The Prague Bulletin of Mathematical Linguistics
NUMBER 92 DECEMBER 2009 115–134

# Improving English-Czech Tectogrammatical MT

Martin Popel, Zdeněk Žabokrtský

**Abstract**

The present paper summarizes our recent results concerning English-Czech Machine Translation implemented in the TectoMT framework. The system uses tectogrammatical trees as the transfer medium. A detailed analysis of errors made by the previous version of the system (considered as the baseline) is presented first. Then several improvements of the system are described that led to better translation quality in terms of BLEU and NIST scores. The biggest performance gain comes from applying Hidden Tree Markov Model in the transfer phase, which is a novel technique in the field of Machine Translation.

## 1. Introduction

We report on a work in progress on developing English-Czech machine translation (MT) system called TectoMT.[1] This system participated at the Workshop on Statistical Machine Translation (WMT) in 2008 and 2009 (Žabokrtský et al., 2008; Bojar et al., 2009). The translation is carried out in three phases: analysis, transfer and synthesis. Similarly to Bojar et al. (2008a), the transfer phase implemented in TectoMT uses tectogrammatical trees and exploits the annotation scheme of the Prague Dependency Treebank, but (unlike in the cited work) the transfer does not use Synchronous Tree Substitution Grammars.

In Section 2, we shortly describe our baseline system. In order to identify its most prominent errors, their types and sources, we have manually annotated a sample of 250 sentences; the resulting error analysis is presented in Section 3. Modifications of our baseline system and their evaluation are described in Section 4. One of the most important modifications – the introduction of Hidden Markov Tree Models to the transfer phase – is explained in Section 5.

[1] `http://ufal.mff.cuni.cz/tectomt/`

Please cite this article as: Martin Popel, Zdeněk Žabokrtský, Improving English-Czech Tectogrammatical MT. The Prague Bulletin of Mathematical Linguistics No. 92, 2009, 115–134.

## 2. Baseline system

The TectoMT version which participated in WMT 2009 is used here as the baseline system. In this version, the translation process consists of about 80 steps implemented in so-called *blocks* (basic TectoMT processing units). We give here only a brief overview.

### 2.1. Analysis

Each sentence is tokenized (roughly according to the Penn Treebank conventions), tagged by the English version of the Morce tagger (Spoustová et al., 2007), and lemmatized in order to obtain the morphological layer (m-layer). Maximum Spanning Tree dependency parser (McDonald et al., 2005) is applied to create analytical trees (a-trees). These are then converted to the tectogrammatical ones using a sequence of heuristic blocks: Functional words (such as prepositions, subordinating conjunctions, articles etc.) are removed. Only morphologically indispensable categories (called *grammatemes*) are left with the tectogrammatical nodes (t-nodes). The information about the original syntactic form is stored in attributes called *formemes*.[2] Several other attributes are filled (e.g. functors, coreference links, named entity types).

### 2.2. Transfer

First, the topology of target-side t-trees is copied from source-side t-trees. Probabilistic dictionaries provide n-best lists of lemmas and formemes. In the baseline scenario, formemes are translated independently for every node as the most probable variant from the n-best list. Consequently, lemmas are translated as the most probable variant that is compatible with the already chosen formeme. The compatibility is ensured by a set of rules. Additional rule-based blocks are used to translate other t-layer attributes (grammatemes) and to change topology and word order where needed.

### 2.3. Synthesis

In this phase Czech analytical trees are created from the tectogrammatical ones (auxiliary nodes are added), but the process of synthesis continuously goes on (morphological categories are filled, word forms are generated), so that in the last block, the sentence is generated by simply flattening the tree and concatenating the word forms.

