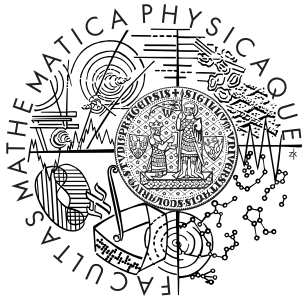
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Jan Rouš

Probabilistic Translation Dictionary

Institute of Formal and Applied Linguistics

Supervisor: Ing. Zdeněk Žabokrtský, Ph.D.
Study programme: Computer Science
Study field: Mathematical Linguistics

Prague, 2009

I would like to thank my supervisor Zdeněk Žabokrtský for his time, his inspiring ideas and enormous amount of patience he had with me. I would also like to thank my parents for their never-ending questions about when I'm going to finish this thesis. I'd like to thank Bětka for her patience, Stáňa for her support and review and Maruška for her joy when I told her that I've finally completed the work. I'd like to thank A.A.Milne for writing the amazing book Winnie the Pooh that helped me overcome bad moments.

I certify that this diploma 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, July 6, 2009 Jan Rouš

Contents

Title: Probabilistic translation dictionary
Author: Jan Rouš
Department: Institute of Formal and Applied Linguistics
Supervisor: Ing. Zdeněk Žabokrtský, Ph.D.
Supervisor's e-mail address: zabokrtsky@ufal.mff.cuni.cz

Abstract:

In this work we present the method of semi-automatic training of the probabilistic translation dictionary using large automatically annotated parallel corpora. According to the study of translation errors and the role of translation dictionary within the TectoMt translatio system in general we propose models of various complexity. These basic models were combined to hierarchical models that were designed to reduce impact of the sparse data problem. Various extensions were implemented to deal with common lexical errors. The dictionary along with extensions was compared to the former approach on test data and the results show improved translation quality.
Keywords: machine translation, translation dictionary, tectogrammatical layer, TectoMT

Název práce: Pravděpodobnostní překladový slovník
Autor: Jan Rouš
Katedra (ústav): Ústav formální a aplikované lingvistiky
Vedoucí diplomové práce: Ing. Zdeněk Žabokrtský, Ph.D.
e-mail vedoucího: zabokrtsky@ufal.mff.cuni.cz

Abstrakt:
V této práci popisujeme poloautomatickou metodu trénování pravděpodobnostního překladového slovníku z rozsáhlých automaticky anotovaných paralelních korpusů. Na základě studia překladových chyb a funkce slovníku v rámci překladového systému TectoMT obecně byly navrženy modely různé složitosti. Tyto základní modely byly zkombinovány do hierarchických modelů, jejichž účel je snížit dopad problému řídkých dat. Slovník byl doplněn o rozšíření, která jsou navržena tak, aby odstraňovala časté problémy lexikálního charakteru. Slovník spolu s rozšířeními byl na testovacích datech porovnán s původním slovníkem a výsledky ukazují, že došlo k zvýšení kvality překladu.
Klíčová slova: strojový překlad, překladový slovník, tektogramatická rovina, TectoMT

Chapter 1

Introduction

1.1 Machine translation today

In nowadays world where communication between subjects from all around the globe is a part of everyday life we often communicate with people using different languages and encounter information written in languages we do not know or maybe are not skilled enough to understand well. Employing human translators might not be optimal solution in many cases. Reliable high quality translation is offered by human translators but it costs quite a lot of money and time. On contrary there are publicly available machine translation systems (Google Translate, Babelfish) offering instant translation for free. Modern era of internet and its services proved that MT[1] is not just another theoretical toy of computational linguists and enthusiast academics anymore but has finally found its place in real world applications – instant translation of articles and web content can help masses of users through fast and easy-to-use interfaces. Though the development and translation systems evolve rapidly we are still far away from the the ultimate goal of machine translation which is to provide a reliable high quality translation system that can correctly parse and translate all aspects and layers of natural language (including semantic and pragmatic aspects). Though systems of today offer somewhat rough and imprecise, sometimes garbled or even totally incorrect translation their service is still valuable one. With the help of our own human post-processing abilities, prediction and knowledge of the context we can possibly deduce the meaning of the text which would not be possible if we had only the untranslated source at hand.

[1] Machine Translation

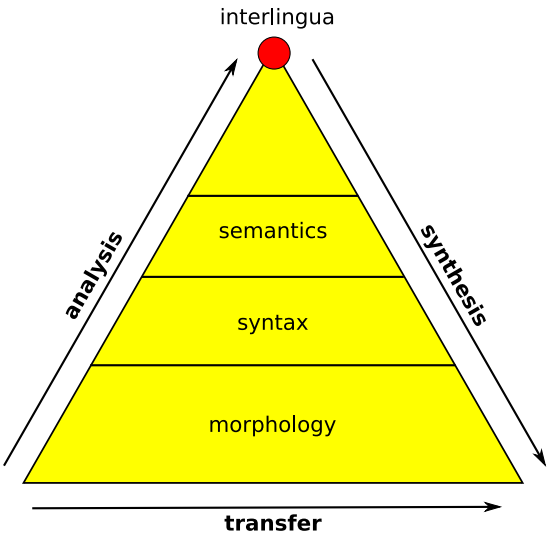


Figure 1.1: Vauquois triangle

1.2 Typology of MT systems

Although there are major differences between various approaches to machine translation, most of the translation pipelines can be depicted as a sequence of consecutive steps involving *analysis* of the source language, *transfer* between analyzed structures and *synthesis* of the target language. Various MT systems can be classified according to the level of analysis on which they perform the transfer. The more complex and rich analysis we have, the more information is present that can aid the translation. On the other hand, more effort is required for both the analysis and synthesis and there is also an added element of errors induced during these two phases. MT systems can be orthogonally classified according to the type of algorithms they use for the translation process. Rule-based systems rely on hand-written translation rules – we can think of them as the parallel grammars. Construction of hand-written grammar is a very expensive process. Mainly because natural languages tend to be quite irregular and require complex grammars with a great number of rules. Moreover only people with really deep insight into the inner workings of the language can analyze and formulate these rules. Many modern translation systems tend to learn the grammar statistically from the corpora. Most of the state-of-the-art MT systems are in fact representatives of the SMT[2] approach. We will now describe various approaches to the problem of machine translation. Note that this typology is very simple. For further information about this topic and the underlying theory see (Hutchins and Somers, 1992).

[2]Statistical Machine Translation

Widely proposed in theoretical papers on MT but never practically implemented was the concept of Interlingua translation. The transfer between two language-dependent counterpart representations could be eliminated if we were able to transform the sentences to the language-independent representation of information – so called Interlingua. The most appalling feature of Interlingua-based translation system unfolds when we are trying to construct a multi-language system that should be able to translate freely between n different languages. In a traditional system we would have to build $n(n-1)$ language-dependent transfer pipelines along with n analysis and synthesis blocks whereas Interlingua-based system requires only n translation pipelines transforming text from a given language to Interlingua and back. Nevertheless no Interlingua-based system has been developed yet and the following crucial drawbacks can explain why:

1. There is no implementation for the Interlingua itself nor there is an agreement on what kind of system should be used to encode information. While some have proposed use of some kind of logic language others have proposed construction or use of an artificial human language such as Esperanto.

2. Interlingua should contain full information that is embedded in the language including semantic, pragmatic and discourse layers. However the analysis gets much harder for higher levels of annotation such as pragmatics. There is no agreement about what the layer should look like nor there are large enough corpora annotated to such extent available. Therefore there are not any tools for the automatic annotation yet.

3. Even the basic concept of language-independent representation of information can be objected against. For arbitrary pair of languages we could find many phenomena that are handled in a different ways in both languages. There are numerous cases where ontologies are more detailed for certain field in certain language. In Finnish[3] for example there exists variety of words to describe different kinds of *snow*; in Spanish we use *pez* to describe living species of fish and *pescado* when we talk about fish as a food. Fully functional Interlingua should be able to distinguish between these terms. This would lead to very complex universal ontology that has many disadvantages. For example if we are translating from language where certain distinction is not present we are not able to select the correct term directly. Moreover we are basically not interested in the fine-grained ontology for *kinds of snow* if we are not dealing with Finnish.

Hence nowadays translation systems usually rely on more restricted level of analysis. We can divide MT systems into the following categories according to

[3]For details see `http://everything2.com/title/Finnish+words+for+snow`

the level of analysis (or we could say complexity of representation) they perform before proceeding to transfer:

- word-based MT systems,
- phrase-based MT systems,
- syntax-based MT systems,
- higher-order MT systems.

1.2.1 Word-based MT systems

Word-based translation uses isolated words as its basic unit for modelling the language. They can be thought of as a first attempt to grasp the probabilistic nature of translation. They are quite simple and thus quite suitable for teaching purposes. But they are too simple for practical use and other more complex models have provided better results. Notable examples of word-based MT systems are IBM models (often described in the introductory lessons to SMT). GIZA++ (Och and Ney, 2003) training and alignment tool for IBM, HMM and Model 6 translation systems are also well known and are still used in nowadays systems for the word alignment.

1.2.2 Syntax-based MT systems

Syntax-based systems are usually operating on layers where sentences are represented by tree-like structures however the actual form and content of these trees can vary heavily among different MT systems. In this group both systems performing transfer on syntactic or on semantic layer can be included as both are usually working with parallel trees containing surface or deep syntax representation of the sentence. Actual format of the trees can also vary – for English language it is quite common to use constituency trees whereas dependency trees are the traditional approach in the case of some languages with rich inflection and free word order such as Czech.

Compared to the phrase-based models this approach should provide more information as the relations between the words is encoded within the tree. Long-distance relations can be also expressed in this way. However this approach is still less successfull than the phrase-based approach. It can be attributed to the increased complexity of the analysis, propagation (and further spreading) of errors from lower to higher level of analysis, sparseness of the structures we are working with and high diversity of tree representations in the different languages.

1.2.3 Phrase-based MT systems

Phrase-based systems rely on phrases (hunks of words) as their basic units. Opposed to the syntax-based systems which operate on explicit representations of sentence structure, phrase-based systems work more with the implicit structure that is encoded in the phrases. They don't need to know exactly how the words within the phrase relate to each other. They are interested only in the translation patterns on the phrase level. We should point out that phrases are not understood in a purely linguistic meaning. Moreover it was shown that phrase-based systems give better results if they are not limited to linguistic phrases only. Though there is no explicit need for preprocessing and analysis of the text we could further boost the performance by applying preprocessing such as lemmatization or tagging especially when dealing with highly-inflectional languages as can be seen in (Zhang and Sumita, 2007). Most state-of-the-art MT systems are actually phrase-based. The most notable implementations are open-source MT system Moses (Koehn et al., 2007) or Google Translator.

1.2.4 Comparison

Although syntax and higher order levels seem to be promising in a way that they tend to build sharp and exact description of the language structure and they gradually work toward semantic representations that could be understood by the machines they tend to suffer from accumulating errors by putting together many steps that are not individually error-proof. This great complexity is probably the reason why the state-of-the-art SMT systems are usually built upon the simpler and more universal phrase-based frameworks. However, it is vital to further research these structural approaches as they tend to provide tools not just for the machine translation itself but also for other fields of computational linguistics.

1.3 TectoMT

TectoMT is a highly modular NLP[4] framework that aims to provide universal platform for various linguistic tasks. It was primarily developed for the purpose of machine translation but it can be used for a variety of other tasks. TectoMT contains a wide variety of assorted tools for various NLP tasks such as tokenization, morphological and syntactic analysis and many others. Re-usability and modularity were therefore emphasized a lot. Simple tasks are implemented within blocks that can be chained together creating a scenario. Scenarios with specific purpose can be thought of as applications. Applications within the TectoMT system are

[4]Natural Language Processing

usually just scenarios along with some simple Makefile-based interfaces. Various third-party tools were integrated into the systems by the means of simple block wrappers.

Layers of annotation used within the TectoMT system conform to the standards proposed in the PDT[5] project (Hajič et al., 2006):

- w-layer – sentences are represented as a stream of individual tokens,
- m-layer – tokens are extended with morphological analysis,
- a-layer – analytical representation of the sentence uses dependency trees to represent syntactic relations,
- t-layer – semantic dependency structure is used at the tectogrammatical layer.

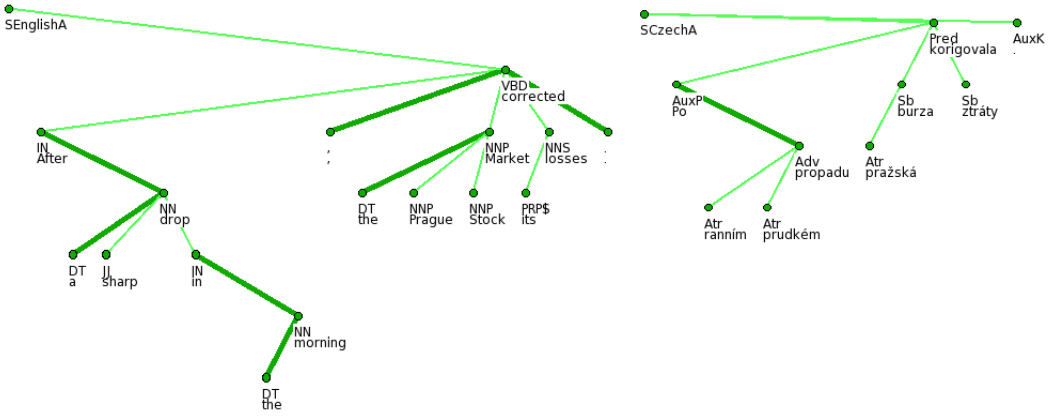
For the analysis up to the analytical layer a wide variety of tools for the necessary steps exists and various approaches of the morphological as well as syntactic analysis (both based on dependency as well as immediate constituents relations) exist and were intensively studied. Interesting step is the transformation between analytical and tectogrammatical layer. The latter tries to express the deep grammar of the sentences which is closely related to the semantic content (or logical structure) of the sentence. While the syntactic tree is transformed into the tectogrammatical one, non-autosemantic and auxiliary words are removed from the tree while some extra nodes are introduced (for example due to the ellipsis resolution or due to the verb valency).

As we can see in the Figure 1.2 tectogrammatical representation is much more compact and contains less language dependent features in comparison with the respective analytic tree. We can also notice that the shapes of counterpart tectogrammatical trees are almost identical. There are of course some differences caused by the structural difference of the two languages such as word order and fertility (both these cases can be seen in the example above) but it was fairly justifiable decision to build the current transfer sequence on an the assumption that the tectogrammatical trees are identical. There are two primary components of the node which are considered during the transfer:

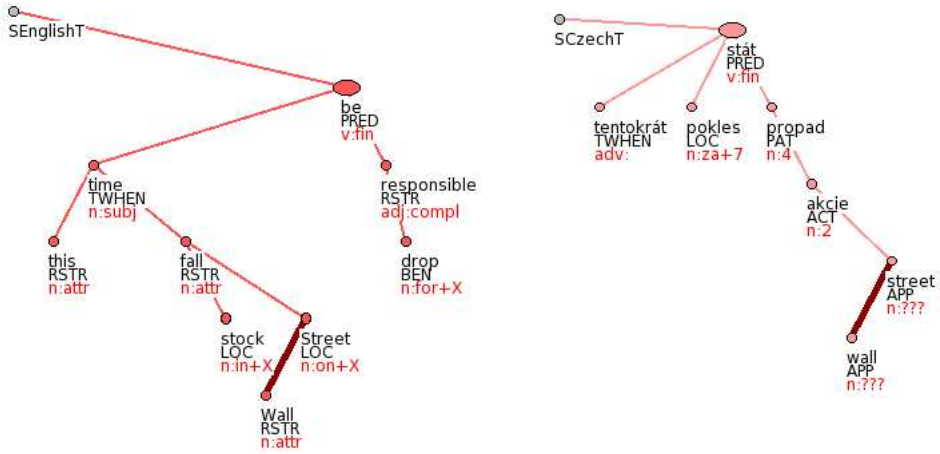
- lexeme represents the lexical content of the node and consists of tectogrammatical lemma and part of speech. Selection and weighting of lexemes is performed by the translation dictionary,
- formeme represents the intended (or former in the case of analysis) morphosyntactical form of the node. Selection is handled by the formeme translation model. (see (Žabokrtský et al., 2008) for further details).

[5]Prague Dependency Treebank

Figure 1.2: Counterpart analytical and tectogrammatical trees (simplified)



(a) a-trees



(b) t-trees

After a sharp drop in the morning, the Prague Stock Market corrected its losses.

Po ranním prudkém propadu pražská burza korigovala ztráty.

The baseline transfer sequence did the selection of the formemes and lexemes in an isolated manner. In a first step formemes were selected maximizing the score for the entire tree. Then the most probable lexeme compatible with the formeme was assigned to every node. A lexeme is considered to be compatible with a formeme if it has compatible part of speech. Aspect compatibility is also considered for the verbs (this is however not implied by the formeme itself). Formeme specifies *sempos* – semantic part of speech with restricted set of possible values (only *adv, adj, n, v* are allowed). Every sempos restricts compatible parts of speech and possibly even the *subpos* [6] – for example *adv* sempos allows adverbs and also ordinary numerals, indefinite pronouns and some others.

More advanced transfer algorithm using Hidden Markov Tree Model was also developed lately (Žabokrtský and Popel, 2009). This method considered combinations of lexemes and formemes and maximized the overall score (similarly to the previous formeme selection algorithm) using weights that reflect both the formeme and lexeme probabilities. Backward lexeme probabilities were also considered.

We can break down the transfer sequence into the following few steps:

1. source tectogrammatical tree is cloned,
2. formeme variants are selected for every node,
3. lexeme variants are selected for every node,
4. most probable combination of lexeme and formeme is selected,
5. rule-based post processing is applied to the target t-tree

The selection of lexemes and formemes is either done by the baseline sequence (formemes are selected by tree Viterbi, then the most probable compatible lexemes are assigned to every node) or by the HMTM (formeme and lexeme scores are merged together and tree Viterbi is executed on the combinations)

During the post-processing the shape of the resulting tree could be slightly altered using rule-based fixes that deal with structural differences between languages. For example genitives are moved to post position after their governing nodes when translating from English to Czech (therefore *awareness processes* is correctly translated as *procesy podvědomí* and not as *podvědomí procesy*).

1.4 TectoMT dictionary

The dictionary component within the TectoMT transfer block is a probabilistic model providing a list of translations for given source lemma and part of speech.

[6] This attribute of the Czech morphological tag defines other relevant morphological categories. For further details about the possible values see `http://ucnk.ff.cuni.cz/bonito/znacky.php`

Wide range of translations with various part of speech is usually returned and a suitable candidate is selected according to the local context. In the current simple approach just the most probable translation that is compatible with the formeme assigned to the given node is selected. However the dictionary itself does not prevent the implementation of more advanced method such as selecting the most probable combination of both formeme and compatible translation or further post-processing by tree language models. The dictionary should be designed so that it could be used in different transfer scenarios. Therefore it should be able to provide the best translation for given context (including constraints imposed by formeme) and should not rely on any assumptions about its neighboring blocks.

The following assumptions were considered to be key features of the desired dictionary:

1. locality – translation of a node should not depend on the translation of other nodes unless it is necessary to do so

2. richness – the dictionary should provide many translations for different cases (verbs can sometimes be translated as nouns).

