siemens Matsushita Components G.m.b.H.*
*[**5**] The of he base have in Munich.*
*[**6**] The share Matsushita in that company increases the 1 Oct. 1990 35% and the 1 Oct. of a follow year 50%.*
*[**7**] Siemens retains majority voting rights.*
*[**8**] At the first fiscal year the parent companies predict the sale of company the 750 million of marks said the attorneys Matsushita.*
*[**9**] At the four years the sales increases the 1 billion of marks expects he.*

C.6 Output from SMT System without Preprocessing

*An output from the SMT system trained on raw-text and evaluated in 6.2 as system **(A)**,* $BLEUscore = 0.3592$.

*[**1**] Firms Matsushita Electric Industrial Co., a joint venture to build a $ 100 million marks of electronic parts and Siemens AG of West Germany reported that the close of 52,2 million.*
*[**2**] In the fiscal year, the venture will Siemens Matsushita Electronic Co., 25,1%. Components Matsushity, the company's and % dceřiná the first vlasnit 74,9*
*[**3**] The agreement between the two companies was announced in June.*
*[**4**] The new company is called Siemens Matsushita Components G.m.b.H.*
*[**5**] He will have headquarters in Munich.*
*[**6**] The venture will rise to Oct. 1, 1990, 35, and Oct. 1 to 50% the next year. Matsushity this %*
*[**7**] Siemens will retain většinová hlasovací rights.*
*[**8**] Mateřské the company's sales of about $ 750 million marks in the first predict the fiscal year, officials said Matsushity.*
*[**9**] The four years after the sales increase to $ 1 billion marks.*
*[**10**] The company will have manufacturing facilities in West Germany, France and Spain, Rakousku.*

C.7 Output from SMT with Lemmatized Input

An output from the SMT system trained on lemmatized Czech and evaluated in 6.2 as system **(B)**, $BLEUscore = 0.3747$.

*[**1**] Matsushita Electric Industrial Co. and Siemens AG of Germany said it entered a joint venture to build a $ 100 million marks of electronic parts of the West 52,2 million.*
*[**2**] The venture will Siemens Matsushita, a unit of Matsushita Electronic Co., 25,1% in the first fiscal year vlasnit 74,9% of the business and Components.*
*[**3**] Basic agreement between the two companies announced in June.*
*[**4**] The company has appointed a new Siemens Matsushita Components G.m.b.H.*
*[**5**] Its headquarters is in the Mnichov.*
*[**6**] Stake in the Oct. 1, 1990, to 35% and 50% on Oct. 1 next year Matsushita increased.*
*[**7**] Siemens will retain a majority voting.*
*[**8**] The parent company of predicts sales of about 750 million marks the first in the fiscal year, Matsushita, officials said.*
*[**9**] The after four years to increase sales of $ 1 billion marks.*
*[**10**] Manufacturing facility in West Germany, France and Spain, the company has Rakousko.*

C.8 Output from SMT with *Czech'* Preprocessing

An output from the SMT system trained on the corpus with *Czech'* preprocessing and evaluated in 6.2 as system **(C)**, $BLEUscore = 0.3858$.

*[**1**] The Matsushita Electric Industrial Co. – $ 100 million marks (52,2 million) from Japan and Siemens AG of West Germany, said it closed joint Venture, to make electronic parts in.*
*[**2**] The venture will Siemens midst of vlasnit 74,9% of the venture and a subsidiary Matsushita, Matsushita Electronic Components Co., 25,1% in the first fiscal year.*
*[**3**] The basic agreement between the two companies was announced in June.*
*[**4**] The new company is expected to name Components G.m.b.H. Siemens Matsushita.*
*[**5**] It's going to have its headquarters in Munich.*
*[**6**] Stake in the venture will increase to 35% on Oct. 1, 1990 and Oct. 1, the following year Matsushita to 50%.*
*[**7**] Siemens will retain the majority voting rights.*
*[**8**] Parent companies in the first fiscal year in sales of about 750 million marks of Matsushita, officials said the venture predict.*
*[**9**] It is expected that sales will increase to $ 1 billion marks, after four years.*
*[**10**] The company will have production facilities in West Germany, France and Spain, Rakousko.*

Bibliography

[Al-Onaizan et al., 1999] Al-Onaizan, Y., Cuřín, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., and Yarowsky, D. (1999). The statistical machine translation. Technical report. Proceedings of the Summer Workshop on Language Engineering, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

[Berger et al., 1994] Berger, A., Brown, P., Della-Pietra, S., Della-Pietra, V., Gillett, J., Lafferty, J., Mercer, R., Printz, H., and Ureš, L. (1994). The Candide system for machine translation. In *Proceedings of the ARPA Human Language Technology Workshop*.