[2] Formemes are not used in Prague Dependency Treebank, they were introduced to TectoMT with regards to the needs of MT (Žabokrtský et al., 2008). Formemes cannot be considered as a genuine component of the tectogrammatical layer of language description, but they facilitate formalizing the relation between tectogrammatics and surface syntax and morphology. Examples of formemes are: `n:subj` – semantic noun in subject position, `n:for+X` – semantic noun with preposition *for*, `v:because+fin` – semantic verb as a head of subordinating finite clause introduced by *because*, `v:without+ger` – semantic verb as a gerund after *without*, `adj:attr` – semantic adjective in attributive position.

## 3. Error annotations and analysis

Manual analysis of translation errors is expensive and time-demanding, but it can identify types and sources of errors. This knowledge is very helpful for developers of MT systems, that perform transfer on some level of abstraction that is higher than simple phrase-to-phrase. There are many papers on manual evaluation of MT errors, (e.g. Koehn and Monz, 2006), but they are mostly limited to scoring *fluency* and *adequacy*. Some papers (Hopkins and Kuhn, 2007) use manual analysis based on some form of *edit distance*, i.e. the number of editing steps (of various types) needed to transform the system output into an acceptable translation. One of the most detailed manual analysis frameworks is the Error Classification Scheme described in Vilar et al. (2006), which classifies errors into a hierarchical structure.

Our proposed error analysis framework is similar to that of Vilar et al. (2006), but instead of three hierarchical properties of errors (*type, subtype* and *sub-subtype*) we have five properties: *seriousness, type, subtype, source* and *circumstances*. Errors are marked in text by *error markers* which the annotator simply inserts in front of relevant words. If needed, one word may have more than one error marker. Every error marker describes all the five properties of an error. Details about the error analysis framework including several examples of annotated text can be found in Popel (2009).

| Source | | Description | #errors |
|---|---|---|---:|
| Analysis | `tok` | tokenization errors | 16 |
| | `tagger` | PoS tagging errors | 37 |
| | `lem` | lemmatization errors | 1 |
| | `parser` | errors associated with parsing and related tasks (building a-layer from m-layer) | 300 |
| | `tecto` | tecto-analysis errors (building t-layer from a-layer) | 68 |
| Transfer | `noniso` | errors caused by the assumption of t-tree isomorphism (which is currently required in the TectoMT translation) | 109 |
| | `other` | other errors associated with the transfer (translation of lemmas, formemes, grammatemes, noun gender assignment,...) | 845 |
| | `syn` | synthesis errors (generation of text from the target t-layer) | 42 |
| | ? | source unknown | 45 |
| | | total | 1463 |

*Table 1. Distribution of translation errors with respect to their sources*

| **Circumstance** | **Description** – errors associated with … | **#errors** |
|---|---|---|
| `ne` | named entity | 104 |
| `num` | numbers (numerals) | 40 |
| `coord` | coordination or apposition | 117 |

*Table 2. Distribution of translation errors with respect to their circumstances*

The first author of this paper annotated 250 sentences. Tables 1 – 3 show numbers of occurrences of errors for categories *source, circumstances, type* and *subtype*.[3] As expected, most errors lie in the transfer phase. Only 8% of errors are caused by the unfulfilled presumption of isomorphic t-trees, whereas 56% are other transfer errors that could be repaired within the node-to-node transfer paradigm.[4] Another notable source of errors is parsing – 21%. We have found that 39% of these parsing errors are associated with coordinations. Also other observations indicate that the parsing of coordinations is a significant problem in TectoMT: There were 89 coordinations in the test data and more than half of them is parsed incorrectly which results in 1.13 serious errors per coordination on average.

The most common type of error is a wrong choice of lemma (`lex` = 37%), followed by a wrong choice of formeme (`form` = 33%) and grammateme (`gram` = 10%). Several subtypes of `lex` were classified (compound words, errors associated with named entities or reflexivity of lemmas), but most `lex` errors remain unclassified. We have not carried out any subclassification of `form` errors except registering problems with the Czech formeme *v:že+fin*. Among subtypes of `gram`, the most problematic one is the choice of correct gender[5] and number.