3. good ordering – ordering of the possible translations is of utmost importance. Actual probability distribution is not that important as long as the best translations are on top of the list.

Because the dictionary will be used primarily in TectoMT system which is based on one-to-one correspondence of tectogrammatical trees, only single token translations are considered. It could be necessary to break this rule in specific cases (such as in the case of compounds or some specific words), but these are quite rare in the standard text so we could fix it by specific extensions.

1.5 Structure of the thesis

In Chapter 2 mathematical foundations, especially Maximum-Likelihood models that constitute the basis for our translation model, are described.. The translation dictionary is compared to the tightly related word-to-word alignment task and the former TectoMT dictionary is explained in a greater detail. In Chapter 3 various data sources used within this thesis are described. In Chapter 4 we describe the data preprocessing and preliminary analysis that had been done before the translation dictionaries were built. Chapter 5 is the primary part of this thesis and describes the construction of the translation dictionary and various aspects of the training procedure. Chapter 6 studies extra components that should eliminate most common errors not covered by the translation dictionary itself. Finally, there is Chapter 7 where we propose some simplified metrics, evaluate the final

dictionary using various methods. The implementation details and design decisions are discussed in Chapter 8.

Chapter 2

Mathematical foundations and the current state

2.1 Dictionary and alignment

The inner structure and the content of the dictionary depends heavily on the type of translation system it is going to be used with. In phrase-based translation system the dictionary usually covers not just the counterpart words but also more complex phrases, even the nonlinguistic ones. However, this has little impact on how the data can be acquired from the parallel corpus.

The problem of training the translation dictionary is very closely related to the problem of alignment. In this paper we are training the Maximum-Likelihood estimates $P(translation|source)$ using the aligned pairs. On the other hand the translation model could be used to select alignment A^* from the space of all possible alignments $\mathfrak{A}$ such that:

$$A^* = \underset{A \in \mathfrak{A}}{\operatorname{argmax}} \prod_{(s,t) \in A} p(t|s) \tag{2.1}$$

However many alignment methods can work without external dictionary nor do they require any prior knowledge of the language itself. These algorithms are typically based on iterative improvement of the alignments. They begin with roughly aligned texts (such as corresponding documents or chapters) and create small kernel dictionary consisting of words with high co-occurrence probability (using Maximum-Likelihood estimates). They use this information to create more fine-grained alignment which is then used to improve the dictionary and the process is repeated until a final stage is reached.

2.2 Probabilistic dictionary

Training data is a collection of source and target tectogrammatical trees whose corresponding nodes are pairwise aligned. In the basic sense, every node has its unique context and unique translation. If we are to build some useful translation model, we must make some independence assumptions and condition the translation probabilities using a reduced set of source attributes (such as source word lemma, part of speech, tag or possibly others) – we will call these attributes the *conditioning attributes*. We will use the term *conditioning strings* to denote the actual values of conditioning attributes (e.g. for conditioning attributes *lemma, part of speech* the conditioning strings might be `work#N`, `work#V`, ...). For certain model we assume that the translation probability depends only on the conditioning strings. If we select some arbitrary set of conditioning attributes and outcome (the set of target node attributes we are intersted in), we can infer *model* from the training set by simply counting how many times a given pair of conditioning string and outcome has been seen in the training data. This leads to the following definition:

Definition 1 *We call $\mathcal{M} = (S, T, c_{\mathcal{M}})$ a model where S is the set of conditioning strings, T is set of outcomes and $c_{\mathcal{M}} : S \times T \mapsto \mathbb{N}$ is a function that assigns count to every possible pair.*

Even the training corpus itself can be represented as a model $\mathcal{T} = (S, T, c_{\mathcal{T}})$ where S is set of source nodes, T is set of target nodes and for the count we put:

$$c_{\mathcal{T}}(s,t) = \begin{cases} 1 & \text{if s and t are aligned} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Such model represents the training set ideally but is incapable of dealing with any previously unseen data and is therefore completely useless. We need to reduce the complexity of the event space by grouping together observations that express similar behavior. The model reduction serves exactly this purpose – the detailed event space is collapsed into equivalence classes (this operation is possible for both the conditioning strings and outcomes but usually the context doesn't change at all) whose counts are summed together.

Definition 2 *The model $\mathcal{N} = (A', B', c_{\mathcal{N}})$ is a reduction of a model $\mathcal{M} = (A, B, c_{\mathcal{M}})$ (we write $\mathcal{N} \preceq \mathcal{M}$) if there exists injections $\phi : A \mapsto A'$ and $\psi : B \mapsto B'$ (this mapping is however usually identity) such that:*

$$c_{\mathcal{N}}(x,y) = \sum_{\substack{a \in A \\ \phi(a)=x}} \sum_{\substack{b \in B \\ \psi(b)=y}} c_{\mathcal{M}}(a,b) \tag{2.3}$$

In other words the reduction groups observations into equivalence classes. Counts for the equivalence classes are computed simply by summing counts of their members.

It is easy to show that if $\mathcal{L} \preceq \mathcal{M} \preceq \mathcal{N}$ then $\mathcal{L} \preceq \mathcal{N}$. Therefore the relation is transitive. It is in fact the ordering (nonlinear) on the space of all possible models for a given training set. This feature is very useful because it is much more convenient to compute some reduced model from some higher-order model than to compute it directly from a training corpus because we could deal with much less data and the whole operation would be much faster.

For every model $\mathcal{M} = (A, B, c_{\mathcal{M}})$ there exists conditional probabilistic model on the space of events $A \times B$ that uses Maximum-Likelihood estimates:

$$P_{\mathcal{M}}(a|b) = \frac{c_{\mathcal{M}}(a,b)}{\sum_{x \in A} c_{\mathcal{M}}(x,b)} \tag{2.4}$$

$$P_{\mathcal{M}}(b|a) = \frac{c_{\mathcal{M}}(a,b)}{\sum_{y \in B} c_{\mathcal{M}}(a,y)} \tag{2.5}$$

Maximum-Likelihood estimates provide quite simple and powerful tools for the language modelling. But we have to keep in mind that the reliability depends on the values of $c_{\mathcal{M}}$ heavily – for the lower counts the estimates tend to overfit the training data. The more complex event space $A \times B$ is used the more training data is needed in order to train reliable model. This is caused by the sparse data problem – for large event space the probability that certain pair (x, y) was not observed in the training data is higher – then the Maximum-Likelihood estimate $P(y|x) = 0$ even if the alignment is valid (so it should have nonzero probability).

Although more complex models tend to suffer heavily from the sparse data problem they could still provide extra discriminative power and we might want to use them. Sparse data problem (including unreliable low counts estimates) can be avoided using various techniques such as backoff or smoothing by lower order models.

In general if we have models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$ (where usually $\mathcal{M}_i \preceq \mathcal{M}_{i+1}$ but it is not necessary) we can define linear interpolation model P_Λ with vector $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ where $\sum_{i=1}^{n} \lambda_i = 1$ by settings:

$$P_\Lambda(t|s) = \sum_{i=1}^{n} \lambda_i P_{\mathcal{M}_i}(t|s) \tag{2.6}$$

Optimal interpolation weights can be found using EM Smoothing algorithm

Backoff strategy does not combine the distributions directly. Instead the model suitable for certain conditioning string is chosen by some decision function $H(s)$

such as in the following example where $\mathcal{N} \preceq \mathcal{M}$:

$$P(t|s) = \begin{cases} P_{\mathcal{M}}(t|s) & \text{if } H(s) \\ P_{\mathcal{N}}(t|s) & \text{if not } H(s) \end{cases} \tag{2.7}$$

The decision function $H(s)$ identifies conditioning strings s where model $\mathcal{M}$ is reliable and leaves the rest to the lower order models. Although we have shown a simple example with only two models here, generalizations with longer chains and various decision functions for each level can be implemented. For the above example the following decision function H_α is commonly used:

$$H_\alpha(s) \Leftrightarrow \sum_x c_{\mathcal{M}}(s, x) \geq \alpha \tag{2.8}$$

2.3 Former TectoMT dictionary

Prior to the completion of this work TectoMT system used the dictionary based on the work of Jan Cuřín (Cuřín, 2006) that was embedded into a simple block which selected translations so as to maximize translation probability for every node while not breaking the compatibility with assigned formemes. Zdeněk Žabokrtský has further extended the dictionary with hand-written derivational procedures and various rule-based fallbacks.

The former dictionary was constructed in a slightly different way – whereas we are aiming at exploiting automatically aligned parallel corpora (though this step usually involves some dictionary that could be partially hand-made) the former dictionary was based on various human dictionaries that were combined with dictionary extracted from EuroWordNet. Aligned parallel corpus was used to train probabilistic model.

The dictionary construction consisted of the following steps:

- **Merging and pruning** – combine and prune human dictionaries (WinGED, GNU/FDL, Euro WordNet) using monolingual frequencies (North American News Text),
- **POS tagging** – of the dictionary data on both English and Czech side,
- **GIZA++** – dictionaries were combined with data extracted from aligned parallel corpora (Penn TreeBank corpus)

Pruning and selection of translation pairs of the Czech-English translation dictionary was based on the following assumptions:

1. no filtering on the entry side is needed,

2. if only single translation for certain token is found it is kept in the dictionary,
3. translations not found in the monolingual English corpora were discarded,
4. translations occurring in all input manual dictionaries were considered to be better than those occurring in only some,
5. different weights were assigned to input manual dictionaries (e.g. if both source and its translation are in the same synset of Euro WordNet it is more important that if the pair occurs in regular dictionary),
6. POS tag assignment of the target tokens can be disambiguated by the POS tag of the source. This is especially true for English-Czech language pair.

There are several differences between the former approach and the new one proposed in this thesis. The former work used smaller parallel corpora and relied more on data from various manual dictionaries. We are primarily focused on exploiting the parallel corpus. The training parallel corpus was aligned using GIZA++ at analytical level in the former work while we are using more advanced alignment methods at the tectogrammatical layer. Tectogrammatical alignments are more sparse but also more reliable – especially due to the 1:1 nature of alignments and also because of the omission of various auxiliary words.

Chapter 3

Data sources

We decided to extract the translation dictionary automatically from an aligned tectogrammatical treebank. We found a treebank with the required level of annotation – PCEDT (Čmejrek et al., 2004) consisting of 21.600 sentences. This treebank was however too small for our purposes. Another available corpus CzEng 0.7 (Bojar and Žabokrtský, 2006) was of reasonable size as it contained about 1.3M aligned sentences but lacked the required level of annotation and alignment. However the TectoMT framework allowed us to perform the annotation and alignment of the data in a quite convenient way. Therefore we decided to use the CzEng corpus as our primary data source. Monolingual corpora for both languages were also used for pruning and final model smoothing. British National Corpus (Consortium, 2007) for English and SYN2005 variant of the Czech National Corpus (cnk, 2005) for Czech were used.

3.1 CzEng

Our primary data source was Czech-English parallel corpus CzEng (Bojar and Žabokrtský, 2006). This corpus is being developed by the Institute of Formal and Applied Linguistics (ÚFAL), Charles University, Prague. We have worked with the CzEng 0.7 which was the most recent version at the time of writing this thesis. This version has been developed in the years 2005-2006. CzEng consists of large set of parallel texts from a wide range of topics. Texts included are limited to those that were already available in electronic forms and were not protected by author's rights in the Czech Republic.[1]

Texts in CzEng were collected from the following sources:

[1] CzEng is publicly available for educational and research purposes, but users should read license agreement first. For more information see: `http://ufal.mff.cuni.cz/czeng`

- celex – Acquis Communautaire Parallel Corpus, EU legislative texts written between 1950s and 2005,
- euconst – EU constitution proposal published in Corpus OPUS,
- navajo_user_translations – Anonymous user translations for the Navajo project,
- GNOME projects localization files,
- KDE localization files,
- Articles from Project Syndicate,
- eujournal – Random samples of Official Journal of the European Union,
- Reader's Digest stories,
- kacenka – Subset of parallel corpus Kačenka,
- books – electronic books freely available on the internet both in Czech and English.

CzEng 0.7 contains 20,967,030 English and 23,415,945 Czech words (including punctuation) in 1,375,908 English and 1,383,203 Czech sentences in 13,793 document pairs. Texts are tokenized and sentence-aligned using freely-available `hunalign` tool (Varga et al., 2005).

CzEng contains vast amount of aligned parallel documents but it lacks the level of annotation we require for the dictionary extraction. Parallel texts were therefore converted into TectoMT format where it was easy to analyze both English source and Czech source sentences up to tectogrammatical layer and to align counterpart tectogrammatical trees. Alignment of tectogrammatical trees was done by the method described in (Mareček et al., 2008). CzEng 1.0 which is currently in development and is not yet released is aligned and annotated using the methods similar to those described in the paper.

3.2 Other corpora

Other corpora were used primarily to aid pruning and for smoothing of the final model. British National Corpus (Consortium, 2007) contains 100M words from a wide variety of sources and was used to build frequency table of English lemmas (with distinguished part of speech in order to reduce lemma homonymy). The most advanced corpora for Czech were developed by the joint effort of UCNK and UFAL (cnk, 2005). We have decided to use the following Czech corpora:

- SYN2006PUB is a collection of about 300M words from press texts from years,
- SYN2005 is smaller corpus consisting of about 100M words which aims to be balanced.

3.3 Manual dictionaries

Automatic dictionaries could be improved by the integration of classical non-probabilistic dictionaries. These are not directly suitable for the translation but they could be used as another feature integrated into the training process. Translations that are included in variety of reliable classical dictionaries should receive some extra points. While there is no probabilistic model associated with classical dictionaries we will use them as boolean features. These dictionaries also have only very limited part of speech disambiguation if any (they usually use it to distinguish roles of homophonic words). Because such distinction is crucial for our task we need to annotate the data prior to use. The process of part of speech annotation of manual dictionaries will be described in the following chapter.

Non-probabilistic translation dictionaries that are publicly available include:

- GNU/FDL English-Czech dictionary consists of about 3M of text. Entries can be annotated with part of speech and possibly domain (for specific terminology).

Chapter 4

Preparatory work

4.1 Penn tagset conversion

English morphological analysis in the training data as well as in the translation system uses Penn treebank part-of-speech classification system[1]. The former TectoMT dictionary used parts of speech to distinguish between different English words using simple reduction $r : T_s \mapsto POS_s$ where T_s denotes the set of source tags and POS_s the set of source parts of speech. This conversion significantly reduced the number of classes and thus led to more compact and robust model. We believe that using full tags for the translation conditioning could improve discriminative power of the translation model and we are therefore going to examine the behavior of tags and their relations with part of speech in order to:

1. see where the former reduction could be problematic,
2. examine behavior of tags
3. and design reduction algorithm that better reflects the properties of data.

4.1.1 Tag to part of speech conversion

Tags could be divided into groups with common part of speech. Due to the nature of strings assigned to tags in Penn treebank it is easy to implement the conversion by a simple pattern matching table (see Table 4.1).

4.1.2 Tag correspondence model

We assume that the part of speech should remain unchanged during the translation. This is not always true, because some language structures are expressed in

[1]For further details see `http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html`

Table 4.1: Former tag to PoS conversion table

Tag pattern	Assigned POS
`^JJ`	`A`
`^W`	`P`
`^DT`	`P`
`^R`	`D`
`: ^(.)`	`$1`

a different way in the various languages. However we could assume that it should hold in most cases. Therefore we decided to have a look at the correspondence between source tags T_s and target part of speech P_t classes. Three correspondence models have been trained from different data sets using Maximum-Likelihood estimates (see Table 4.2) in order to see how stable the model is with respect to the choice of training data. For the comparison of the distributions there are some well established methods such as Kullback-Leibler divergence (Kullback and Leibler, 1951):

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4.1}$$

Kullback-Leibler divergence is not metric because it doesn't satisfy the triangle inequality and it is not symmetric. It expresses how much information (measured in bits) we loose if we represent P using probability distribution Q. Therefore the following holds: $D_{KL}(P\|Q) = 0 \Leftrightarrow P = Q$. We have decided to use Jensen-Shannon divergence (Wong and You, 1985) which is based on Kullback-Leibler and is symmetric:

$$D_{JS}(P\|Q) = \frac{1}{2}\left(D_{KL}(P\|M) + D_{KL}(Q\|M)\right) \tag{4.2}$$

$$M = \frac{(P+Q)}{2} \tag{4.3}$$

We have measured Jensen-Shannon divergence for the three samples and the results in Table 4.3 show that the model is relatively stable with respect to the choice of training data. Three most probable parts of speech for every tag (extracted from model M^+) and the former part of speech assignments are shown in Table 4.7. Tags where the most probable part of speech differs from the former assignment are marked with an asterisk. We can see that the former approach was valid for major tag classes although there are certain cases where even the most frequent part of speech has a relatively low probability. This occurs for `VBG` tag

(and also of JJR, JJS). We expect that the relatively high probability of $P(\texttt{N}|\texttt{VBG})$ is due to the structural differences of English and Czech where gerund verbs (tagged with VBG) are quite often expressed by nouns in Czech. The following example illustrates this phenomenon:

```
This will lead to increasing#VBG productivity in Japan.
To povede ke zvýšení#N produktivity v Japonsku.
```

Table 4.2: Tag to part of speech models

Model	Source	$\sum_{x,y} c(x,y)$
M_1	Train sample, 10 files	12063
M_2	Heldout sample, 10 files	12099
M_3	Test sample, 10 files	12380
M^+	Merged models M1, M2, M3	36542

Table 4.3: Jensen-Shannon divergence between tag to part of speech models

x, y	M_1	M_2	M_3	M^+
M_1	0	0.0070	0.0068	0.0033
M_2	0.0070	0	0.0071	0.0033
M_3	0.0068	0.0071	0	0.0030
M^+	0.0033	0.0033	0.0030	0

Table 4.4: Tag clusters with threshold $\lambda = 0.05$

$\|C\|$	$D(C)$	Members
4	0.0280	VB VBD VBP VBZ
3	0.0457	FW NN NNP
3	0.0487	WDT WP WP$
3	0.0322	$ NNPS NNS
2	0.0461	DT PDT
2	0.0495	RB RBR
2	0.0368	JJS RBS
Not-clustered		, -LSB- : CC CD JJ JJR LS MD UH VBG VBN WRB

Table 4.5: Tag clusters with threshold $\lambda = 0.1$

$\|C\|$	$D(C)$	Members
7	0.0845	MD VB VBD VBG VBN VBP VBZ
6	0.0803	$ FW NN NNP NNPS NNS
3	0.0487	WDT WP WP$
2	0.0461	DT PDT
2	0.0495	RB RBR
2	0.0985	CD LS
2	0.0368	JJS RBS
Not-clustered		, -LSB- : CC JJ JJR UH WRB

Table 4.6: Tag clusters with threshold $\lambda = 0.2$

$\|C\|$	$D(C)$	Members
7	0.0845	MD VB VBD VBG VBN VBP VBZ
6	0.0803	$ FW NN NNP NNPS NNS
5	0.1659	DT PDT WDT WP WP$
4	0.1526	JJ JJR JJS RBS
3	0.1117	RB RBR WRB
2	0.0985	CD LS
Not-clustered		, -LSB- : CC UH

4.1.3 Clustering

We can use Jensen-Shannon divergence to compare conditional distributions of tags in the correspondence model M^+. We can break the model down to probabilities induced by tags by putting $P_T(x) := P(x|T)$ for each $T \in T_s$ and $x \in POS_t$. Then we can measure the divergence of tags T_1, T_2 by Jensen-Shannon divergence of their respective probability mass functions:

$$D_{JS}(P_{T_1} \| P_{T_2})$$

We can then identify sets of similar tags by constructing partitioning $\mathcal{P} = \{C_i\}_{i=1}^n$ of the tagset $\mathcal{T}$ with the following properties:

1. $C_i \subseteq \mathcal{T}$
2. $\bigcup_{C \in \mathcal{P}} C = \mathcal{T}$
3. $\forall 1 \leq i < j \leq n : C_i \cap C_j = \emptyset$

We are interested in finding a partitioning where clusters contain tags that are similar – whose Jensen-Shannon divergence is small enough. We will use the following definition of diameter:

Definition 3 *Jensen-Shannon diameter (or diameter) $D(C_i)$ of a cluster C_i is the maximal divergence between its members:*

$$D(C_i) = \max_{x,y \in C_i} D_{JS}(P_x \| P_y) \tag{4.4}$$

We say that the partitioning $\mathcal{P} = \{C_i\}_{i=1}^n$ has a threshold α if $\forall C_i \in \mathcal{P} : D(C_i) \leq \alpha$.