[Böhmová, 2001] Böhmová, A. (2001). Automatic procedures in tectogrammatical tagging. *The Prague Bulletin of Mathematical Linguistics*, 76.

[Böhmová et al., 2005] Böhmová, A., Cinková, S., and Hajičová, E. (2005). A Manual for Tectogrammatical Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic.

[Brown et al., 1988] Brown, P., Cocke, J., Pietra, S. D., Jelinek, F., Mercer, R., and Roossin, P. (1988). A statistical approach to language translation. In *COLING-88*.

[Brown et al., 1990] Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Lafferty, J., Mercer, R., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2).

[Brown et al., 1995] Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Lai, J., and Mercer, R. (1995). U.S. Patent #5,477,451: Method and system for natural language translation.

[Brown et al., 1992] Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

[Brown et al., 1993] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

[Charniak, 1999] Charniak, E. (1999). A maximum-entropy-inspired parser. Technical Report CS-99-12.

[Čmejrek, 1998] Čmejrek, M. (1998). Automatická extrakce dvojjazyčného pravděpodobnostního slovníku z paralelních textů. Master's thesis, Charles University, Prague. In Czech.

[Čmejrek et al., 2005] Čmejrek, M., Cuřín, J., Hajič, J., and Havelka, J. (2005). Prague Czech-English Dependency Treebank: Resource for Structure-based MT. In Hutchins, J., Kis, B., and Prószéky, G., editors, *Proceedings of the 10th EAMT Conference*, pages 73–78, Budapest, Hungary. European Association for Machine Translation. ISBN 963 9206 04 0.

[Čmejrek et al., 2003a] Čmejrek, M., Cuřín, J., and Havelka, J. (2003a). Czech-English dependency-based machine translation. In *Proceedings of the 10th Conference of The European Chapter of the Association for Computational Linguistics*, pages 83–90, Budapest, Hungary.

[Čmejrek et al., 2003b] Čmejrek, M., Cuřín, J., and Havelka, J. (2003b). Treebanks in machine translation. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, pages 209–212, Vaxjo, Sweden.

[Čmejrek et al., 2004a] Čmejrek, M., Cuřín, J., and Havelka, J. (2004a). Prague Czech-English Dependency Treebank: Any Hopes for a Common Annotation Scheme? In *Frontiers in Corpus Annotation. Proceedings of the Workshop of the HLT/NAACL Conference*, pages 47–54. MSM113200006, LN00A063.

[Čmejrek et al., 2004b] Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., and Kuboň, V. (2004b). Prague Czech-English Dependency Treebank. Syntactically Annotated Resources for Machine Translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, volume V, pages 1597–1600, Lisboa, Portugal. European Language Resources Association. ISBN 2-9517408-1-6.

[Cuřín, 1998] Cuřín, J. (1998). Automatická extrakce překladu odborné terminologie. Master's thesis, Charles University, Prague. In Czech.

[Cuřín and Peterek, 2000] Cuřín, J. and Peterek, N. (2000). The experimental results from NSF Workshop'99, CLSP Johns Hopkins University, Czech/English statistical machine translation and automatic speech recognition. *The Prague Bulletin of Mathematical Linguistics*, 73–74:63–76.

[Cuřín and Čmejrek, 1999] Cuřín, J. and Čmejrek, M. (1999). Automatic translation lexicon extraction from Czech-English parallel texts. *The Prague Bulletin of Mathematical Linguistics*, 71:47–57.

[Cuřín and Čmejrek, 2001] Cuřín, J. and Čmejrek, M. (2001). Automatic extraction of terminological translation lexicon from Czech-English parallel texts. *International Journal of Corpus Linguistics*, 6(Special Issue):1–12.