[3] We have also distinguished between serious and minor errors, but for brevity, this last category (*seriousness*) is not shown in the tables. Errors with types `punct`, `order` and `case` were mostly minor, other types were mostly serious.

[4] This finding is for us – TectoMT developers – very important. Of course, we are aware of the cases that cannot be translated within the node-to-node paradigm (e.g. *take part* → *účastnit se, make X public* → *zveřejnit X*) and we plan to solve them in TectoMT in future. However, those 8% is a relatively small number and thus we primarily focus on more frequent types of errors.

[5] It is well known that when translating from English to Czech, gender must be sometimes guessed from context, since English does not indicate gender for verbs, but Czech does.

| **Type**<br>**Subtype** | **Description** | **#**<br>**errors** |
|---|---|---|
| `lex` | wrong lemma | **544** |
| `asp` | wrong aspect of a verb | 6 |
| `se` | wrong reflexivity, e.g. t-lemma *stát_se* instead of *stát* | 15 |
| `neT` | named entity translated, but should remain unchanged | 11 |
| `neU` | named entity unchanged, but should be translated, because the original form is not acceptable in the target language | 4 |
| `neX` | assumed named entity unchanged, but should be translated, because it is not really a named entity (*Bill was approved.*) | 8 |
| `com` | unchanged word due to an unprocessed compound word | 13 |
| `unk` | unchanged (possibly missing in the dictionary) word other than `neU`, `neX` and `com` | 6 |
| `other` | default value when no subtype is specified | 481 |
| `form` | wrong formeme | **481** |
| `ze` | formeme v:že+fin instead of v:rc or v:fin | 39 |
| `other` | default value when no subtype is specified | 442 |
| `gram` | wrong grammateme and related errors | **151** |
| `gender` | wrong grammateme of gender (feminine, neuter, masculine animate, masculine inanimate) | 41 |
| `person` | wrong grammateme of person (first, second, third) | 3 |
| `number` | wrong grammateme of number (singular, plural) except cases classified as `numberU` (see below) | 26 |
| `tense` | wrong grammateme of tense (simultaneous, preceding, subsequent) | 5 |
| `mod` | wrong verbal, deontic, dispositional or sentence modality | 18 |
| `deg` | degree of comparison (positive, comparative, superlative) | 4 |
| `neg` | negation (affirmative, negative) | 19 |
| `svuj` | switched m-lemma *svůj* with *jeho*, *její*, … | 17 |
| `numberU` | number unchanged, but should be changed e.g. *Ministry of Finance*(sg) → *Ministerstvo financí*(pl) | 8 |
| `other` | default value when no subtype is specified | 10 |
| `phrase` | phrases, idioms, deep syntactic structures that cannot be translated node-to-node. | **81** |
| `miss` | missing words that are not covered by the types above | **19** |
| `extra` | superfluous words that are not covered by the types above | **36** |
| `punct` | punctuation errors | **64** |
| `brack` | missing, superfluous or displaced brackets | 24 |
| `other` | default value when no subtype is specified | 40 |
| `order` | wrong word order (except cases classified as `punct`) | **64** |
| `case` | switched upper/lower case | **23** |

*Table 3. Distribution of translation errors with respect to their types and subtypes*

## 4. Modifications and their evaluation

We have implemented several modifications to our system in order to improve the translation quality. We present here an overview of the most important modifications.