Algorithm

We will carry out the clustering with the algorithm described in 4.1. The actual implementation finds the minimum by searching the list of clusters. Therefore it runs in $O(n^2)$ time. Heap-based implementation will carry out the operation in constant time. However the problem space is rather small so the sub-optimal implementation is not a big problem.

4.1.4 Results

We have constructed the tag partitionings with various threshold levels. Three possible clusterings are shown in the Tables 4.4, 4.5 and 4.6. We can see that tags for the same part of speech tend to end up in the same cluster. Interesting observation is that behavior of verbs with `VBG` and `VBN` tags tend to be slightly divergent from the other verb types.

4.2 Annotation of manual dictionaries

For many translation pairs extracted from manual dictionaries no part of speech classification was known. We decided to extract only one-to-one translation pairs from the dictionaries. For every translation pair (L_s, L_t) we have first generated set of possible part of speech classes C_s, C_t for each side using morphological analyzers[2]

We have then used the model M^+ from the previous section in order to pick the classes $(c, d) \in C_s \times C_t$ maximizing the following:

$$(c, d) = \operatorname*{argmax}_{(x,y) \in C_s \times C_t} P(y|x)P(x|L_s) \tag{4.5}$$

[2]Analyzers based on **Morce** have been used for both sides. These are available as perl modules `Morce::English` and `Morce::Czech`. For more info see `http://ufal.mff.cuni.cz/morce/index.php`

Figure 4.1: The clustering algorithm

Let $T = \{t_1, t_2, \ldots, t_n\}$
Let $\mathcal{P} = \{C_1, C_2, \ldots, C_n\}$ where $C_i = \{t_i\}$
Let $D(i, j) = \rho(C_i, C_j)$
Build Min-Heap H from (i, j) where $1 \leq i < j \leq n$ using $D(i, j)$ as keys
while H nonempty **do**
  $(i, j) \leftarrow$ Extract-Min(H)
  if $D(i, j) > \alpha$ **then**
    Exit
  else
    $C_i \leftarrow$ Merge(C_i, C_j) {Merge clusters}
    Delete C_i from $\mathcal{P}$
    Delete all edges $(x, y) \in H$ crossing i, j: $\{x, y\} \cap \{i, j\}$
    Remove edges crossing i, j from H
    for $C_x \in \mathcal{P}$ **do**
      Skip C_x if $x = i$
      $D(x, i) = \max_{y \in \{i,j\}} D(x, y)$ {Divergence between C_x and new cluster is maximum from the two}
      Insert (x, i) into H {Add edges for new cluster}
    end for
  end if
end while

Where $P(x|L_s)$ is calculated from the BNC corpus and $P(y|x)$ is supplied by the correspondence model M^+.

Table 4.7: Tag to part of speech correspondence

*	T_s	$P(T_s)$	$r(T_s)$	a	$P(a\|T_s)$	b	$P(b\|T_s)$	c	$P(c\|T_s)$
	NN	0.227	N	N	0.829	A	0.100	V	0.026
	JJ	0.156	A	A	0.827	N	0.072	D	0.037
	NNS	0.118	N	N	0.937	A	0.033	V	0.017
	NNP	0.109	N	N	0.752	A	0.230	D	0.005
*	CC	0.057	C	J	0.957	N	0.014	D	0.014
	RB	0.056	D	D	0.740	A	0.065	N	0.056
	VB	0.054	V	V	0.821	N	0.142	A	0.014
	VBN	0.035	V	V	0.762	A	0.149	N	0.067
	VBZ	0.032	V	V	0.932	N	0.036	D	0.011
	VBG	0.026	V	V	0.492	N	0.302	A	0.168
	VBD	0.024	V	V	0.900	N	0.047	A	0.032
	DT	0.023	P	P	0.708	A	0.096	D	0.073
	VBP	0.019	V	V	0.910	N	0.046	D	0.019
	CD	0.018	C	C	0.759	N	0.188	A	0.020
	WDT	0.008	P	P	0.964	D	0.026	V	0.007
	JJR	0.007	A	A	0.583	D	0.344	V	0.035
	WP	0.005	P	P	0.912	D	0.041	A	0.021
	NNPS	0.005	N	N	0.948	A	0.034	V	0.017
*	:	0.005	:	Z	0.602	J	0.105	P	0.082
*	WRB	0.005	P	D	0.762	P	0.155	N	0.036
	JJS	0.003	A	A	0.529	N	0.303	D	0.168
	RBR	0.002	D	D	0.840	A	0.086	V	0.037
	PDT	0.002	P	P	0.636	D	0.145	A	0.073
*	MD	0.001	M	V	0.808	N	0.115	J	0.038
*	FW	0.001	F	N	0.737	A	0.211	D	0.053
*	UH	0.001	U	N	0.421	T	0.368	D	0.105
*	$	0.000	$	N	1.000	–	–	–	–
	WP$	0.000	P	P	0.933	N	0.067	–	–
*	RBS	0.000	D	A	0.333	N	0.333	D	0.333
*	,	0.000	,	A	0.500	V	0.500	–	–
*	LS	0.000	L	N	0.500	C	0.500	–	–
*	-LSB-	0.000	-	Z	1.000	–	–	–	–

Chapter 5

Building the dictionary

In this chapter we describe the construction of probabilistic translation dictionary from the training data. Simply said the translation dictionary is transformation that maps conditioning strings[1] into some ordered set of possible outcomes. While the outcomes are quite well defined for the TectoMT dictionary (only lemma and part of speech is required to disambiguate lemma homonymy) conditioning attributes (defining the conditioning strings) is what really matters, because it is the conditioning attributes what can possibly discriminate various situations where different translations should be used. In the course of the following chapter we will:

1. discuss which conditioning attributes might contribute to the discriminative power,
2. induce models from the training data and finally,
3. build complex translation dictionary from the simpler models.

We also discuss further issues regarding the filtering of the noisy training data and training of the model parameters. Assume that we have some conditioning string s and possible translations $t_1, t_2, \ldots t_n$. Probabilistic translation model $\mathcal{M}$ is described by a probability mass function $P_{\mathcal{M}}$ such that for every conditioning attributes s:

$$\sum_{i=1}^{n} P_{\mathcal{M}}(t_i|s) = 1$$

Within the machine translation the dictionary is responsible for:

[1] Defined in Section 2.2

1. the construction of the set of possible outcomes $Tr(s)$ (some translations might be later discarded due to the structural constrains of the actual conditioning attributes which doesn't necessary be modelled by the dictionary itself),

2. the choice of the *optimal translation*:

$$\hat{t} = \underset{t \in Tr(s)}{\operatorname{argmax}} P_{\mathcal{M}}(t|s) \tag{5.1}$$

Because some further selections might occur (due to some external constrains) we are basically interested not only in *the best translation* but the ordering of translations defined by the $P_{\mathcal{M}}$ distribution.

5.1 Selecting conditioning attributes

The independence assumptions are based on the concept of *locality*. We believe that the probability of the translation depends on a very limited context around the source node. We can safely discard all sentence crossing dependencies from the consideration because the cases where the translation really depends on wide discourse are very rare and the training corpus (although quite huge) is still too small to learn such complex relations. Moreover we can achieve very promising results even if we consider the source node alone. This very limited context is actually used in the human dictionaries and it works quite well (more detailed context is required to show collocations or idiomatic uses). If we have English word *"work"* we assume that it could be translated into Czech as *"pracovat"* or *"práce"* depending on the part of speech.

We are now going to discuss which language features could be included into the conditioning attributes. Our main interest is to select features that can discriminate between cases where different translation should be used. We must also note that the actual choice of translation is driven by the *formeme* (for further details about the formemes and the evaluation of compatibility see Section 1.3).

5.2 Models and decisions

By analyzing the nature of texts and errors made by the former translation system we have come to the following conclusions:

- part of speech affects meaning – compare *work#N* → *práce#N* and *work#V* → *pracovat#V* ,

- source tag can affect meaning – information encoded in tags is richer than the POS classification itself and in some special cases it can help determining correct translation. The following examples illustrate this phenomenon:

  number was found to carry lexical information: *wood#NN* → *dřevo#N* , but *wood#NNS* → *les#N* ,

  translation of verbs can be different depending on their *mode* – for example `VBG` verbs were found to be translated to Czech as nouns quite often.

- lexical context can affect the translation, especially in case of collocations (compare *cell* → *buňka* with *cell phone* → *mobilní telefon*)

- some words are irregular with respect to negation and the negation itself is not consistently represented,

- target language inflexion shouldn't be considered by the dictionary as it is implied by the formeme

We decided not to consider lexical context within the dictionary because this problem should be addressed by the tree language model that is currently being developed for the TectoMT system. In addition to the basic attributes such as lemma and tag (or reduction such as part of speech) we decided to extract more attributes of the tectogrammatical nodes that could provide extra discriminative power.

According to the above observations we have decided to build these basic models:

1. `POS` model conditions the probability of translation by source part of speech POS_s and lemma L_s. This model is similar to the model that was used in the former dictionary (Cuřín, 2006),

2. `Tag` model is more detailed and uses full source tag T_s and lemma L_s for conditioning. This model should provide additional discriminative power. `POS` model could be used as a backoff.

5.2.1 Negation

There are two primary phenomena that cause problems with the correct resolution of negation:

- some lexical units are irregular with respect to the negation – their possible translations are different when they are in affirmative and in negative form (for example *nearby* → *nedaleko* but *far* → *daleko*)

- negation imposed by various prefixes is usually represented by grammatemes on tectogrammatical layer while the actual prefix is often removed from the tectogrammatical lemma.

Both phenomena together cause that various negative words are inadvertently made affirmative and recorded within the dictionary with their intended translations. In the end the dictionary won't recognize one case from another and could supply translations with the completely opposite meaning.

5.3 Data extraction

Aligned pairs can be extracted from the training files using a simple printing block `Print::English_Czech_aligned_pairs` that finds alignments and computes values of various attributes we could include in the intended models. We decided to extract all required attributes in a single run because TMT format is quite complex and it takes very long time to process the entire training data. Dealing with the obtained aligned pairs is much more convenient because there is much less amount of data involved. Fully annotated and aligned CzEng takes 5.1 GB of gziped complex data (not including the 4.1 GB for the subtitles section) whereas the extracted aligned pairs take only 479M (and only 76MB if compressed with gzip) of easy-to-process plaintext. The difference is obvious.

For every aligned pair the following attributes are recorded:

- tag and lemma for source node,
- tag and lemma for the head of source node,
- functors of source and head nodes,
- semantic part of speech of source node,
- lemma and part of speech of target node,
- negation grammatemes of source and target nodes,
- source document,
- capitalization.

If we keep the capitalization information (which is usually not present in the lemma itself) we will be able to reconstruct the correct case of the translation. Capitalization of the source is usually accompanied with appropriate change in tag (compare $bill\#NN \rightarrow účet$ with $Bill\#NNP \rightarrow Bill$) which is another feature we can exploit in order to distinguish these cases.

Source document is tracked because we would like to see how various parts of the corpus contributed to the result. Moreover we used this information to obtain more balanced dictionary by penalizing topic specific translations that could prevail in some large but nonstandard source. Most probable translation of `office` within documents from the Microsoft set is obviously `office` which is certainly not what we would like to have in the topic independent dictionary.

The extraction procedure was executed on the Linguistic Reseach cluster with 160 CPUs at the Institute of Formal and Applied Linguistics. Training data were divided to equally sized chunks that were processed in parallel on the cluster nodes which dramatically reduced time needed to perform this action. Results were then merged into the single output file that was further filtered in order to build lower order models. Alignments were then sorted by count of occurrences and models were built from these lists using some pruning methods that are described in the following section.

5.4 Model pruning

Plain Maximum-Likelihood estimates are quite good basis for the probabilistic model training but due to the nature of the training data some pruning is inevitable. Due to the automatic annotation and alignment there is certain amount of incorrect alignments that doesn't improve the final translation model – it can even decrease its performance. Moreover we need to end up with a model that is of reasonable size so we need to set up some thresholds and prune the low-frequency pairs from the dictionary. There are various features that can be considered such as Maximum-Likelihood probabilities $P(t|s)$ and $P(s|t)$, unconditional probabilities of the conditioning attributes and outcome $P(s)$ and $P(t)$ (trained from the monolingual corpora), presence in the manual dictionaries or some other features that can be of more complex nature. Basically we can divide the pruning into:

1. *early rule-based pruning* that can be evaluated as soon as the model is built from the input files and relies on simple rules that can be evaluated prior to the knowledge of the final models,

2. *late filtering* can use more complex rules that are evaluated when the model is already constructed.

Although the power of early pruning is limited it has one great advantage. Because rules are applied while the model is incrementally built, eliminated pairs doesn't even get to the memory. Early pruning can be therefore applied even on training data that would not fit into the memory. Late filtering can be used to further reduce size of the final models in a smarter way but model needs to fit into the memory by then.

5.4.1 Rule based pruning

Estimating the conditional probability

We decided to build models from frequency lists of aligned pairs of conditioning attributes s and outcomes t_i (lemma and part of speech) sorted by the number of occurrences $c(s, t_i)$. This has allowed us to prune out pairs with $P(t|s)$ below certain threshold α even if the model is not yet fully known. Assume that for certain s there are possible outcomes $t_1, \ldots, t_n$ such that $c(s, t_i) \leq c(s, t_{i+1})$. At the time when we encounter the pair (s, t_k) we have already processed all the pairs (s, t_i) for $i < k$ because the input is sorted. If we put $C_j(s) = \sum_{i=1}^{j} c(s, t_i)$ we can define following upper bound P_k:

$$P_k(t_i|s) = \frac{c(s, t_i)}{C_k(s)} \tag{5.2}$$

For the Maximum-Likelihood model $\mathcal{M}$ we have:

$$P_{\mathcal{M}}(t_k|s) = \frac{c(s, t_k)}{C_n(s)} \leq \frac{c(s, t_k)}{C_k(s)} = P_k(t_k|s) \tag{5.3}$$

As soon as we encounter the pair (s, t_k) we can compute $\frac{C(s,t_k)}{C_k(s)}$ and if it lies below the threshold α so does the $P_{\mathcal{M}}(t_k|s)$ because of the inequality 5.3:

$$\frac{C(s, t_k)}{C_k(s)} < \alpha \Leftrightarrow C(s, t_k) < \alpha C_k(s) \Leftrightarrow (1 - \alpha)C(s, t_k) < \alpha C_{k-1}(s) \tag{5.4}$$

The above relation formulate the exact computation that can be used to test whether certain pair have hit the threshold.

If $P(t_{k-1}|s) \geq \alpha$ and $P(t_k|s) < \alpha$ then also $P(t_j|s) < \alpha$ for all $i \geq k$ due to the ordering. Therefore t_{k-1} is the last translation that was not pruned. For the pruned model $\mathcal{M}'$ we have:

$$P_{\mathcal{M}'}(t_{k-1}|s) = P_k(t_{k-1}|s) \geq \alpha \tag{5.5}$$

The inequality follows from the 5.3. This equation is the second part of the validity proof. Translations whose probability lies below the threshold are pruned due to the design of the pruning (equations 5.3 and 5.4) and the last kept translation t_{k-1} has probability above the threshold due to the equation 5.5.

Pruning rules

We have decided to remove the following pairs from further consideration:

- plain numbers `1256#C` → `1256#C`,

- single character tokens on either side `and` → `a`,
- technical nodes such as `#PersPron`,
- words that weren't seen in the monolingual corpora at least k times.

Omission of numbers expressed by digits is motivated by the fact that they are mostly kept unchanged. There is no need to translate numbers occurring in the text. Also there is practically infinite amount of number that can appear in the text though all of them usually behave in a same way. Single character tokens that have some meaning can be also addressed by small look-up tables. This approach was also used in the former translation block so we decided not to change this. There is no need to consider technical nodes by the dictionary either because these nodes are processed by other units of the translation system. There is a limited number of personal pronouns that can be used for certain node and other units of the translation systems can better decide which should be used in particular case. The last step is the most interesting one – we have decided do prune out pairs composed of tokens that are too rare to occur in monolingual corpora. These rare cases should include misspelled words, interjections, fancy proper nouns and other unusual cases. We expect that extra complexity of the model brought by these rare words is not worth the effort. Some of these words could be left intact without any harm (such as interjections and proper nouns), some could not but still their probability is too low to cause significant change of the model performance.

Pairs with low probabilities were then eliminated using following parametrized rules:

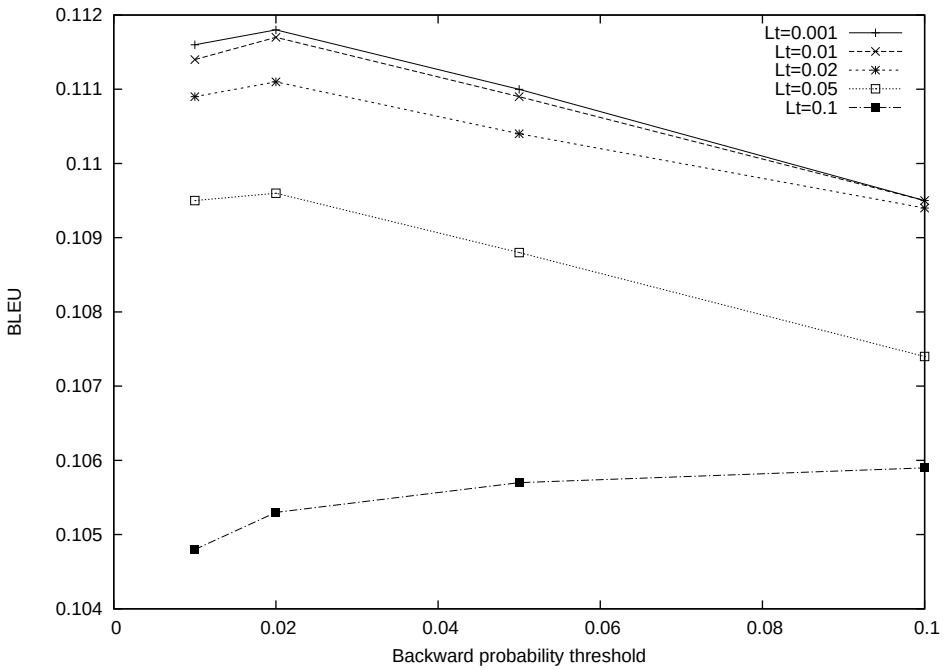
- $P(t|s) < \alpha_t$
- $P(s|t) < \alpha_s$
- $c(s,t) < \alpha_c$

It was shown that the first condition can be evaluated early (see equation 5.4). The second rule needs a model to be fully built because $P(t|s)$ can't be estimated. The third condition eliminating pairs according to their absolute count is very strong and is used only to further reduce the size of higher order models where we can expect that lower order models will take control.