[Cuřín et al., 2002] Cuřín, J., Čmejrek, M., and Havelka, J. (2002). Czech-English dependency-based machine translation: Data preparation for the starting up experiments. *The Prague Bulletin of Mathematical Linguistics*, 78:103–116.

[Cuřín et al., 2004] Cuřín, J., Čmejrek, M., Havelka, J., Hajič, J., Kuboň, V., and Žabokrtský, Z. (2004). Prague Czech-English Dependency Treebank, version 1.0. Linguistic Data Consortium (LDC).

[Cuřín et al., 2005] Cuřín, J., Čmejrek, M., Havelka, J., and Kuboň, V. (2005). Building a parallel bilingual syntactically annontated corpus. In Keh-Yih Su, Jun'ichi Tsujii, J.-H. L. e. a., editor, *Natural Language Processing - IJCNLP 2004: First International Joint Conference, Hainan Island, China, March 22-24, 2004, Revised Selected Papers*, volume 3248 of *LNAI*, pages 168–176. ISBN: 3-540-24475-1.

[Gale and Church, 1993] Gale, W. and Church, K. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1).

[Germann et al., 2001] Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 228–235.

[Hajič, 2004] Hajič, J. (2004). *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Karolinum, Charles University Press, Prague, Czech Republic.

[Hajičová et al., 1999] Hajičová, E., Kirschner, Z., and Sgall, P. (1999). A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic.

[Hajič, 1987] Hajič, J. (1987). Ruslan: An mt system between closely related languages. In *Proc. of the 3rd EACL*, pages 113–117, Copenhagen, Denmark.

[Hajič et al., 1998] Hajič, J., Brill, E., Collins, M., Hladká, B., Jones, D., Kuo, C., Ramshaw, L., Schwartz, O., Tillmann, C., and Zeman, D. (1998). Core Natural Language Processing Technology Applicable to Multiple Languages. Technical Report Research Note 37, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

[Hajič et al., 1992] Hajič, J., Hajičová, E., Hnátková, M., Kuboň, V., Panevová, J., Rosen, A., Sgall, P., and Skoumalová, H. (1992). MATRACE - MAchine TRAnslation between Czech and English. Technical report.

[Hajič and Hladká, 1998] Hajič, J. and Hladká, B. (1998). Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of COLING-ACL Conference*, pages 483–490, Montreal, Canada.

[Hajič et al., 2000a] Hajič, J., Kuboň, V., and Hric, J. (2000a). Česílko - an MT system for closely related languages. In *Tutorial Abstracts and Demonstration Notes at ACL Conference*, pages 7–8, Washington.

[Hajič et al., 2000b] Hajič, J., Kuboň, V., and Hric, J. (2000b). Machine translation of very close languages. In *In Proceedings of 6th ANLP Conference / 1st NAACL Meeting*, pages 7–12, Seattle, Washington.

[Hajič et al., 2002] Hajič, J., Čmejrek, M., Dorr, B., Ding, Y., Eisner, J., Gildea, D., Koo, T., Parton, K., Penn, G., Radev, D., and Rambow, O. (2002). Natural Language Generation in the Context of Machine Translation. Technical report. NLP WS'02 Final Report.

[Hana et al., 2005] Hana, J., Zeman, D., Hajič, J., Hanová, H., Hladká, B., and Jeřábek, E. (2005). Manual for Morphological Annotation, Revision for the Prague Dependency Treebank 2.0. Technical Report TR-2005-27, ÚFAL MFF UK, Prague, Czech Rep.

[Jelinek, 1969] Jelinek, F. (1969). A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13:675–685.

[Jelinek, 1985] Jelinek, F. (1985). Self-organized language modeling for speech recognition. Technical report, IBM Research. Reprinted in A. Waibel, K. F. Lee (eds.), *Readings in Speech Recognition.*

[Jelinek et al., 2000] Jelinek, F., Byrne, W., Khudanpur, S., Hladka, B., Ney, H., Och, F., Cuřín, J., and Psutka, J. (2000). Robust knowledge discovery from parallel speech and text sources. In *Proceedings of the Human Language Technology Conference*, pages 299–301, San Diego, CA.