### 4.1. Analysis

- We have done slight modifications of the tokenization, so for example *3rd* is not split into two tokens anymore.
- We have developed a new implementation of the lemmatization – it fixes some errors made by the original implementation and it is more than 70 time faster.
- We have improved the parsing in the following two ways without actually changing the parsing algorithm or its features:
  We have implemented rule-based blocks that fix some frequent "mistakes" made by the parser. Some of these "mistakes" are real errors, but some are caused by different parsing guidelines concerning for example auxiliary verbs or multi-word prepositions.
  We noticed that in the analysed sample, there are 22 sentences with parentheses and only 2 of them are parsed correctly. Sometimes the parenthesis is incorrectly divided and each part attached to another parent. Sometimes there are parsing errors also in the rest of the sentence, but these errors disappear, when we try to parse the sentence without the parenthesis. By parsing the parenthesis and the rest of the sentence independently we ensure that the parenthesis remains in its own subtree, which is then attached to the main sentence tree.
- *Analytical function* is the key attribute of the a-layer. It specifies the type of dependency relation of a node to its governing node. The baseline system used analytical functions only to mark coordinations and prepositions. We have added a block that recognizes also other types of dependencies, e.g. subject, object, predicate, adverbial, attribute, auxiliary verb, article. As there are no guidelines for English analytical functions yet, we had to decide how to annotate phenomena without any Czech equivalent (articles, phrasal verb particles, infinitive marker *to*, negation *not*). For details see Popel (2009).
- We have implemented a new procedure that builds the t-layer from the a-layer. It exploits analytical functions, which makes the procedure more clear. It deals with special cases that were not solved properly in the baseline implementation. We have aimed at a robust implementation that can handle also some cases with inaccurate parsing. Also, we have aimed at a modular implementation – the procedure is divided into five blocks and three of them are language independent.

### 4.2. Transfer

Our new design of the transfer phase is more modular. We have created 10 new blocks which can be combined in various translation scenarios.

- rule-based blocks that translate some special phenomena, e.g. ordinal numerals (*1st, 32nd, 999th*) can be translated by a simple rule (to *1., 32., 999.*),
- blocks that save all translation variants proposed by the dictionaries[6] to the attributes of nodes,
- blocks that rerank these variants using either more detailed models (e.g. valency formeme translation dictionary) or rules (e.g. the rule that filters out verbal lemmas whose aspect is incompatible with the given context),
- a block that selects the optimal combination of lemmas and formemes for every node using Hidden Tree Markov Model (HMTM). This is discussed in detail in Section 5.

### 4.3. Synthesis

- Word forms are generated according to lemmas and morphological categories. In theory, the word form should be fully specified by the lemma and morphological tag and there is a deterministic Czech word form generator suited for the task (Hajič, 2004). In practice, the tags are "underspecified", because they are generated from the t-layer that was translated from English. Some categories are not known and must be guessed.
  We have created a module that includes a subroutine for generating all forms of a given lemma whose tags match a given regular expression. The word forms are sorted according to their frequency. The model was trained on the corpus SYN (with 500 million words) of Czech National Corpus.[7]
- Commas (more precisely, a-nodes corresponding to commas) are added to boundaries of finite clauses. We have refined the rules for special cases such as quotations. We have also created a new block that coindexes all nodes belonging to the same finite clause.

### 4.4. Evaluation

Aside from evaluating the total difference of BLEU score between the baseline and our new modified version of TectoMT (see Table 4), we want to evaluate also the effect of each modification separately. However, many of the modified blocks would not work with the baseline system, because we have meanwhile added some functionality also to TectoMT internals. Therefore, we have chosen the opposite way – we take the new modified system, substitute one or more blocks with their baseline equivalent

[6] We use a probabilistic dictionary of lemmas (Rouš, 2009) created from the parallel corpus CzEng (Bojar et al., 2008b) and other sources as a replacement for the older PCEDT dictionary (Cuřín et al., 2004). For the translation of formemes we use the so-called valency formeme translation dictionary, which models the probability of target formeme given source formeme and source parent's lemma, and simple formeme-to-formeme dictionary as a fallback.

[7] `http://www.korpus.cz`

| system | BLEU | NIST |
|---|---|---|
| baseline | 0.0659 | 3.9735 |
| modified | 0.0981 | 4.7157 |

*Table 4. BLEU&NIST evaluation of the new system*

(called "original implementation") and we measure the impairment caused by the absence of the modification in question. This value can be loosely interpreted as an improvement caused by the modification, but we must be careful, because there may be "interferences" between some blocks.