We would like the lists of translations to be of reasonable size for each conditioning attributes. However if we don't further prune the dictionary we have guaranteed that if α_t threshold was applied to $P(t|s)$ there could be at most $\lfloor \frac{1}{\alpha_t} \rfloor$ translations for each s because:

$$1 = \sum_{i=1}^{n} P(t_i|s) \geq \sum_{i=1}^{n} \alpha_t = n\alpha_t \tag{5.6}$$

Figure 5.1: BLEU score with fixed λ_t and variable λ_s



Therefore $n \leq \frac{1}{\alpha_t}$.

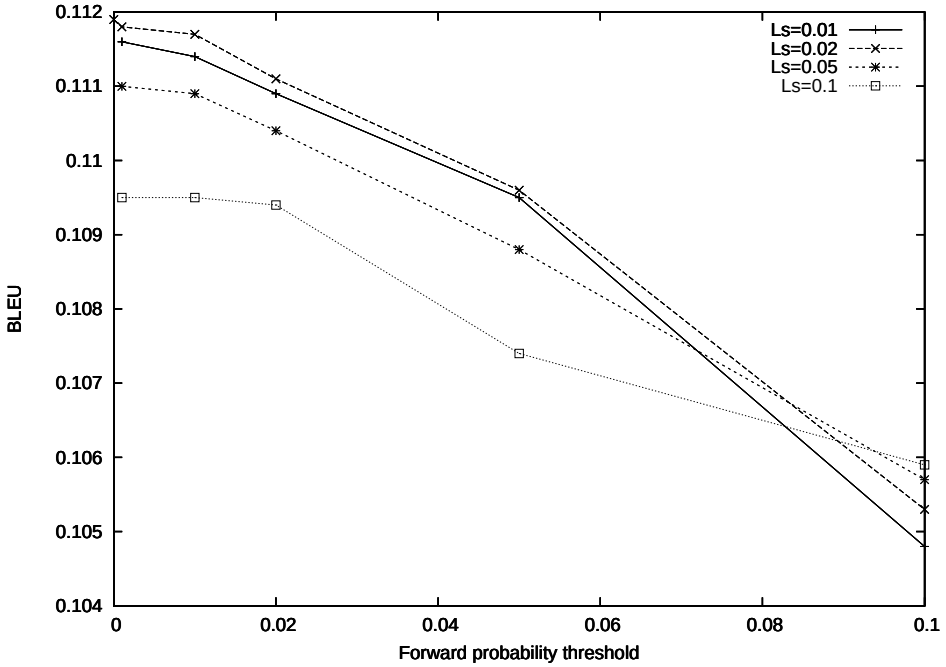
Estimating the thresholds

We have experimented with various pruning thresholds for the `POS` model and evaluated the performance of the model using TTM (see Section 7.1 for details) and BLEU/NIST metric. The results in Table 7.1 show that best results are achieved with $\alpha_t = 0.02$ and $\alpha_s \leq 0.001$ (BLEU and NIST metrics are not consistent for the cases where $\alpha_s = 0.001$ and $\alpha_s = 0$), we prefer to use nonzero pruning thresholds. Graphs displaying how BLEU changes with various threshold values are shown in Figures 5.1 and 5.2.

5.4.2 Hierarchical models

We have built models of various conditioning attributes and different complexity. In this section we describe methods how these models could be combined into single robust model that can combine good properties of its components – using specific information from higher order models whenever available and using lower order models as an backoff when sparse data problem emerges. The hierarchical model $\mathcal{H}$ is structure composed of:

Figure 5.2: BLEU score with fixed λ_s and variable λ_t



1. ordered list of *member* models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$
2. function providing list of possible translations $Tr_{\mathcal{H}} : S \mapsto 2^T$,
3. function computing conditional probability of translations $P_{\mathcal{H}}$

The list of possible translations can be constructed in various ways. We have considered the following approaches:

1. translations from the lowest order model $\mathcal{M}_1$ are considered,

$$Tr_{\mathcal{H}}(s) = Tr_{\mathcal{M}_1}(s)$$

2. translations provided by any of the model $\mathcal{M}_i$ are considered,

$$Tr_{\mathcal{H}}(s) = \bigcup_i Tr_{\mathcal{M}_i}(s)$$

Due to the simpler implementation and assumption that $\mathcal{M}_1$ would be always the last resort backoff model we have decided to use first approach.

Computation of the hierarchical probability function $P_{\mathcal{H}}$ could be done in various ways. Smoothing methods such as linear interpolation (see Equation 2.6)

or backoff (see equation 2.7) could be applied. We have however decided to use perceptron[2] tool to train ranking function on a set of features (mainly derived from the conditional probabilities of the member models) along with a concept of bucketing.

5.4.3 Bucketed smoothing

In NLP the bucketed smoothing is a well known approach that can improve performance of certain smoothing strategies. It is based on an observation that data could be sometimes separated into subsets that express similar behavior and for each such subset the different smoothing parameters could be selected. For example if we consider linear interpolation the task is to find certain $\Lambda = (\lambda_1, \ldots, \lambda_n)$ such that $P(t|s) = \sum_{i=1}^{n} \lambda_i P_i(t|s)$ maximizes certain metric. The bucketing assumes that there is a function $\psi : S \mapsto 0, 1, 2, ...k$ that splits the source into k sets. We must then train k vectors $\Lambda^j = (\lambda_1^j, \ldots, \lambda_n^j)$ and the probability is computed as:

$$P(t|s) = \sum_{i=1}^{n} \lambda_i^{\psi(s)} P_i(t|s) \tag{5.7}$$

We have to consider that as the number of buckets grows the less training data belongs to each bucket so we need to be careful about the sparse data problem. On the other hand this approach offers additional level of freedom and the final model could model the data better.

5.4.4 Perceptron

Perceptron is the ranking algorithm based on the neural network paradigm. For certain set of features $f_1, \ldots, f_n$ the score for given event x is computed as:

$$w(x) = \sum_{i=1}^{n} \lambda_i f_i(x) \tag{5.8}$$

It supports both real valued and categorial features (where categorial feature $g : X \mapsto C$ are converted to set of boolean features $g_c(x) \Leftrightarrow g(x) = c$. In a same way how categorial features are handled by the perceptron the bucketing can be implemented. For the feature f and bucketing function $\psi : X \mapsto 1, 2, \ldots, k$ we can use the following set of features instead:

$$f_i(x) = \begin{cases} f(x) & \text{if } \psi(x) = i \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

[2]Implementation available as perl module `ML::Reranker`

Moreover we can use various orthogonal bucketing functions at the same time and let the perceptron decide.

The main problem is that the reranker score doesn't define probability distribution. The range of feature is not bounded anyway. We can still transform the score into probability distribution with the following equation:

$$P(t|s) = \frac{1}{\alpha} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(s,t)\right) \tag{5.10}$$

where α is the normalization factor:

$$\alpha = \exp\left(\sum_{x \in Tr(s)} \lambda_i f_i(s,x)\right) \tag{5.11}$$

5.4.5 Selecting features and perceptron training

We have tried to substitute the task of finding optimal linear interpolation weights by the reranker training. Therefore we have only considered forward conditional probabilities $P(t|s)$ of the simple models and their variants limited to certain bucket. However, tests have shown that the reranker is not suitable for the task. And even for various numbers of buckets we weren't able to reach the score that was achieved by simply combining the two models (`POS` and `Tag`) using linear interpolation. The parameters were trained on the vectors extracted from the second part of the `newstest` data set (tmt files 030–064) and evaluated on the first ten files from the newstest data sets.

5.4.6 Linear interpolation training

We have finally decided to build simplified hierarchical model $\mathcal{H}$ based on the `POS` and `Tag` models with optimal pruning thresholds (`POS` thresholds were trained to provide highest score when it is used as the only model; `Tag` pruning thresholds were trained to provide highest score when used as the higher-order model for the optimal `POS` from the previous step) where probabilities are computed using simple linear interpolation:

$$P_{\mathcal{H}}(t|s) = \lambda P_{\texttt{POS}}(t|s) + (1-\lambda) P_{\texttt{TAG}}(t|s) \tag{5.12}$$

The λ was trained by hand on the 250 sentences from the portion of the *newstest* data set and the optimal $\lambda = 0.1$ was used finally.

5.4.7 HMTM Parameter training

We have primarily tested and developed the dictionary using baseline translation approach where formemes were pre-selected for every node prior to the choice of lemma. Incorrectly chosen formemes could force the selection of incorrect lemma even if the dictionary would suggest correct translation with higher probability. Lately a new algorithm was developed by (Žabokrtský and Popel, 2009)Among some promising features such as tree language model it weights the selection of formeme F_t and lexeme L_t using the following equation (where s is the current conditioning attributes):

$$w(F_t, L_t) = \lambda_f P(F_t|s) + \lambda_l \left(\lambda_b P(s|L_t) + (1 - \lambda_b) P(L_t|s)\right) \tag{5.13}$$

The actual computation is slightly more complicated and could include some other features as well but the weights $\lambda_f, \lambda_l, \lambda_b$ are parameters that should be provided by the translation sequence. We have tried to translate a small data set (500 sentences from the `newstest`) using the hierarchical model in order to find weights that provide optimal results. There is no tool for the automatic parameter training in the TectoMT system yet so we have tried to evaluate translations using various setups. We have used $\lambda_f = 1$ and tried to train λ_l. For the optimal $\lambda_l = 0.45$ we have evaluated various λ_b. All results are shown in the Table 5.1. For the final evaluation these weights have been used.

Table 5.1: Training λ_l and λ_b for the HMTM translation

λ_l	NIST	BLEU
0.1	4.7545	0.1347
0.2	4.8613	0.1409
0.3	4.9562	0.1474
0.4	4.9725	0.1502
0.45	4.9822	0.1504
0.5	4.9658	0.1488
0.6	4.9120	0.1460
0.7	4.9117	0.1445
0.8	4.8824	0.1430
0.9	4.8608	0.1428

λ_b	NIST	BLEU
0.1	4.8883	0.1428
0.2	4.9395	0.1464
0.3	4.9516	0.1469
0.4	4.9691	0.1480
0.5	4.9725	0.1502
0.6	4.9674	0.1494
0.7	4.9664	0.1489
0.8	4.9353	0.1470
0.9	4.8690	0.1431

Chapter 6

Extensions

In this chapter we discuss the situations and phenomena where the simple approach tends to be insufficient. We also describe dictionary extensions designed to solve these problems.

6.1 Negation grammateme swapping

Negation related issues described in Section 5.2.1 are partially solved by the `PosNegation` model that can distinguish probabilities for English words in positive and negative form when needed. However this model is incapable of dealing with the situations where negation is lexicalized in English but formative in Czech (e.g. in *necessary* → *nezbytný* the Czech lemma *zbytný* with negation grammateme set to `neg1` should be used). For all these problematic cases the negation grammateme is swapped – its value differs for source and target node (there are only two possible values `neg0` and `neg1`[1]). We can start with the simple fallback model $P(Neg_t|Neg_s)$ that keeps the negation grammateme unchanged:

$$P(Neg_t|Neg_s) = \begin{cases} 1 & \text{if } Neg_t = Neg_s \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

Actual Maximum-Likelihood estimates for this fallback model are shown in the Table 6.1. We train more detailed model $\mathcal{N}$ using the conditional probability $P_{\mathcal{N}}(Neg_t|Neg_s, L_s, L_t)$. We assume that the value with higher probability is always assigned so we can prune the observations whose probability does not differ significantly from the fallback model $P(Neg_t|Neg_s)$ thus satisfying the following inequality:

[1] sometimes the negation grammateme is not assigned at all. In such cases we assume that the value is `neg0`

$$P_{\mathcal{N}}(Neg_t = Neg_s | Neg_s, L_s, L_t) < \alpha_n \tag{6.2}$$

Table 6.1: $P(Neg_t|Neg_s)$ probabilities

	$Neg_s = 0$	$Neg_s = 1$
$Neg_t = 0$	0.989	0.011
$Neg_t = 1$	0.376	0.624

We have pruned the model using $\alpha_n = 0.5$ at first but many assignments were not correct from the linguistic standpoint (examples shown in Table 6.2) so we have decreased the threshold to $\alpha_n = 0.1$. A sample of the most frequent entries is shown in Table 6.3. Finally the model was converted to the exception list and the probabilities were discarded entirely.

Table 6.2: Incorrect negation swaps

Count	Neg_s	L_s	L_t	$P(Neg_t = 1|Neg_s, L_s, L_t)$
3032	0	`find#V`	`naleznout#V`	0.570
2375	1	`create#V`	`vytvořit#V`	0.252
2092	0	`be#V`	`lze#V`	0.575
2061	1	`open#V`	`otevřít#V`	0.254
1544	0	`can#V`	`lze#V`	0.862
1526	1	`use#V`	`použít#V`	0.220
1404	1	`find#V`	`najít#V`	0.341
1331	1	`delete#V`	`odstranit#V`	0.203
1321	1	`start#V`	`spustit#V`	0.355
1316	1	`save#V`	`uložit#V`	0.307
1197	0	`no#D`	`být#V`	0.663

Block `SEnglishT_to_TCzechT::Swap_negation` performs the actual swapping of the negation grammateme using the swaplist derived from the above model. Integrating this component directly into the translation dictionary was considered too complicated mainly because there is no way how arbitrary information could be exchanged easily among various blocks[2]. The block searches through already translated nodes, constructs swaplist keys (collecting source negation grammateme and lemma with part of speech for both sides) and assigns negation value opposite

[2]other than using the schema of the TMT files which has fixed format and does not allow ad-hoc structures

Table 6.3: Most frequent negation swaps, $P(Neg_t|*) = P(Neg_t = 1|Neg_s, L_s, L_t)$

Count	Neg_s	L_s	L_t	$P(Neg_t|*)$
7493	0	fail#V	zdařit_se#V	1.000
7007	0	necessary#A	zbytný#A	1.000
1347	0	dangerous#A	bezpečný#A	0.987
1149	1	tel#N	intel#N	0.000
913	1	known#A	neznámá#N	0.000
841	1	available#A	dispozice#N	0.000
642	1	coming#A	příchozí#A	0.000
631	0	delay#N	prodleně#D	1.000
569	0	immediately#D	prodleně#D	1.000
564	0	create#V	vytvořit#V	1.000
550	1	employment#N	nezaměstnanost#N	0.000
503	0	miss#V	naleznout#V	0.954
485	0	no#D	existovat#V	0.975
480	1	able#A	možný#A	0.000
467	1	contact#V	obrátit_se#V	0.000
431	0	require#V	zbytný#A	1.000
423	0	adverse#A	příznivý#A	1.000
413	1	regular#N	nesrovnalost#N	0.000
396	0	need#V	zbytný#A	1.000
386	0	use#V	používat#V	1.000

to source (`neg0` or `neg1`) if the key is found in swaplist. This correction was implemented in the former transfer within the block `Fix_grammatemes_after_transfer` that recognized only very limited set of words and allowed only correction of negation grammateme to `neg1`. Note that this module should be applied prior to the `Fix_negation` block which further adjusts negation grammatemes according to the context (lexical content is not considered in that block).

Both approaches were evaluated and compared on the 500 sentences from newstest2009 using only `POS` dictionary. Two sentences contained different resolution and in both cases the new approach provided better translation (first case was *informal* → *neformální* , second case *less unambiguously* → *méně jednoznačně*).

6.2 Compounds

Another major issue that was not solved before was the handling of compounds – structures often found in English text that are defined below[3]:

> *A word that consists either of two or more elements that are independent words, such as loudspeaker, baby-sit, or high school, or of specially modified combining forms of words, such as Greek philosophia, from philo-, "loving," and sophia, "wisdom."*

We are primarily interested in the compounds where words are connected by the hyphen. Depending on the choice of tokenization these compounds may be split into standalone components that are processed (tagged and parsed) independently or they can be left as a whole and processed as single unit.

In the parsed CzEng 0.7 corpus these compounds were split and they were not recognized by the dictionary learning process. Though the current translation pipeline often leaves such compounds as a whole – they are not recognized by the dictionary built upon data that does not contain such compounds.

Both approaches have their advantages – if the compounds are split its components are processed by the tagger and they can be easily translated as isolated tokens. Though this is not suitable for non-analytic compounds (those that violate compositionality principle). If we leave the compounds intact we can extract alignment directly and could possibly have them in the dictionary.

6.2.1 Extraction of compounds from CzEng

Compounds were extracted using the following simple algorithm:

1. source tectogrammatical nodes are processed in the order defined by their *deepord*, [4]
2. the longest intervals of nodes interleaved by the hyphens are extracted,
3. for every interval we find target nodes that are aligned with the members of the interval,
4. compound and its translation are printed.

Occurrences of each compound were counted, rare translations (seen less than 5 times) were discarded and standard compound dictionary was built. Part of

[3] The definition was found at `http://dictionary.reference.com/browse/compound`

[4] *deepord* defines the order of the node within tectogrammatical tree. x is to the left of y if $deepord(x) < deepord(y)$

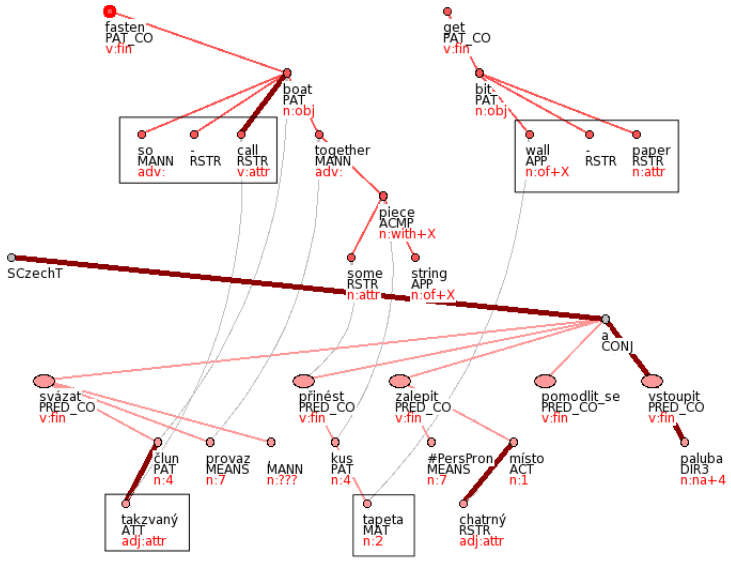


Figure 6.1: Examples of n:1 aligned compounds

speech on source side was discarded. Multiple tokens were allowed on target side. Most frequent compounds are shown in the Table 6.4. The case of *non-member* → *třetí* is quite an odd alignment. We have searched through the text and found out that it is probably due to the collocation *non-member country* → *třetí země* which is commonly used in EU legislative documents (see Section 3.1).