[Kirschner, 1987] Kirschner, Z. (1987). *APAC3-2: An English-to-Czech Machine Translation System.* MFF UK.

[Kirschner and Rosen, 1989] Kirschner, Z. and Rosen, A. (1989). APAC - An Experiment in Machine Translation. *Machine Translation*, 4:177–193.

[Knight, 1999] Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

[Knight and Al-Onaizan, 1998] Knight, K. and Al-Onaizan, Y. (1998). Translation with finite-state devices. In *Proc. AMTA.*

[Koehn, 2004] Koehn, P. (2004). Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

[Koehn et al., 2003] Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *HLT-NAACL*, pages 127–133, Edmonton, Canada.

[Krbec, 2005] Krbec, P. (2005). *Language Modeling for Speech Recognition of Czech*. PhD thesis, Institute of Formal and Applied Linguistics at the Faculty of Mathematics and Physics, Charles University, Prague.

[Langkilde, 2000] Langkilde, I. (2000). Forest-based statistical sentence generation. In *Proceedings of NAACL'00*, Seattle, WA.

[Linguistic Data Consortium, 1995] Linguistic Data Consortium (1995). North American News Text Corpus. LDC95T21.

[Linguistic Data Consortium, 1999] Linguistic Data Consortium (1999). Penn Treebank 3. LDC99T42.

[Linguistic Data Consortium, 2001] Linguistic Data Consortium (2001). Prague Dependency Treebank 1. LDC2001T10.

[Melamed, 1996] Melamed, I. D. (1996). A geometric approach to mapping bitext correspondence. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*.

[Melamed, 2001] Melamed, I. D. (2001). *Empirical Methods for Exploiting Parallel Texts*. MIT Press.

[Minnen et al., 2001] Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223.

[Och, 2002] Och, F. J. (2002). *Statistical Machine Translation: From Sigle-Word Models to Alignment Templates*. PhD thesis, RWTH, Aachen, Germany.

[Och and Ney, 2000] Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hongkong, China.

[Och and Ney, 2003] Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

[Och et al., 1999] Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*.

[Papineni et al., 2001] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176, IBM.

[Petkevič, 1999] Petkevič, V. (1999). Czech Translation of G. Orwell's '1984': Morphology and Syntactic Patterns in the Corpus. In *TSD*, pages 77–82.

[Ratnaparkhi, 1996] Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger publication. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 17–18, University of Pennsylvania.

[Rosen, 2005] Rosen, A. (2005). In search of the best method for sentence alignment in parallel texts. In *SLOVKO – Computer Treatment of Slavic and East European Languages*, Bratislava, Slovakia.

[Sgall et al., 1986] Sgall, P., Hajičová, E., and Panevová, J. (1986). *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects.* Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands.

[Stolcke, 2002] Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceeding of 7th International Conference on Spoken Language Processing*, Denver, Colorado.

[Svoboda, 2004] Svoboda, M. (2004). Anglicko-český slovník. University of West Bohemia in Pilsen, http://slovnik.zcu.cz.

[Vauquois, 1975] Vauquois, B. (1975). La traduction automatique 'a grenoble.

[Weaver, 1955] Weaver, W. (1955). Translation. *Machine Translation of Languages*, pages 15–27. Reprint of 1949 memo.

[Zeman et al., 2005] Zeman, D., Hana, J., Hanová, H., Hajič, J., Hladká, B., and Jeřábek, E. (2005). A Manual for Morphological Annotation, 2nd edition. Technical Report 27.

[Žabokrtský and Kučerová, 2002] Žabokrtský, Z. and Kučerová, I. (2002). Transforming penn treebank phrase trees into (praguian) tectogrammatical dependency trees. *The Prague Bulletin of Mathematical Linguistics*, 78.

[Žabokrtský et al., 2002] Žabokrtský, Z., Sgall, P., and Sašo, D. (2002). Machine learning approach to automatic functor assignment in the prague dependency treebank. In *Proceedings of LREC 2002 (Third International Conference on Language Resources and Evaluation)*, volume V, pages 1513–1520, Las Palmas de Gran Canaria, Spain.