We divided the evaluation data of WMT 2009 Shared Task (news-test2009) into two parts:

- First 250 sentences were used for the manual annotation of errors of the baseline implementation (as presented in Section 3).
- The rest (2 777 sentences) is our test set. Tables 4 and 5 are evaluated on this test set.

| **Modification** | diff (BLEU) | diff (NIST) |
|---|---:|---:|
| **original analysis** | **0.0078** | **0.1363** |
| —original tokenization | 0.0008 | 0.0105 |
| —original lemmatization | 0.0006 | 0.0294 |
| —original parsing | 0.0072 | 0.3006 |
| —original building of t-layer | 0.0053 | 0.1024 |
| **original transfer** | **0.0171** | **0.4189** |
| —without HMTM | 0.0130 | 0.2483 |
| **original synthesis** | **0.0031** | **0.0621** |
| original quotation marks | 0.0085 | 0.1757 |
| **all above together** | **0.0322** | **0.7422** |

*Table 5. Modifications of analysis, transfer and synthesis*

**Note on BLEU&NIST scores reliability**

Correct opening and closing quotation marks are in Czech „ and ". These symbols are produced by TectoMT as a translation of English " and ". However, reference translations in WMT09 training and test data use plain ASCII quotes ("). Statistical MT systems trained on such data produce of course also ASCII quotes. For the purpose of a fair comparison with those systems, we have created a simple block `Ascii_quotes` that converts correct Czech directional quotes to incorrect ASCII ones. We were surprised how a large "improvement" can be achieved with this block on our test data –

0.0085 BLEU (0.1757 NIST). This fact only confirms that neither BLEU nor NIST can be used as the ultimate measure for comparing two MT systems of different types.

## 5. Hidden Markov Tree Models

### 5.1. Motivation

**Most errors are caused by the transfer of lemmas and formemes**
In the manual annotation of translation errors we have discovered that more than half of all errors are caused by the transfer phase and 92% of these errors are wrong lemmas and wrong formemes. The choice of correct lemma and formeme is of course a very difficult task and the quality of translation depends heavily on the quality of the dictionaries used. However, even with an ideal dictionary many errors will occur if we just select the most probable variant for each node without considering the context.

**Two meanings of the word *speaker***
For example, word *speaker* with the sense *loudspeaker* should be translated as *reproduktor* and according to the lemma dictionary used in our scenario the translation probability is $\mathrm{P}(\textit{reproduktor}\,|\,\textit{speaker}) = 0.45$. When the sense is *spokesperson*, the correct translation is *mluvčí* and $\mathrm{P}(\textit{mluvčí}\,|\,\textit{speaker}) = 0.26$. Perhaps, there were more texts about loudspeakers than texts about spokespersons in the CzEng parallel corpus upon which the dictionary is based. The baseline system translates every word *speaker* as *reproduktor*, so we encounter errors in phrases like *speaker of the Ministry of Transport*.

**Linear context and tree context**
In phrase-based MT, the context used to select the best translation of a word is linear – basically, the context is a phrase, i.e. a string of surrounding words. There are some experiments with "phrases with gaps" (Simard et al., 2005), but in most systems a phrase is defined as a contiguous string of words (not necessarily forming a phrase in a linguistic sense).

We believe that it is more appropriate to use a local tree context, i.e. the children and the parent of a given node. Not only that it is appropriate according to linguistic intuition, but it should help us to face the data sparseness.

For illustration, consider the before-mentioned example with the phrase *speaker of the Ministry of Transport*. Human translators recognize from semantics that the *speaker* is a human being (not a loudspeaker) and translate it as *mluvčí*. Phrase-based MT systems can learn the whole phrase or possibly just the phrase *speaker of the Ministry*, but they must also learn phrases like *speaker of the Chinese Ministry*, *speaker for the Foreign*

*Ministry, speaker for the Indian External Affairs Ministry* etc. in order to translate them correctly.[8]

When using the local tree context, we can for example learn that *speaker* should be translated as *mluvčí* if it has a child node with the lemma ministry. This way we cover all the before-mentioned phrases including the unseen ones. Another knowledge learned from a parallel dependency treebank may be that *speaker* should be translated as *mluvčí* if its parent node has the lemma name (e.g. in phrases *speaker's name, name of the next speaker*) or that *speaker* should be translated as *reproduktor* if its parent node has the lemma buy (e.g. in a phrase *buy an expensive speaker*).