Further examination of the data showed that compounds can be classified by various attributes, mainly by their:

- compositionality,
- fertility.

Compositionality distinguishes between compounds whose meaning can be systematically derived from its parts by patterns or simple rules (*well-X* → `dobře X#A` and compounds whose meaning breaks compositionality principle (such as *foot-and-mouth* → *kulhavka a slintavka*). It is necessary to keep non-compositional or complex compounds in dictionary whereas purely compositional compounds can be translated using pattern based rules or one-by-one translation of components.

Fertility of a compound is the number of words used on tthe arget side of the translation. Many compounds use just a single word (see *wall-paper* → *tapeta*) but sometimes more words are required as in *skimmed-milk* → *odstředěný#A mléko#N* or *log-book* → *lodní#A deník#N* . This aspect needs to be addressed in the front-end of the translation block because extra nodes need to be added

Table 6.4: Most frequent compounds extracted from CzEng 0.7

Count	source	target
206	anti-dump	antidumpingový#A
198	intra-community	společenství#N
131	type-approval	typ#N schválení#N
121	non-member	třetí#C
110	cross-border	přeshraniční#A
97	long-term	dlouhodobý#A
79	tran-european	transevropský#A
74	middle-earth	středozemě#N
74	foot-and-mouth	slintavka#N a#J kulhavka#N
65	skimmed-milk	odstředěný#A mléko#N
63	non-target	cílový#A

to the target tectogrammatical tree. One-to-one principle does not hold in such cases.

Study of the PoS patterns on the target side revealed that about 32.7% of all compounds are translated by a single word and also provided list of the most frequent PoS patterns. These were examined and rules for their compositional members were constructed. Compounds that translate as single verbs were considered to be mostly useless as they consisted of some prefix based compounds (*re-examine* → *přezkoumat*) and many cases with just single occurrence.

Table 6.5: compound POS patterns

$P(Pattern)$	Pattern	Examples
0.1831	N	*time-limit* → *lhůta*
		co-financing → *spolufinancování*
0.1443	A	*medium-sized* → *střední*
		anti-personell → *protipěchotní*
0.1070	A N	*buying-price* → *kupní#A cena#N*
		sea-bed → *mořský#A dno#N*
0.0804	N N	*mountain-top* → *hora#N vrchol#N*
		duty-free → *clo#N osvobození#N*
0.0456	N Z N	*know-how* → *know-how*
0.0367	N A	*director-general* → *ředitel#N generální#A* ,
		knowledge-base → *znalost#N založený#A*

6.2.2 Compound lemmatization

The different tokenization of compounds in the translation system and the training data from CzEng 0.7 (compounds are separated in CzEng 0.7 but they are kept as a single node in the actual translation) have one important side-effect. Components were analysed and lemmatized in CzEng 0.7 while they are kept intact in the current translation system. This reduces the recall of the dictionary while there might be some slight differences between lemma and actual forms in compounds (compare `so-called` and `so-called`). Therefore we added a simple block in our translation scenario that lemmatizes components of compounds using `TnT` tagger and `EnglishMorpho::Lemmatizer`

6.2.3 Rule based translation of compounds

Prefix translation

Even some of the single word translations express compositionality to some extent. For the English Czech language pair we can find many cases where the first word in the English compound is expressed as a prefix on the Czech side. These prefixes have usually low amount of possible translations and it is usually easy to pick the right one with the help of target language analyzer that discards the nonexistent word forms. We've manually constructed a list of the most frequent prefixes along with their possible translations. For a compound $P - W$ where P is prefix with translations $Q_1 \dots Q_n$ and P is the probability of the token in the target language we take the following steps:

1. list of possible translations $T(W)$ is looked up in the dictionary,
2. for each translation $t_i \in T(W)$ we choose:

$$t_i^* = t_i \oplus \operatorname*{argmax}_{Q_j} p_T(t_i \oplus Q_j) \tag{6.3}$$

   ,

3. we discard translation for which $p(t_i)$ is below threshold (t_i was unseen in monolingual corpus)

Another examples of these easy-to-translate prefixes are: *self, anti, half, pre, re, inter, sub*. We have also discovered a specific case of frequent prefix *non-* indicating negation. There are existing mechanisms handling negation during the synthesis. Therefore we will not add this prefix to target lemma. We will just trigger the negation grammateme and leave its interpretation to other components of the translation system.

Due to the nature of the training data translations of prefix-based compounds were sometimes aligned with the content words of the compound. These cases are easily recognized because retrieved translations begin with designated prefixes (e.g. there is *pollinate#V* → *samosprašný#A* which is in fact the correct translation of `self-pollinate` compound).

One-by-one translation

For compounds that were not found in the dictionary and they do not begin with known prefix we will try to translate tokens one-by-one and select the most suitable translation. For compound $C =< w_1, w_2, \ldots, w_n >$ let $T(w_i)$ denominate the set of translations of w_i and let $T_m(w_i) = t_1 | t \in T(w_i) \wedge POS(t) = m$ be the subset of translations with given PoS. For every possible PoS marks pattern $M = \langle m_1, m_2, \ldots, m_n \rangle$ we will choose the most probable sequence of translations:

$$B(M, i) = \underset{t \in T_{m_i}(w_i)}{\operatorname{argmax}} P(t|w_i)$$

$$S^M = \langle S^M(1), S^M(2), \ldots, S^M(n) \rangle$$

Probability of selected sequences is recomputed using the following equation:

$$P(S^M|C) = \lambda \prod_{i=1}^{n} p(S_i^M|w_i) + (1 - -\lambda)P(M)$$

where $P(M)$ is the probability of a given PoS pattern. Fixed weight $\lambda = 0.8$ was used.

6.2.4 Multiple word translations

Translations containing multiple words require the tree structure to be extended with some extra nodes. We decided that all complex structures should be represented as a head node and one or more direct children. As the head node interacts with the rest of the tree most tightly it will be placed into the existing node. New nodes will be created for all the remaining words. We need to resolve the following:

- select head node of the translation,
- fill in minimal set of node attributes of the children nodes.

According to the various compounds and patterns encountered we decided that the rightmost word compatible with fomeme should be selected as the head. This is valid for `AN` patterns and even for most of the `NN` patterns.

Newly created nodes need to be initialized with some attributes for the synthesis to work properly. The minimal set of the attributes contains:

- *nodetype* should always be set to `complex`,
- *sempos* can be usually determined by designated POS,
- *functor*,
- *formeme*.

Both formeme and functor depend on part of speech and also on the current context. About 94.75% of adjectives in compounds get `RSTR` functor with `adj:attr` formeme. Therefore we use fixed assignment. For other part of speech categories the functors and formemes do not prevail that much (see Table 6.6).

Table 6.6: Functor distribution within compounds

$P(f_1\|N)$	f_1	$P(f_2\|A)$	f_2	$P(f_3\|D)$	f_3
0.27	RSTR	0.94	RSTR	0.24	CM
0.14	ACT	0.02	PAT	0.18	MANN
0.13	ID	0.01	ACT	0.16	RSTR
0.12	PAT	0.01	MANN	0.10	EXT
0.10	APP			0.09	TWHEN
0.04	LOC			0.05	RHEM
0.04	PAR			0.03	PRED
0.02	MAT			0.02	THL

Chapter 7

Evaluation

7.1 T-tree match

In early stages of the project we based the evaluation of progress exquisitely on BLEU/NIST score (see (Papineni and Zhu, 2002) and (Martin and Le, 2008)). This metric is still used for evaluation of important milestones because it provides score that is comprehensible and established within the field but it is unfeasible to use BLEU/NIST score for quick evaluation (especially during training) because its computation is very costly. In order to obtain BLEU/NIST score as a measure of the transfer component we need to run both transfer and synthesis sequences. Therefore we tried to come up with a simplified metric that is cheaper to compute but provide results comparable to that of BLEU/NIST. If we had evaluation data where reference sentences are parsed up to tectogrammatical trees and aligned with source nodes we could compare shape of the respective tectogrammatical representations. We expect that if we have reference tree R and two translation trees T_1 and T_2 where ratio of nodes consistent with R is higher for T_1 then we expect that the final translation would also receive higher score.

This approach barters cost of synthesis for the cost of analyzing and aligning the reference sentences. Altough the latter is more expensive in a single run we can apply it once on the set of evaluation files which will be then used many times for evaluation.

Full-featured evaluation tool has been implemented because we also required some deeper insight into the workings of the dictionary and especially into the nature and typology of its most frequent errors.

7.1.1 Basic metric

Let $\mathcal{T}_s$ ($\mathcal{T}_t$ and $\mathcal{T}_r$) be the source (target and reference) tectogrammatical tree (see Figure 7.1) There are one-to-one relations between counterpart source and target

nodes (target tree is constructed by cloning the source) and also relations between aligned source and reference nodes (these relations are exploited while training the dictionary). Let $A(\mathcal{X}, \mathcal{Y}) \subseteq \mathcal{X} \times \mathcal{Y}$ denote set of aligned pairs of nodes between trees $\mathcal{X}$ and $\mathcal{Y}$. Having the direct sets $A(\mathcal{T}_r, \mathcal{T}_s)$ and $A(\mathcal{T}_s, \mathcal{T}_t)$ we can easily define transitive set $A^*(\mathcal{T}_r, \mathcal{T}_t) \subseteq \mathcal{T}_r \times \mathcal{T}_t$ where $(r, t) \in A^*(\mathcal{T}_r, \mathcal{T}_t)$ iff there is $s \in \mathcal{T}_s$ such that $(r, s) \in A(\mathcal{T}_r, \mathcal{T}_s)$ and $(s, t) \in A(\mathcal{T}_s, \mathcal{T}_t)$.

For every $t \in T$ one of the following cases hold:

1. t is fully aligned iff $\exists r \in R : (r, t) \in A^*(\mathcal{T}_r, \mathcal{T}_t)$
2. t has no reference iff $\exists s \in S : (s, t) \in A(\mathcal{T}_s, \mathcal{T}_t) \wedge \forall r \in R : (r, t) \neq A^*(\mathcal{T}_r, \mathcal{T}_t)$
3. t has no source iff $\nexists s \in S : (s, t) \in A(\mathcal{T}_s, \mathcal{T}_t)$

For the direct evaluation only fully aligned nodes are considered. Let us therefore define set $\mathcal{T}^R \subset \mathcal{T}$ containing only nodes that are fully aligned. We can define the following alignment projections:

- $a_s : \mathcal{T}^R \mapsto \mathcal{T}_s$ such that $(t, a_s(t)) \in A(T_t, T_s)$,
- $a_r : \mathcal{T}^R \mapsto \mathcal{T}_r$ such that $(t, a_r(t)) \in A^*(T_t, T_r)$.

Finally for $r \in \mathcal{T}_r$ and $t \in \mathcal{T}_t$ we say that nodes are equivalent ($r \equiv t$) iff they have the equal lemma and part of speech. Then we can define *t-tree match* metric $M_1(\mathcal{T}_r, \mathcal{T}_t)$ as a ratio of nodes consistent with reference:

$$M(R, T) = \frac{|\{t | t \in \mathcal{T}^R \wedge a_r(t) \equiv t\}}{|T^R|}$$

Without loss of generality we can work with forests instead of trees and thus compute M_1 for entire text.

7.1.2 Recall metric

Current TMT schema [1] can hold ordered lists of suggested translations for every node. Therefore we can compute sort of k-recall score that indicates how good we would perform if we had oraculum that would correctly pick translation out of best k variants. Assume that for every $t \in \mathcal{T}_t$ we denote i-th best translation variant by t_i. Then we will define *k-recall t-tree match metric* $M_k(\mathcal{T}_r, \mathcal{T}_t)$ in the following way:

[1] TMT is shorthand for TectoMT and is used as an extension for xml files that are used by the TectoMT system

$$M_k(T_r, T_t) = \frac{|\{t|t \in \mathcal{T}^R \wedge \exists i \leq k : t_i \equiv a_r(t)\}|}{|\mathcal{T}^R|}$$

It is clear that $M_1 = M$. We can therefore work only with M_k from now on.

7.1.3 Excluding nodes by filters

In some cases we might be interested in measuring the performance only on a given subset of nodes. For arbitrary $X \subseteq \mathcal{T}^R$ let:

$$M_k^X(R, T) = \frac{|\{t|t \in X \wedge \exists i \leq k : t_i \equiv a_r(t)\}|}{|X|}$$

There are various practical applications of these restrictions. If we are interested only in how well the dictionary picks nouns we can consider set of $t \in \mathcal{T}^R$ for which part of speech of $a_s(t)$ is `N`.

Another very interesting example is related to the formeme compatibility constraint. If we have formemes already assigned to target tree nodes we must select only translations (lexemes) that are compatible with formemes (for further details about the formeme compatibility see Section 1.3). In some cases it might happen that for given $t \in T$ the reference lexeme L_r is not compatible with the target node formeme F_t. Even though the dictionary could have suggested the valid translation it would not be selected as it would break the formeme compatibility constraint. In such cases dictionary can't improve the translation score. Using the filtered metric we could omit such problematic nodes from the evaluation.

Various filters can be switched on at the run-time of the evaluation tool. We have primarily used the following two:

- `POS(X)` – t is considered iff $a_s(t)$ part of speech is `X`,
- `F-Comp` – t is considered iff $a_r(t)$ is compatible with F_t.

7.1.4 Comparison with BLEU/NIST

We would like to see whether TTM metric is consistent with BLEU/NIST in a way that maximizing one metric is equivalent to the maximization of the others. We expect that this relation doesn't hold for all cases because not even BLEU and NIST are always consistent with each other so we are going to see to what extent the similarity between the BLEU/NIST and simplified TTM metric hold.

Definition 4 *Metrics ψ and ϕ are **consistent** on the set $\mathcal{X}$ iff:*

$$\operatorname*{argmax}_{x \in \mathcal{X}} \psi(x) = \operatorname*{argmax}_{x \in \mathcal{X}} \phi(x) \tag{7.1}$$

Which is equivalent to:

$$\forall x, y : \psi(x) \geq \psi(y) \Leftrightarrow \phi(x) \geq \phi(y) \tag{7.2}$$

We have computed BLEU, NIST, TTM(k) and `F-Comp`(k) metrics on various `POS` dictionary samples. Results can be seen in the Table 7.1. The table shows that not even BLEU and NIST are always consistent with each other (compare rows 2 and 3). Correlations between metrics have been computed and the results are shown in the Table 7.2. Therefore we can say that `F-Comp` metric is more consistent with both NIST/BLEU, and TTM(k) is relatively consistent with BLEU. The Table 7.1 shows that TTM based metrics can be used to select candidates for further costly evaluation with BLEU/NIST – results with high BLEU/NIST scores tend to receive high TTM based score. However for some cases the TTM based score is high even if the corresponding BLEU/NIST score is relatively low.

Table 7.1: BLEU/NIST and TTM score for various thresholds

Thresholds				TTM(n)			F-Comp(n)		
α_t	α_s	NIST	BLEU	0	1	2	0	1	2
–	–	4.4706	0.1120	0.4880	0.6203	0.6708	0.5713	0.7120	0.7649
0.01	0.001	4.3859	0.1116	0.5073	0.6517	0.7087	0.5693	0.7238	0.7807
0.01	0.01	4.3880	0.1114	0.5073	0.6506	0.7067	**0.5699**	0.7232	0.7784
0.01	0.02	4.3834	0.1109	0.5060	0.6469	0.6980	0.5682	0.7195	0.7710
0.01	0.05	4.3642	0.1095	0.5008	0.6374	0.6839	0.5635	0.7114	0.7595
0.01	0.1	4.2883	0.1048	0.4904	0.6230	0.6669	0.5521	0.6968	0.7433
0.02	0	4.3947	**0.1119**	0.5069	0.6517	0.7072	0.5689	0.7239	0.7801
0.02	0.001	4.3969	**0.1118**	**0.5076**	0.6515	0.7067	**0.5699**	0.7240	0.7799
0.02	0.01	4.3978	**0.1117**	0.5071	0.6506	0.7030	**0.5699**	0.7232	0.7765
0.02	0.02	4.3949	0.1111	0.5065	0.6471	0.6975	0.5690	0.7201	0.7705
0.02	0.05	4.3753	0.1096	0.5014	0.6372	0.6815	0.5646	0.7118	0.7577
0.02	0.1	4.2984	0.1053	0.4913	0.6232	0.6629	0.5543	0.6980	0.7402
0.05	0.001	4.4026	0.1110	**0.5076**	0.6497	0.6940	0.5707	0.7228	0.7681
0.05	0.01	4.4004	0.1109	0.5069	0.6490	0.6905	0.5705	0.7222	0.7651
0.05	0.02	4.3998	0.1104	0.5063	0.6466	0.6857	0.5699	0.7205	0.7605
0.05	0.05	4.3756	0.1088	0.5023	0.6361	0.6694	0.5659	0.7106	0.7455
0.05	0.1	4.3177	0.1057	0.4922	0.6216	0.6508	0.5554	0.6959	0.7274
0.1	0.001	4.3932	0.1095	0.5076	0.6433	0.6727	0.5733	0.7206	0.7510
0.1	0.01	4.3926	0.1095	0.5073	0.6429	0.6694	0.5736	0.7200	0.7485
0.1	0.02	4.3947	0.1094	**0.5076**	0.6401	0.6659	0.5738	0.7178	0.7455
0.1	0.05	4.3683	0.1074	0.5038	0.6297	0.6523	0.5692	0.7071	0.7322
0.1	0.1	4.3309	0.1059	0.4977	0.6197	0.6396	0.5636	0.6976	0.7198

Table 7.2: Correlation of various metrics

			TTM(i)			F-Comp(i)		
	NIST	BLEU	1	2	3	1	2	3
NIST	1.000	0.882	0.455	0.510	0.476	0.873	0.764	0.607
BLEU	0.882	1.000	0.636	0.795	0.824	0.786	0.924	0.888
TTM(1)	0.455	0.636	1.000	0.902	0.597	0.751	0.845	0.522
TTM(2)	0.510	0.795	0.902	1.000	0.871	0.648	0.936	0.811
TTM(3)	0.476	0.824	0.597	0.871	1.000	0.410	0.823	0.977
F-Comp(1)	0.873	0.786	0.751	0.648	0.410	1.000	0.835	0.479
F-Comp(2)	0.764	0.924	0.845	0.936	0.823	0.835	1.000	0.836
F-Comp(3)	0.607	0.888	0.522	0.811	0.977	0.479	0.836	0.000

7.2 Oraculum evaluation

To explore theoretical limits of the translation dictionary we have also experimented with *oraculum evaluation*. The oraculum with threshold k simply picks the correct translation (according to the reference) for given node if it is found among the top k translations provided by the dictionary. The implementation requires that the reference tectogrammatical trees are built and aligned (the same requirements as for the TTM metric described in the previous Section 7.1).

7.3 Intrinsic evaluation

Intrinsic evaluation – isolated evaluation not in the context of the wider translation system. TTM is attempt to isolate the evaluation to the bare-bone though it still relies on some of the MT pipeline. We could however compare former and new dictionary "by hand" and see how they differ and guess which one is better or worse.

7.4 Extrinsic evaluation

Extrinsic evaluation should consist of comparing BLEU/NIST score between the former dictionary and new dictionary using the same pipeline. Also we should note that new dictionary is useful because it provides backward probabilities that can be used by more evolved pipelines (such as tree Viterbi). Show tables with results for various evaluation sets (wmt/czeng sample/...).

7.5 Manual evaluation

Various MT approaches could be also compared manually. Though this method requires lot of time and human resources it is still widely used technique mainly because automated metrics are not capable of measuring translation quality in its full complexity. There are many subtle features that contribute to the intuitive feeling of accurate and good translation including fluency, word order and natural syntactic structure. We are in a state of sinful expectation if we hope for a simple metric that could express all of this without actually being full-blown high-quality MT system itself. For such reasons human evaluation is inevitable part of many MT system comparisons. In this section we will study the differences between translation produced using the original dictionary (see Section 2.3) and by the presented new dictionary. We are certain that any difference in the translation is induced by the different choice of translation (even though if syntactical difference occurs in later phase). As we would like to come up with some comparison we distinguished between the following three cases:

- **improvement** occurs whenever the new dictionary suggests better translation than the former dictionary,
- **draw** is situation when both dictionaries suggest different translations but both are approximately equally good (or bad),
- **regression** is opposite to the improvement. Occurs whenever the new dictionary suggests translation that is worse than the former one.

If we encounter situation where it is unclear whether it should be treated as regression or improvement we will treat it as being draw. Errors were furthermore classified according to their type using one of the following categories (multiples allowed):

- `Sem` – translation is semantically inappropriate,
- `Comp` – wrong translation of compound,
- `Coll` – incorrectly translated collocation,
- `BadPOS` – translation with incorrect part of speech (might be caused by invalid formeme).

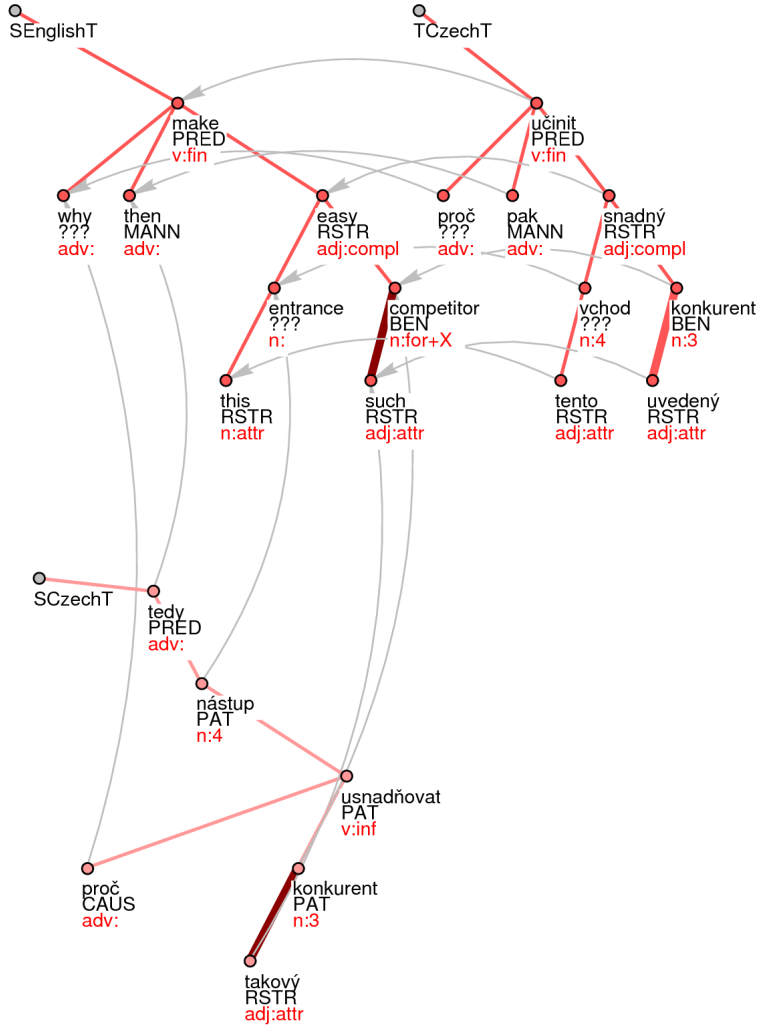


Figure 7.1: Reference, source and target tectogrammatical trees (simplified)

Chapter 8

Implementation

In this section a short description of the implementation is given. Further details – especially documentation of the Perl modules – can be found in the Appendix C. The implementation can be divided into the following categories:

- internals – perl modules implementing Maximum-Likelihood and other derived models, translation dictionary providing convenient interface
- translation blocks – alternative transfer blocks using the new translation dictionary backend, implementation of extensions
- extraction procedures – scripts, blocks and makefiles used for the dictionary training and development of models
- evaluation tools – most of the evaluation tools were already part of the TectoMT system, TTM based metrics and tools were added
- tests – small unit tests were used to ensure that the code was not broken during the development

8.1 Internals

8.1.1 Models

Collection of underlying classes that represent the dictionary are used both during the training and later when we are using the dictionary within the MT system. They are focused primarily on providing object representation of the translation model which:

- is capable of representing various models,

- provides interface for various tasks such as filtering, traversing the tree, smoothing,

- can be extended by custom functions such as pair filters and probability models.

The basic conditional model is implemented in the class `DictBuilder::Model`. Specific instances of the model need to implement at least `make_context` method that converts source attributes to conditioning string. Models support two input formats – topsort lists the aligned pairs sorted by their occurence count and is used to build Maximum-Likelihood estimates. Final models are printed so that conditioning strings are sorted alphabetically and all possible outcomes for certain conditioning string are sorted by their probabilities in a decreasing order. Documentation of the class interfaces for both basic and derived models can be found in the Appendix C.

8.1.2 Translation dictionary

The class `TranslationDict::EN2CSAlt` provides middle layer that sits between the translation block and the underlying translation model (or models). It queries the translation model and implements some rule based fallbacks and the implementation of compound translation.

8.2 Translation blocks

Alternative translation block for both baseline and HMTM based translation approaches has been developed. Both blocks were modified so that they use the new backend. Baseline block was further modified so that it fills the entries into the `language_model/t_lemma_variats` so that the TTM based metrics could be evaluated for the baseline as well. `Swap_negation` extension was implemented as standalone block.

8.3 Training tools

The block `Print::English_Czech_aligned_pairs` is the core part of the dictionary training; various post-processing tools that handle data splitting ,filtering and transfromation into format suitable for specific models training have been also implemented. Additionaly there are tools implementing the pruning (`raw_model.pl`) and perceptron weight training (`rerank.pl` and `create_clusters.sh`).

8.4 Evaluation tools

The block *Print::TTM* extracts data from the source and translated t-trees. `ttm.pl` is script that will compute the actual TTM score and implements some filtering capabilities.

8.5 Tests

Some simple tests based on the `Test::More` class were implemented. The test driven development theory suggests extensive use of tests during the development because they can prevent the unintentional code breakage or they can help to discover the problem early. We have adopted the tests in the late stage of the development and we have used it primarily to:

1. test the new functionality on well defined isolated test data,
2. simulate discovered bugs – this allows us to test that the fix works and that the bug doesn't re-appear later on.

The current test set has still a very limited coverage and has to be run manually which is tedious. Still we believe that it contributed to the faster and more convenient development.

8.6 Lexicon training

Dictionary training uses blocks for data extraction, shell scripts bfor data preprocessing and various perl tools based on the `DictBuilder::Model` representation that perform pruning and merging of partial results. This chapter describes the construction of `Tag` and `POS` models so that the reader could use provided tools to build his own models easily. All the tools can be found in the TectoMT repository within `personal/rous/build_dictionary`.

8.6.1 Building your own dictionary

We assume that you have your training data available in TMT format with both SEnglishT and SCzechT layers present and aligned. If you don't have them you should have a look into `tools/format_convertors` where you can find some scripts that can convert between well-known formats and the TMT. If your format is not supported you will have to write your own conversion script. If needed TectoMT system can be also used to annotate and align your corpus if you don't have the annotation yet (or not as deep as is required).

Extraction of the aligned pairs works with lists of files that are to be processed because task can be paralellized and filelists can be easily splitted into smaller parts. These filelists are easy to get but you can use make to build it for you:

```
INPATH=~/my_training_corpus make train.fl
```

Then you can run the extraction process. If you have access to the grid you can run the tasks parallely:

```
make train.fl.qrun
```

Otherwise you will have to settle for the linear processing. In such case you can also split the whole filelist into parts of roughly the equal size with files from a single source[1]:

```
make train.fl.categories
for file in categories/*.fl ; do
make $file.run
done
```

Extraction will create files within the `extracted` dictionary. These files are just large lists of all encountered alignments. Now we are going to build two baseline models – `Pos` and Tag – first we are going to make frequency lists of translations (which should reduce the file size) and then we will apply some modest pruning and will merge the partial models together.

```
make tag-to-pos-topsort
make tag-to-pos-dicts
make tag-to-pos-reduce

make pos-to-pos-topsort
make pos-to-pos-dicts
make pos-to-pos-reduce
```

For the construction of `POS` model, tags need to be translated using the conversion described in Section 4.1

[1]Source name detection was developed for the CzEng and it depends on the directory structure heavily

8.6.2 Pruning tools

The models can be pruned using the `raw_model.pl` tool that supports various pruning techniques described in Section 5.4 The following attributes can be specified:

- `--prune-cp-below` α_t – forward conditional probability threshold
- `--prune-rev-cp-below` α_s – backward conditional probability threshold
- `--prune-infrequent-cp-below` α_{UNK} – eliminate infrequent pairs with probability lower than the given threshold
- `--infrequent-count` k – eliminate pairs
- `--prune-numbers` – eliminate numbers when reading the input
- `--no-filtering` – do not apply filters
- `--en-whitelist file` – keep only those pairs where English token has been found within the file
- `--cs-whitelist file` – keep only those pairs where Czech token has been found within the file
- `--min-count` k – eliminate pairs that were not observed at least k times

8.7 Transfer blocks

The basic block `SEnglishT_to_TCzechT::Baseline_tlemma_translation_alt` requires formemes to be already selected for the nodes. For every node list of translations both from the dictionary and from its extensions are obtained and then matched against the formeme. First compatible translation is assigned. Best k translations (regardless of the compatibility) are assigned to the `translation_model` fields. They are later used for dictionary evaluation but they can be also used by various advanced translations block to adjust choice of both formeme and compatible translation.

Dictionary block consists of component that take care of:

- interaction with the dictionary,
- translation rules for nonstandard situations, translation of compounds,
- formeme compatibility detection – for each word all possible analyses are generated and then matched against formemes .

Chapter 9

Conclusion

We have built the translation dictionary using large paralel corpora automatically annotated up to tectogrammatical layer. We have trained models of various complexity and studied how these models could be combined together to prevent sparse data problem and provide higher performance. The attempt to train the model parameters automatically using reranker has shown that the algorithm is not suitable for this particular task. We have settled for the simplified combinatino based on the linear interpolation and we have searched for the parameters manually. We believe that the adoption of some MERT based training tool such as (Zaidan, 2009) into the TectoMT system would simplify the further parameter training tasks where the reranker doesn't work well. This tool could be used to find optimal pruning thresholds for basic models and weights of the smoothed hierarchical models. TTM based metrics could be used with MERT training to speed up the evaluation phase.

Using additional conditioning attributes such as tags could discriminate between various contexts in order to provide more accurate translations. The final interpolated hierarchical model has been evaluated on the 500 sentences from the `newstest` set. The resulting BLEU score 0.1170 is sligtly better than the former dictionary 0.1120. The NIST metric, however, show decrease from 4.4706 points to 4.4285. When the dictionary was used with the HMTM based transfer it shows more significant improvements - the former dictionary has BLEU 0.1349 and NIST 4.7862 while the new dictionary has BLEU 0.1485 and NIST 4.9967. The results are shown in the Table 9.1. The new dictionary shows improved translation quality especially when used with the new HMTM based translation approach.

There is still space for further improvements. Due to the unbalanced nature of the training corpus where large amount of texts came from the computer domain, some domain-specific translation have prevailed (such as *driver* $\rightarrow$ *ovladač*) – this should be addressed by further balancing the texts from various sources. The dictionary is also having problems translating compounds spanning multiple words.

Table 9.1: Comparison of the dictionaries

Scenario	BLEU	NIST
Baseline with the former dictionary	0.1120	4.4706
Baseline with the new dictionary	0.1170	4.4285
HMTM with the former dictionary	0.1349	4.7862
HMTM with the new dictionary	0.1485	4.9967

However, this issue is partially addressed by the tree language model that is part of the HMTM based transfer. The t-tree isomorphism assumption is another issue that is quite problematic for the translation of some compounds where deletion or insertion of nodes is assumed. This, however, is slightly beyond the basic concept of the translation dictionary.

Bibliography

Český národní korpus - syn2005. 2005. 23, 24

Ondřej Bojar and Zdeněk Žabokrtský. CzEng: Czech-English Parallel Corpus, Release version 0.5. *Prague Bulletin of Mathematical Linguistics*, 86:59–62, 2006. ISSN 0032-6585. 23

Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, and Vladislav Kuboň. Prague Czech-English Dependecy Treebank: Syntactically Annotated Resources for Machine Translation. *4th International Conference on Language Resources and Evaluation*, 2004. 23

BNC Consortium. *The British National Corpus*. Oxford University Computing Services, 2007. 23, 24

Jan Cuřín. *Statistical Methods In Czech-English Machine Translation*. PhD thesis, Charles University, 2006. 21, 35

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jiří Havelka, Jan Štěpánek, and Marie Mikulová. Prague dependency treebank 2.0. 2006. 13

W.John Hutchins and Harold L. Somers. *An introduction to machine translation*. Academic Press, 1992. 9

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, June 2007. 12

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics, Vol. 22, No.1*, 1951. 27

David Mareček, Zdeněk Žabokrtský, and Václav Novák. Automatic Alignment of Czech and English Deep Syntactic Dependency Trees. *Proceedings of EAMT 2008*, 2008. 24

Alvin F. Martin and Audrey N. Le. Nist 2007 language recognition evaluation. *Odyssey-2008*, (016), 2008. 54

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. 11

Roukos S. Ward T. Papineni, K. and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 54

D. Varga, L. Németh, P. Halácsy, V. Trón A. Kornai, and V. Nagy. Parallel corpora for medium density languages. *Proceedings of the RANLP 2005 Conference*, pages 590–596, 2005. 24

A. K. C. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, pages 7:599 – 609, 1985. 27

Zdeněk Žabokrtský and Martin Popel. Hidden Markov Tree Model in Dependency-based Machine Translation. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, August 2009. in print. 15, 44

Zdeněk Žabokrtský, Petr Pajas, and Jan Ptáček. Tectomt: Highly modular mt system with tectogrammatics used as transfer layer. *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, 2008. 13

Omar F. Zaidan. Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems . *The Prague Bulletin of Mathematical Linguistics*, pages 79–88, January 2009. 66

Ruiqiang Zhang and Eiichiro Sumita. Boosting statistical machine translation by lemmatization and linear interpolation. *Proceedings of ACL2007 Demo and Poster Sessions*, pages 181–184, 2007. ISSN 0032-6585. 12

Appendix A

List of abbreviations

BNC	British National Corpus
MT	Machine Translation
NLP	Natural Language Processing
PDT	Prague Dependency Treebank
PoS	Part of Speech
PML	Prague Markup Language
SMT	Statistical Machine Translation
TMT	XML based file format used by the TectoMT system
TTM	Tecto-Tree Match, see page 54

Appendix B

Content of the CD

The CD supplied with this thesis contains this text in PDF format (`thesis.pdf`), final data files (in `data`), hierarchical model description (`final.model`), test set and scenarios used for the dictionary evaluation (in `evaluation` as well as some neccessary evaluation tools(`mteval-v11b.pl` and `txt_2_mteval_sgm.pl`) and the current checkout of the TectoMT repository. Note that for running the TectoMT, additional resources are required (data files, models, additional Perl libraries). For more details about the running of the TectoMT please refer to the Tutorial[1].

Data

- `frequency.cs` – frequency list of Czech lemmas
- `frequency.en` – frequency list of English lemmas
- `pos-cp-0.02-rev-0.001.dict` – `Pos` model with pruning thresholds $\alpha_t = 0.02, \alpha_s = 0.001$.
- `tag-cp-0.001-rcp-0.02-wl.dict` – `Tag` model with pruning thresholds $\alpha_t = 0.001, \alpha_s = 0.02$ and with Czech and English lemmas pruned by the `Pos` model
- `swaplist-new` – negation swap list for the `Swap_negation` block

Evaluation

- `sgm` – evaluation set

[1] see `https://wiki.ufal.ms.mff.cuni.cz/external:tectomt:tutorial`

`newstest2009-ref.cz.sgm` – reference Czech translation

`newstest2009-src.en.sgm` – source English text

`newstest2009-tst.cz.sgm` – test translation by the HMTM based transfer with the new dictionary

- `scenario` – scenarios used for the evaluation

  `all_s0_analysis.scen` – English an-alysis common for all tests

  `all_s2_synthesis.scen` – Czech synthesis common for all tests

  `new_s1.1_hmtm_transfer.scen` – HMTM based transfer using the new dictionary

  `new_s1.2_hmtm_transfer.scen`

  `new_s1_baseline_transfer.scen` – baseline transfer using the new dictionary

  `old_s1.1_hmtm_transfer.scen` – HMTM based transfer using the former dictionary

  `old_s1.2_hmtm_transfer.scen`

  `old_s1_baseline_transfer.scen` – baseline transfer using the former dictionary

TectoMT

This directory contains recent checkout of the TectoMT repository. For more detailed description of the repository layout please refer to the Developer's Guide[2]. Only the most important directories for the thesis are described here.

- `personal/rous/thesis` – source files for this thesis text
- `personal/rous/build_dictionary` – various tools and scripts for the dictionary extraction
- `personal/rous/build_dictionary/evaluation` – directory with various evaluation scenarios
- `libs/other/DictBuilder` – implementation of the Models and other perl tools
- `libs/other/TranslationDict` – implementation of both the former `EN2CS` and new `EN2CSAlt` dictionaries

[2]see `http://ufal.mff.cuni.cz/tectomt/guide/guidelines.html`

- `libs/blocks/SEnglishT_to_TCzechT` – transfer blocks

Appendix C

API Documentation

This apendix lists the API documentation of various blocks and perl classes developed in the course of this thesis.

C.1 Model

`filter_pairs($filter)`

Apply given boolean function $filter on every pair ($context, $outcome) and prune pair that have returned true. Returns number of pruned pairs.

The filter is called with the following parameters:

```
&$filter($dictionary, $context, $outcome)
```

`get_translations($context_ref)`

This method obtains sorted list of translation candidates for the given context.

context_ref should be hashref with various attributes. Actual value depends on the instance of the model. The following attributes are often used:

- en_tlemma,
- en_tag,
- en_form,
- en_negation – value of the source `gram/negation`

Translations are hashmaps fith following attributes:

- cs_given_en, en_given_cs, count, en_count

- cs_tlemma, cs_pos
- features – hashmap with feature values

`get_translations_lite($lemma, $tag)` Retrieve translations for given lemma and tag (lightweight context)

`get_values(context, outcome)`

Return hashref map with en_to_cs, cs_to_en and count values for given entry. Return empty hashref if given pair doesn't exists within the model.

`priv_get_translations($context)`

Get array of all outcomes for given context. Each outcome is hashref with the following attributes:

- *cs_given_en, en_given_cs* – conditional probabilities $P(cs|en)$, $P(en|cs)$
- *count* – number of occurrences
- *cs_tlemma* – outcome lemma
- *cs_pos* – outcome part of speech (might be undefined for certain models)
- *key* - outcome key (lemma#pos)
- *source* - name of the model providing this

`set_param(parameter, value)`

Set *parameter* to *value*.

`set_param(parameter)`

Get actual value of *parameter*

`get_bucket_thresholds`

Return array of frequencies (counts) C(1),C(2), ..., C(n) such that they divide data into roughly equally sized partitions P(0) .. P(n) such that

(c,d) is within P(i) iff C(i+1) >= count(c) > C(i)

where C(0) = -infinity and C(n+1) = +infinity

This class implements the basic translation model and universal Loader that can load inherited models. Basic functionality regarding:

opening and parsing input files – suppor for dictionary and topsort formats

filtering and pruning – see methods `add_filter, delete_pairs`

item traversal –`apply_method_on_pairs, get_all_contexts, get_outcomes`

Maximum-Likelihood model – `get_cond_prob, get_reverse_cond_prob`

C.1.1 Model::Tag

make_context

Create context string for the TAG model. Requires *en_tlemma* and *en_tag* attributes to be defined.

The TAG model could be placed on top of the PoS model to increase the discriminative power.

C.1.2 Model::Frequency

`load_lists($en_file, $cs_file)`

Load frequency lists from the given files.

`make_context`

Create context for the `DictBuilder::Model::Frequency`. Requires *en_tlemma* to be defined.

`priv_get_values`

Returns frequencies and counts from the monolingual wordlists using the following keys:

Frequency lists of tokens based on the monlingual frequency lists. get_values method supported for reading the actual probabilities.

C.1.3 Model::PosNegation

`make_context`

Create conditioning string for the PosNegation model. Requires *en_tlemma, en_tag* and *en_negation* to be defined.

PosNegation model should be used to discriminate between positive and negative forms and their possible translations.

C.1.4 Model::Hierarchical

`autosplit(`*`number_of_buckets`*`)`

Create bucket thresholds for all active models. Buckets are constructed so that every bucket has approximately equal probability of occurence.

`get_logbucket(N)`

Compute bucket number using the following equation:

$$B(N) = \frac{\log(N)}{\log(\text{params.logbucket})}$$

`get_splitbucket(count, splits)`

Compute bucket number of given count by using providen split list *splits* or using the internal params.splits

`parse_file(file)`

Load Hierarchical model from given *file*. Format expects tab separated lines in the following format:

```
MODEL model_name model_file
...
WEIGHT weight_file
```

Models are added in the same order as they appear in the file. First model provides list of translations while other models will only modify the probabilities and contribute to the features.

If `WEIGHT` is given (should be specified only once within the model description file) then the final score for translations will be computed using `ML::Reranker` using specified weights.

`add_model(model_name, file)`

Adds model named *model_name* with contents loaded from *file* to the end of model list.

`get_model(name)`

Access model with given *name*

`load_reranker(file)`

Load ML::Reranker weights from *file*

`unload_reranker`

Remove reranker from the model. This is useful especially when constructing clusters for weight training.

`get_translations( src_lemma, src_tag, attr_hashref )`

Returns list of translations suggested by the Hierarchical model.

`sort_translations`

Recompute translation probabilities accoriding to the model parameters and reranker.

Probabilities are then normalized (unless they sum to zero) and sorted.

`get_all_translations`

This method retrieves all possible translations for given context. According to the params.selector it either constructs the list from the basle model translations, if `params.selector == 'merge'` then all models are queried and translations present in at least one models are suggested.

`get_all_contexts`

This method will return all possible context contained within the model. According to the param.selector it will either use base model to obtain the list or it will enrich the list with all contexts from all models. This could be a bit problematic if the models are of different type.

`Load`

This is the `Model::Hiearchical` factory. It takes single optional argument which is filename with model description. If ommited then the model will attempt to use following environment variables to determine its contents:

```
TMT_PARAM_HIERARCHICAL_MODELS="M1 M2 M3 ..."
```

Space separated list of models involved with the traditional naming format: `Mi = @ModelName:[\#[FileFormat:][ModelType=]]Filename` where:

- `ModelName` identifies the model within Hierarchical model
- `FileFormat` is either t for topsort or d (default) for dictionary (with precomputed probabilities)
- `ModelType` is class used for given model – `DictBuilder::Model::ModelType` will be used if defined, `DictBuilder::Model` otherwise
- `Filename` contains model data

```
TMT_PARAM_HIERARCHICAL_WEIGHT="weight_file"
```

Specifies weight file for `ML::Reranker` to be used with the Hierarchical model

Appendix D

Translation scenarios

Scenarios used to evaluate both the former and the new dictionary are listed here. For the English analysis and Czech synthesis the procedure has not been changed in the course of this thesis so the common scenarios are used. For the transfer phase both the baseline and HMTM based scenarios are presented for both the former and the new dictionary.

Analysis

```
SEnglishW_to_SEnglishM::Penn_style_tokenization
SEnglishW_to_SEnglishM::TagMorce
SEnglishW_to_SEnglishM::Fix_mtags
SEnglishW_to_SEnglishM::Lemmatize_mtree
SEnglishM_to_SEnglishA::McD_parser_local
    TMT_PARAM_MCD_EN_MODEL=conll_mcd_order2_0.01.model
SEnglishM_to_SEnglishA::Fix_McD_Tree
SEnglishP_to_SEnglishA::Fix_multiword_prep_and_conj
SEnglishA_to_SEnglishT::Mark_auxiliary_nodes
SEnglishA_to_SEnglishT::Mark_negator_as_aux
SEnglishA_to_SEnglishT::Build_ttree
SEnglishA_to_SEnglishT::Mark_named_entities
    TMT_PARAM_NER_EN_MODEL=ner-eng-ie.crf-3-all2008.ser.gz
SEnglishA_to_SEnglishT::Fill_is_member
SEnglishA_to_SEnglishT::Fix_tlemmas
SEnglishA_to_SEnglishT::Assign_coap_functors
SEnglishA_to_SEnglishT::Fix_is_member
SEnglishA_to_SEnglishT::Distrib_coord_aux
SEnglishA_to_SEnglishT::Mark_clause_heads
```

```
SEnglishA_to_SEnglishT::Mark_passives
SEnglishA_to_SEnglishT::Assign_functors
SEnglishA_to_SEnglishT::Mark_infin
SEnglishA_to_SEnglishT::Mark_relclause_heads
SEnglishA_to_SEnglishT::Mark_relclause_coref
SEnglishA_to_SEnglishT::Mark_dsp_root
SEnglishA_to_SEnglishT::Mark_parentheses
SEnglishA_to_SEnglishT::Recompute_deepord
SEnglishA_to_SEnglishT::Assign_nodetype
SEnglishA_to_SEnglishT::Assign_sempos
SEnglishA_to_SEnglishT::Assign_grammatemes
SEnglishA_to_SEnglishT::Detect_formeme
SEnglishA_to_SEnglishT::Detect_voice
SEnglishA_to_SEnglishT::Mark_person_names
```

Transfer

Baseline transfer with the former dictionary

```
SEnglishT_to_TCzechT::Clone_ttree
SEnglishT_to_TCzechT::Baseline_formeme_translation
SEnglishT_to_TCzechT::Baseline_tlemma_translation
# unfortunately there were more cons than pros with
# SEnglishT_to_TCzechT::Improve_translation_by_tree_LM
SEnglishT_to_TCzechT::Fix_transfer_choices
SEnglishT_to_TCzechT::Add_noun_gender
SEnglishT_to_TCzechT::Add_PersPron_below_vfin
SEnglishT_to_TCzechT::Add_verb_aspect
SEnglishT_to_TCzechT::Fix_date_time
SEnglishT_to_TCzechT::Fix_grammatemes_after_transfer
SEnglishT_to_TCzechT::Fix_negation
SEnglishT_to_TCzechT::Fix_verb_reflexivity
SEnglishT_to_TCzechT::Move_genitives_to_postposit
SEnglishT_to_TCzechT::Reverse_number_noun_dependency
SEnglishT_to_TCzechT::Move_dicendi_closer_to_dsp
SEnglishT_to_TCzechT::Override_pp_with_phrase_translation
SEnglishT_to_TCzechT::Recompute_deepord
SEnglishT_to_TCzechT::Find_gram_coref_for_refl_pron
```

Baseline transfer with the new dictionary

```
SEnglishT_to_TCzechT::Clone_ttree
SEnglishT_to_TCzechT::Baseline_formeme_translation
SEnglishT_to_TCzechT::Baseline_tlemma_translation_alt
# unfortunately there were more cons than pros with
# SEnglishT_to_TCzechT::Improve_translation_by_tree_LM
SEnglishT_to_TCzechT::Fix_transfer_choices
SEnglishT_to_TCzechT::Add_noun_gender
SEnglishT_to_TCzechT::Add_PersPron_below_vfin
SEnglishT_to_TCzechT::Add_verb_aspect
SEnglishT_to_TCzechT::Fix_date_time
SEnglishT_to_TCzechT::Fix_grammatemes_after_transfer
SEnglishT_to_TCzechT::Swap_negation
SEnglishT_to_TCzechT::Fix_negation
SEnglishT_to_TCzechT::Fix_verb_reflexivity
SEnglishT_to_TCzechT::Move_genitives_to_postposit
SEnglishT_to_TCzechT::Reverse_number_noun_dependency
SEnglishT_to_TCzechT::Move_dicendi_closer_to_dsp
SEnglishT_to_TCzechT::Override_pp_with_phrase_translation
SEnglishT_to_TCzechT::Recompute_deepord
SEnglishT_to_TCzechT::Find_gram_coref_for_refl_pron
```

HMHM transfer with the former dictionary

```
SEnglishT_to_TCzechT::Clone_ttree
SEnglishT_to_TCzechT::Translate_F_try_rules
SEnglishT_to_TCzechT::Translate_F_add_variants
SEnglishT_to_TCzechT::Translate_F_rerank
SEnglishT_to_TCzechT::Translate_F_fix_by_rules
SEnglishT_to_TCzechT::Translate_L_try_rules
SEnglishT_to_TCzechT::Translate_L_add_variants
SEnglishT_to_TCzechT::Translate_LF_numerals_by_rules
SEnglishT_to_TCzechT::Translate_L_filter_aspect

SEnglishT_to_TCzechT::Cut_variants
    MAX_LEMMA_VARIANTS=6
    MAX_FORMEME_VARIANTS=6
SEnglishT_to_TCzechT::Rehang_to_eff_parents
SEnglishT_to_TCzechT::Translate_LF_tree_Viterbi2
    LM_WEIGHT=0.4
    FORMEME_WEIGHT=1
    BACKWARD_WEIGHT=RE_BW
```

```
SEnglishT_to_TCzechT::Rehang_to_orig_parents
SEnglishT_to_TCzechT::Cut_variants
    MAX_LEMMA_VARIANTS=3
    MAX_FORMEME_VARIANTS=3  # This only reduces the size of final tmt file
SEnglishT_to_TCzechT::Fix_transfer_choices
SEnglishT_to_TCzechT::Add_noun_gender
SEnglishT_to_TCzechT::Add_PersPron_below_vfin
SEnglishT_to_TCzechT::Add_verb_aspect
SEnglishT_to_TCzechT::Fix_date_time
SEnglishT_to_TCzechT::Fix_grammatemes_after_transfer
SEnglishT_to_TCzechT::Fix_negation
#SEnglishT_to_TCzechT::Fix_verb_reflexivity
# depending on previous blocks in scenario, this may cause more errors than fixes
SEnglishT_to_TCzechT::Move_genitives_to_postposit
SEnglishT_to_TCzechT::Reverse_number_noun_dependency
    # this should be done at TCzechT_to_TCzechA
SEnglishT_to_TCzechT::Move_dicendi_closer_to_dsp
SEnglishT_to_TCzechT::Recompute_deepord
    # this won't be needed after all blocks will use
    # $node->shift... instead of $node->set_attr('deepord', $x-0.0001)
SEnglishT_to_TCzechT::Find_gram_coref_for_refl_pron
    # this should be done at SEnglishA_to_SEnglishT
    # (can distinguish him/himself there)
```

HMTM transfer with the new dictionary

```
SEnglishT_to_TCzechT::Clone_ttree
SEnglishT_to_TCzechT::Translate_F_try_rules
SEnglishT_to_TCzechT::Translate_F_add_variants
SEnglishT_to_TCzechT::Translate_F_rerank
SEnglishT_to_TCzechT::Translate_F_fix_by_rules
SEnglishT_to_TCzechT::Translate_L_try_rules
SEnglishT_to_TCzechT::Translate_L_add_variants_new
SEnglishT_to_TCzechT::Translate_LF_numerals_by_rules
SEnglishT_to_TCzechT::Translate_L_filter_aspect

SEnglishT_to_TCzechT::Cut_variants
    MAX_LEMMA_VARIANTS=6
    MAX_FORMEME_VARIANTS=6
SEnglishT_to_TCzechT::Rehang_to_eff_parents
SEnglishT_to_TCzechT::Translate_LF_tree_Viterbi2
    LM_WEIGHT=0.5
    FORMEME_WEIGHT=1
    BACKWARD_WEIGHT=0.5
```

```
SEnglishT_to_TCzechT::Rehang_to_orig_parents
SEnglishT_to_TCzechT::Cut_variants
    MAX_LEMMA_VARIANTS=3
    MAX_FORMEME_VARIANTS=3  # This only reduces the size of final tmt file
SEnglishT_to_TCzechT::Fix_transfer_choices
    IGNORE_NEGATION=yes
SEnglishT_to_TCzechT::Swap_negation
    SWAPLIST=swaplist-new
SEnglishT_to_TCzechT::Add_noun_gender
SEnglishT_to_TCzechT::Add_PersPron_below_vfin
SEnglishT_to_TCzechT::Add_verb_aspect
SEnglishT_to_TCzechT::Fix_date_time
SEnglishT_to_TCzechT::Fix_grammatemes_after_transfer
SEnglishT_to_TCzechT::Fix_negation
#SEnglishT_to_TCzechT::Fix_verb_reflexivity
# depending on previous blocks in scenario, this may cause more errors than fixes
SEnglishT_to_TCzechT::Move_genitives_to_postposit
SEnglishT_to_TCzechT::Reverse_number_noun_dependency
    # this should be done at TCzechT_to_TCzechA
SEnglishT_to_TCzechT::Move_dicendi_closer_to_dsp
SEnglishT_to_TCzechT::Recompute_deepord
    # this won't be needed after all blocks will use
    # $node->shift... instead of $node->set_attr('deepord', $x-0.0001)
SEnglishT_to_TCzechT::Find_gram_coref_for_refl_pron
    # this should be done at SEnglishA_to_SEnglishT
    # (can distinguish him/himself there)
```

Synthesis

```
TCzechT_to_TCzechA::Clone_atree
TCzechT_to_TCzechA::Init_morphcat
TCzechT_to_TCzechA::Impose_rel_pron_agr
TCzechT_to_TCzechA::Impose_subjpred_agr
TCzechT_to_TCzechA::Impose_attr_agr
TCzechT_to_TCzechA::Impose_compl_agr
TCzechT_to_TCzechA::Drop_subj_pers_prons
TCzechT_to_TCzechA::Add_prepositions
TCzechT_to_TCzechA::Add_subconjs
TCzechT_to_TCzechA::Add_reflex_particles
TCzechT_to_TCzechA::Add_auxverb_compound_passive
TCzechT_to_TCzechA::Add_auxverb_modal
TCzechT_to_TCzechA::Add_auxverb_compound_future
TCzechT_to_TCzechA::Add_auxverb_conditional
```

```
TCzechT_to_TCzechA::Add_auxverb_compound_past
TCzechT_to_TCzechA::Resolve_verbs
TCzechT_to_TCzechA::Clause_numbering
TCzechT_to_TCzechA::Move_clitics_to_wackernagel
TCzechT_to_TCzechA::Add_sent_final_punct
TCzechT_to_TCzechA::Add_subord_clause_punct
TCzechT_to_TCzechA::Add_coord_punct
TCzechT_to_TCzechA::Add_parentheses
TCzechT_to_TCzechA::Choose_mlemma_for_PersPron
TCzechT_to_TCzechA::Generate_wordforms
TCzechT_to_TCzechA::Recompute_ordering
TCzechT_to_TCzechA::Delete_superfluous_prepos
TCzechT_to_TCzechA::Vocalize_prepositions
TCzechT_to_TCzechA::Capitalize_named_entities
TCzechT_to_TCzechA::Capitalize_sent_start
TCzechA_to_TCzechW::Concatenate_tokens
```

Appendix E

Sample translation

In this appendix the sample of 100 sentences from the `newstest2009` set used for the final model evaluation is shown. We have included the English source text and the translation obtained using the HMTM based transfer with the new dictionary.

E.1 Source

Prague Stock Market falls to minus by the end of the trading day After a sharp drop in the morning, the Prague Stock Market corrected its losses. Transactions with stocks from the Czech Energy Enterprise (ČEZ) reached nearly half of the regular daily trading. The Prague Stock Market immediately continued its fall from Monday at the beginning of Tuesday's trading, when it dropped by nearly six percent. This time the fall in stocks on Wall Street is responsible for the drop. The reaction of the market to the results of the vote in the American House of Representatives, which refused to support the plan for the stabilization of the financial sector there, has manifested itself here as well. Stocks fall in Asia Stocks in the Asian markets experienced a dramatic drop on Tuesday, even though the indexes ultimately erased a part of the losses during the day. The Hang Seng Index of the Hong Kong Stock Exchange wrote off nearly four percent during the day, but later it erased a part of the losses and reduced the decrease to roughly 2.5 percent. The Hang Seng China Enterprises Index, which follows the movement of Chinese stocks on the stock market in Hong Kong, dropped by 3.8 percent, in Shanghai the markets were closed. Stocks on the market in Sydney lost more than five percent, but ultimately lowered their losses to 4.3 percent. The stock exchange in Taiwan dropped by 3.6 percent according to the local index. "The timing of the bailout action in the USA is uncertain and it will influence financial markets all over the world," remarked the head of the Hong Kong Currency Board, Joseph Yam. Despite the fact that it is a part of China, Hong Kong determines its currency policy separately, that is, without being dependent on the Chinese Central Bank. Hong Kong has interest rates at the same level as the United States. American legislators should quickly return to their negotiations and approve the bill to support the financial system, according to Australian Prime

Minister Kevin Rudd. Otherwise there reputedly looms the threat that other countries will also feel the impacts. American stock bloodbath On Monday the American House of Representatives rejected the plan to support the financial system, into which up to 700 billion dollars (nearly 12 billion Czech crowns) was to be invested. The legislators thus ignored President George Bush's appeal for them to support the plan. According to Bush, the plan would tackle the basic causes of the financial crisis and help stabilize the entire economy. American stocks suffered a bloodbath on Monday and the major stock indexes registered their greatest fall in more than 20 years. The Dow Jones Index dropped by nearly seven percent, having registered a similarly-ranged fall the last time in 1987. The index had dropped even prior to the vote, but as soon as it was revealed that the bill had not passed in the House, the index went into free fall. Congress yields: US government can pump 700 billion dollars into banks The top representatives of the American Congress and George W. Bush's cabinet have agreed upon a broader form of the agreement on financial assistance for the American financial system. The vote on it will take place at the beginning of next week. American legislators made a breakthrough in their talks about the approval of a bailout plan in the form of financial assistance for the American financial system amounting to 700 billion dollars (approximately 12 billion crowns). But all is not won yet. That is, the members of congress have to complete some details of the agreement before they can make the final version of the law public and vote on it. The plan to support the financial system will be discussed in the House of Representatives on Monday. The chair of the Financial Services Committee, Barney Frank, told Reuters this on Sunday. Sources say that the senate could evidently vote on the plan on Wednesday at the soonest. Economists say that the announcement that the bailout plan will be approved should be the first psychological factor significant to the revival of financial markets. Afterward, however, a "sobering up" will take place due to the complicated nature of the mechanisms with which assistance to the markets can be achieved in practice. Paulson: Plan must be effective "We've made great progress. We've resolved our differing opinions on how the package for the stabilization of markets should look," Democrat Nancy Pelosi told Bloomberg. According to her, the final vote could take place as early as Sunday. Representatives of the legislators met with American Finance Minister Henry Paulson Saturday night in order to give the government fund a final form. The fund is meant to purchase unsellable mortgage assets which are pulling financial companies down into heavy losses and are endangering the stability of the entire system. "We're on the edge of a definitive agreement on a plan which will function and which also must be effective on the market. It's necessary to continue in the curative plan, but I think we're there," Paulson said. A signal for Asian trading The global financial crisis is significantly impacting the stock markets, which are dropping sharply. According to Nevada Democratic senator Harry Reid, that is how that legislators are trying to have Congress to reach a definitive agreement as early as on Sunday. Namely, by doing this they want to calm investors prior to trading on the Asian financial markets, which, given their time zones, are the first ones where the decision by Congress could influence Monday's trading. In the meantime, however, it is not yet clear with any certainty when both chambers of the American Congress will vote on the bill, nor whether the

negotiations will not become hindered by some problem. The legislators hope that it will be approved in the next few days. However, the bill will still go through a series of changes. The total amount designated for assistance to the system is to be divided into two parts. The initial 350 billion dollars is to become available as soon as possible, as requested by president George Bush. But Congress can block the release of the remaining amount, in the sequence of a further 100 billion dollars and later, the final 350 billion dollars, if it has the impression that the program is not fulfilling its function. Bush appreciates progress in negotiations Though the president can veto this decision, Congress can override his veto. Even in spite of these changes, the essential idea of the program, to gain finances for the buyout of bad mortgage stocks, the value of which had dropped because hundreds of thousands of Americans were unable to pay off their mortgages, has remained intact. "We've drawn it all up. The House of Representatives should be able to vote on the bill on Sunday and the Senate on Monday," said Republican senator Judd Gregg. Even American president Bush is satisfied with the progress in negotiations. His speaker Tony Fratto declared that Bush was satisfied with Saturday's progress and appreciated the "two-party effort to stabilize our financial markets and save our economy". Constraining golden parachutes The chairman of the Senate Banking Committee, Christopher Dodd, the chief negotiator, said that the legislators had agreed upon the responsibilities for the financial program, protection of the taxpayers against losses, closing of relief packages, and timing of the financial plan. The new law is also meant to constrain the "golden parachutes" for the heads of the companies participating in the program and establishes a committee which will oversee the fund. The committee is to be directed by the Ministry of Finance. "We've worked very hard on this and we've made great progress toward an agreement that will work and that will be useful for all Americans," Paulson said. The plan also includes assistance to homeowners who have problems making their payments. The cabinet should negotiate new payments for the mortgages it buys, with the aim of lowering the monthly payment costs for those in debt and enabling them to keep their houses. Marek: Psychological help According to the chief economist of Patria Finance, David Marek, the proposed plan is a good idea, but its realization will be very difficult. "And this is because it is not known through what mechanism and for what price the problematic assets will be bought. This will determine how the financial sector and the cabinet will divide the losses amongst themselves," Marek told the server iHNed.cz this week. Whether the plan is approved now or at the beginning of next week is not so significant, according to Marek. "More important is for the members of congress to agree this week to support the plan and to approve it as soon as possible. The psychological significance of the agreement that the approval will take place is the most important thing at this time," Marek emphasized. Right-wing populists triumph in Austria, have total of 29 percent According to the first preliminary results of the early parliament elections in Austria have brought about a perceptible weakening of both parties in the present large coalition and a significant boost for the right-wing populist parties. The Austrian People's Party (ÖVP), where the position of the current head, Wilhelm Molterer, is being severely jolted, suffered particularly great losses. Conversely, the campaign leader of the Alliance for the Future

of Austria (BZÖ), Carinthian Governor Jörg Haider, is preparing a triumphant return to national politics. According to the preliminary results, the Social Democratic Party (SPÖ) remains the strongest in the country with 29.8 percent of votes, however, it has lost 5.5 percent of votes since the last elections in 2006. ÖVP, which with its 25.6 percent loses nearly nine percent of votes, fared even worse. These are the worst results of both large parties in Austrian post-war history, and particularly for the People's Party, who urged the early elections, it is literally a catastrophic result. At the beginning of July, when ÖVP left the coalition, the People's Party still had a significant lead on SPÖ in the polls. Voters, however, apparently punished them for letting the government flounder. In the first reactions to the results, there were already speculations about the possible resignation of the party head and current vice-chancellor Wilhelm Molterer. Observers anticipate that this could take place as early as at the extraordinary meeting of the party leadership on Monday. Such a development would certainly simplify the journey toward the increasingly most likely recourse from the election results, that is, the renewal of the large coalition of SPÖ and ÖVP. Given the strengthening of both right-wing populist parties - the Freedom Party (FPÖ) gained a preliminary 18 percent and BZÖ eleven percent of the votes - however, at the same time, the Social Democrats expressed fear of a repetition of the year 1999, when the People's Party agreed with the populists (FPÖ was still united at that time, it broke away from BZÖ in 2005) on a common government, which eventually provoked sanctions from the European Union. Haider, who has already announced that he is prepared to work together with any party and presumes he will return to Vienna to national politics, is evidently banking on this development. The Green Party also got into the parliament, but suffered a slight loss and fell from third to fifth place among Austrian political parties. Evidently none of the other parties exceeded the four percent mark and got into the parliament. Razor's edge battle: MP3 players vs. cell phones. Our advice on how to choose While nearly every cell phone can play MP3 files, no MP3 player can make phone calls. This makes it seem clear that it is better to only buy a phone. In spite of this, there are many reasons to get a separate MP3 player. The choice depends entirely on the manner of use and demands of the future owner. It is not likely that you will get a top-of-the-line expensive record player in order to listen to the newest album by Maxim Turbulenc. First of all such music products not available on LPs, but also, from the qualitative point of view there no objective reason for it. However, if your shelves are crowded with art rock, jazz, or blues vinyl records, you are likely at least dreaming of a record player like that. It's the same with music on trips, that is, compressed music, simply put, MP3 music.

E.2 Translation

Pražský akciový trh spadá, minus, konec obchodního dne. Po ostré kapce do ráno Pražský akciový trh opravil jeho ztráty. Transakce s akciemi Českého energetického podniku z ČEZ dosáhly téměř poloviny pravidelného denního obchodování. Pražský akciový trh okamžitě pokračoval v jeho pádu z pondělí na začátek obchodování úterý, kdy kleslo téměř šest procenta. Tato doba pádu akcií na Wall street je odpovědná za pok-

les. Reakce trhu na výsledky hlasování v Americké sněmovně zástupců, které odmítlo podporovat plán stabilizace finančního sektoru tam, projevovala se to tady stejně tak. Akcie spadají do Asie. Akcie na asijských trzích zažily dramatický propad v úterý, indexy nakonec vymazaly část ztrát během dne. Seng index momentu Hongkongské akciové burzy kongy odepsal téměř čtyři procentům během dne, ale pozdě vymazal část ztrát a snížil pokles na zhruba 2,5 procenta. Seng čínský Enterprises index momentu, který sleduje pohyb čínských akcií na akciovém trhu v Hong konze, klesl 3,8 procenta, v Šanghaj trhy byly uzavřeny. Akcie na trhu v Sydney ztratily více než pět procent, ale nakonec sníží jejich ztráty na 4,3 procenta. Výměna akcie v Tchaj poklesla 3,6 procenta s místním indexem. "Načasování" bailout akce v USA je nejisté a zapůsobí na finanční trhy na všechno na svět," prohodil hlavu Hongkongské měnové rady kongy Joseph Yam. Skutečnost, že to je část Číny, Hong konga určuje jeho měnovou politiku samostatně, to je, je závislá na Čínské centrální bance. Hong konga má úrokové sazby na stejnou úroveň jako Spojené státy. Američtí zákonodárci má, rychle vrátí k jejich jednáním a schválí zákon podporovat finanční systém, australian premiéra ministra Kevin Rudd. Jinak nádeník rýsuje hrozbu, že jiné země také pocítí dopady. Americké akciové krveprolití. V pondělí Americká sněmovna zástupců zamítla plán podporovat finanční systém, na který 700 až miliardy dolarů téměř 12 miliardy českých korunních bylo, jsou investovány. Zákonodárci tedy ignorovali odvolání prezidenta George Bushe podporovat plán. Podle Bushe plán dal si, řeší základní příčiny finanční krize a pomoc stabilizovat celou ekonomiku. Americké akcie utrpěly masakr v pondělí a hlavní indexy burzy zaregistrovaly jejich největší pád ve více než 20 letech. Dow Jones index klesl téměř sedm procenta, registroval similarly-ranged pád poslední dobu v roce 1987. Index poklesl dokonce před hlasováním, ale bylo odhaleno, že zákon neprojít Sněmovnou index šel do volného pádu. Výnosy Kongresu: Naše vláda může napumpovat 700 miliardy dolarů do bank. Vysocí zástupci Americana sjezdu a George W kabinetu Bushe dohodli se na širší formě dohody o finanční pomoci o americkém finančním systému. Hlas o tom přijme místo na začátku příštího týdne. Američtí zákonodárci učinili průlom ve svých rozhovorech o schválení bailout plánu v podobě finanční pomoci amerického finančního systému, činí 700 miliardy dolarů přibližně 12 miliardy korunních. Ale všechno není vyhráno ještě. To je, členy Kongresu musí dokončit některé podrobnosti dohody, mohou udělat konečnou verzi zákona veřejnosti a hlasu na tom. Plán podporovat finanční systém bude projednán ve Sněmovně zástupců v pondělí. Předseda Finančního výboru služeb Barney Franku řekl Reuters to v neděli. Zdroje tvrdí, že Senát zřejmě mohl by hlasovat o plánu ve středu na keats. Ekonomové říkají, že oznámení, že bailout plán bude schválen, mělo by být první psychologický faktor významný oživení finančních trhů. Nato ovšem "vystřízliví" will, vezme místo důvod složité povahy mechanismů s jehož pomocí trhům, mohou být dosaženy v praxi. Paulsen: plán musí být účinná. "Jsme učinili velký pokrok. Jsme vyřešili své odlišné názory, jak balíček pro stabilizaci trhů měl by vypadat," demokrat Nancy Pelosi řekla Bloomberg. Podle ní konečné hlasování mohlo by brát místo stejně brzy jako neděle. Zástupci zákonodárců sešli se s Američan finanční ministra Henry Paulson sobotní noc, dá vládnímu fondu konečnou podobu. Fond je znamenán nakupovat unsellable hypotéky aktiva, který táhne finanční

společnosti dolů do těžkých ztrát a ohrožuje stabilitu celého systému. Jsme na okraji konečné dohody o plánu, který bude fungovat a který také bude muset být účinná na trhu. To je nutné pokračovat v léčebném plánu, ale myslím, že jsme tam," Paulson řekla. Signál pro asijské obchodování. Globální finanční krize výrazně ovlivňuje akciové trhy, které klesají prudce. Podle Nevada demokratického senátora Harryho Reida, který je, jak podle zákonodárců, jsou, při pokusu Kongres muset dosáhnout konečnou dohodu tak brzy v neděli. Totiž, dělá to, chtějí, aby uklidnila investory před obchodováním na asijských finančních trzích, které daly jejich časová pásma, jsou první ti, kde rozhodnutí Kongresu mohlo by ovlivnit obchodování pondělí. Zatím ovšem to není ještě jasné s jakoukoli jistotou, kdy, obojí komory Amerického kongresu budou hlasovat o zákoně ani zda jednání nebudou stát se, brání nějaký problém. Zákonodárci doufají, že bude schváleno v příštích několika dnech. Ovšem zákon stále poběží řadu změn. Celková částka určená pro pomoc systému je, je rozdělen na dvě části. Původní 350 miliardy dolary jsou být se dostupných možných požádat prezidenta George Bushe. Ale Kongres může zablokovat uvolnění zbývající částky v sekvenci dalších 100 miliard dolarů a pozdě, konečných, 350 miliardy, dolarů, má dojem, že program neplní jeho funkci. Bush oceňuje pokrok v jednáních. Prezident může vetovat toto rozhodnutí, Kongres může obejít své veto. Dokonce přes tyto změny základní myšlenka programu získat finance na odkup špatných hypotečních akcií hodnota, které vynechal, protože stovky tisíců američan byly neschopných splatit jejich hypotéky, zůstávala neporušená. "Jsme vypracovali to všechno. Dům zástupců měl by být schopen hlasovat o zákoně v neděli a Senátu v pondělí," řeklo republikánského senátora Judda Gregga. Dokonce americký prezident Bush je uspokojen s pokrokem v jednáních. Jeho reproduktor Tony Fratto prohlásila, že Bush byl uspokojen s pokrokem soboty a ocenila "two-party snahu stabilizovat naše finanční trhy a zachránit naši ekonomiku." Omezuje zlaté padáky. Předseda Bankovního výboru senátu Christopher Dodd hlavního vyjednavače řekl, že zákonodárci dohodli odpovědnosti za finanční program ochrany poplatníků před ztrátami uzavření humanitárních balíčků a načasování finančního plánu. Nový zákon, také je znamenán omezovat "zlaté padáky" hlavy společností zúčastněných programu a zřídí výbor, který bude dohlížet na fond. Výbor je být namířený Ministerstvem financí. "Jsme pracovali velmi tvrdě na tom a jsme učinili velký pokrok k dohodě, která bude pracovat a že to bude užitečné, všichni američan" Paulson říkali. Plán také zahrnuje pomoc domovníci, kdo mají problémy, provedou jejich platby. Kabinet měl by vyjednat nové platby za hypotéky, kupuje z cíle snížit měsíční náklady na platbu na ty v dluhu a umožnit udržet jejich domy. Marka: psychologická pomoc. Hlavní ekonom Patria financí David Marek navrhovaný plán, je, dobrý nápad, ale jeho realizace bude velmi obtížným. "A to je být známý jaký mechanismus a za jakou cenu problematická aktiva nebude koupeno. Určí, jak finanční sektor a kabinet rozdělí, Ztráty mezi sebou" Marek řekly, že server ihn, cz tento týden. Plán není schválen nyní nebo na začátku příštího týdne, tak významnou, podle Marka. "Důležitější je pro členy Kongresu, souhlasit tento týden, že podporuje plán a schválit to možné. Psychologický význam dohody, schválení bude trvat místo, je nejdůležitější věc v této době," Marek zdůrazňoval. Right-wing populisté triumfujou v Rakousko, má součet 29 procent. První předběžné výsledky brzkých voleb v

parlamentu v Rakousko musely přinést znatelné oslabení obojích stran současné velké koalice a významné stoličky pro right-wing populistické strany. Strana rakouských lid ÖVP, kde pozice současné hlavy Wilhelm Molterer, je těžce, otřásla, utrpěl zejména velké ztráty. Naopak vůdce kampaně spojenectví pro budoucnost (Rakousko) (BZÖ) Carinthian guvernéra Jörg Haider připravovat triumfální návrat do národních politik. (Předběžné výsledky Sociální demokratická strana Spö zůstávají nejsilnějších v zemi s 29,8 procenty hlasů ovšem, ztratilo 5,5 procenta hlasů od minulých voleb v roce 2006.) ÖVP, který svého 25,6 procenta ztrácí téměř devět procenta hlasů, vedl si dokonce hůř. Ti jsou, nejhorší výsledky obojích velkých stran rakouské post-war historie a zejména pro Stranu lidí, která vyzývala brzké volby, to jsou doslova katastrofální výsledek. Na začátku července, kdy ÖVP opustil koalici, Strana lidí stále měla významné olovo na Spö v průzkumech. Voliči ovšem zřejmě potrestali je, že dají vládní ubrousek. V prvních reakcích na výsledky byly už spekulace o možné rezignaci hlavy strany a aktuální vice-chancellor Wilhelm Molterer. Pozorovatelé předvídají, mohlo by brát místo tak brzy na mimořádném zasedání vedení strany v pondělí. Takový vývoj rozhodně by zjednodušil cestu k stále nejpravděpodobnějšímu postihu výsledků volby, to je obnovení velké koalice Spö a ÖVP. (Posílení obojích right-wing populistických stran - Strany svobody Fpö přibralo předběžných 18 procent a BZÖ jedenácti procent hlasů - ovšem ve stejné době, sociální demokraté vyjádřili obavu z opakování roku roku 1999, kdy Strana lidí souhlasila s populisty, že, když Fpö stále byl sjednocena v té době, zlomilo od BZÖ v roce 2005 při společné vládě, která nakonec vyprovokovala sankce Evropské unie.) Haider, který již ohlásil, je připraven, aby pracoval spolu s každou stranou a předpokládá, že vrátí do Vídeň národním politikám, zřejmě sází na tento vývoj. Zelená strana také dostala se do parlamentu, ale utrpěla mírnou ztrátu a klesla z třetí na páté místo mezi rakouskými politickými stranami. Zřejmě žádný z druhých stran nepřekročil čtyři známku procenta a dostane se do parlamentu. Kraj battle břitvy: MP3 hráč v mobilní telefon. Naše rada jak zvolit. Téměř každý mobilní telefon moci hrát MP3 soubory žádný MP3 hráč nemůže zvládnout telefonní hovory. Dělá, zdá se jasné, že to je lepší, pouze koupí telefon. To je mnoho důvodů získat samostatného MP3 hráče. Volba závisí zcela na způsobu používání a poptávek budoucího vlastníka. To není pravděpodobné, že dostanete top-of-the-line drahého hráče záznamu poslouchat nejnovější album Maximou Turbulenc. Nejprve všech takových hudebních produktů ne dostupných na LPs, ale rovněž z kvalitativního pointu zobrazení tady žádného objektivního důvodu toho. Pokud ovšem vaše police jsou přeplněny uměleckým kamenem, jazzem nebo bluesovými vinylovými deskami, jste pravděpodobné, alespoň sníš o hráči záznamu. Současně s hudbou výlety je komprimováno hudbu, prostě dá, MP3 hudební.