**How to learn, represent and use tree context?**

The obvious question is how can we learn, represent and use such knowledge. The preceding paragraph formulates the knowledge in a form of rules. Although this approach could be used in MT (rules can be automatically learned from the treebank), it is difficult to combine it with probabilistic methods. We have decided to represent the knowledge in a form of a model that describes the probability of a node given its parent node. More precisely, we model the probability of a lemma and formeme of the dependent node given a lemma and formeme of the governing node.

The model can be learned from a treebank using maximum likelihood estimate, but similarly to traditional (linear) language models it is necessary to smooth the probabilities and there are many possible ways how to perform the smoothing.

**Tree context: bilingual or target-language?**

The probabilistic model introduced in the previous paragraph is a monolingual tree model and can be learned from a target-language treebank (Czech in our case). With the availability of parallel treebanks we can develop also "bilingual tree models". An example of bilingual tree model is the valency formeme translation dictionary. It specifies the probability of formeme of the target-side node given formeme of the source node and lemma of the source node's parent.

Ideally, we would like to use more complex bilingual tree model that defines also target-side lemmas and that is conditioned also by other attributes (lemma of the source node, lemmas of its children etc.). This complex model would supersede both

[8] The example if oversimplified. First, in phrase-based MT systems, it is the target-language model that should cover such long phrases, so it would be more accurate to present Czech translations of the phrases. Second, we suppose that the hypothetical phrase-based system is trained on the same parallel corpus as our dictionary, so $\mathrm{P}(\textit{reproduktor}\,|\,\textit{speaker}) > \mathrm{P}(\textit{mluvčí}\,|\,\textit{speaker})$ and similarly for backward probabilities $\mathrm{P}(\textit{speaker}\,|\,\textit{reproduktor}) > \mathrm{P}(\textit{speaker}\,|\,\textit{mluvčí})$. Otherwise, there would be no need for the language model to cover the phrases, if the translation model itself would choose the correct translation. Third, since the phrases learned by phrase-based MT systems are usually not constrained to linguistically adequate constituency phrases, it is possible that the system will learn that *speaker of the* should be translated as *mluvčí*. However, there are plenty of more relevant examples of long-distance dependencies that are not covered even by 6-gram or 7-gram language models.

formeme and lemma dictionaries as well as the target-language tree model. However, we do not have enough parallel data to reliably train such a model. Since the amount of monolingual training data is much larger, we try to exploit it as much as possible.

**First attempts at using tree context**

In the baseline translation of lemmas and formemes, the only usage of tree context was in the valency formeme translation dictionary. Moreover, lemmas and formemes were translated almost independently – there was only a rule to check for compatibility of a lemma with a formeme, but no probabilistic model describing their joint or conditional probability. In other words, the target-language tree model was not used in the baseline implementation.

One of the first attempts at exploiting the target-language tree model performed a top-down depth-first traversal through the t-tree translated by the baseline system. Its main idea was to choose the best lemma and formeme according to a loglinear combination of three models: translation probability of lemma, translation probability of formeme and target-language tree model created by Václav Novák. The main difference from HMTM and the tree-modified Viterbi algorithm presented in this paper is that the top-down traversal allows only local optimization based on the parent node (but no children nodes), whereas the tree-modified Viterbi algorithm searches for the global maximum.

**Why do we need Hidden Markov Tree Models?**

The apparent weak point of the before-mentioned top-down traversal occurs when the correct lemma or formeme can be determined only from the children rather than from the parent (e.g. *He is a speaker of the ministry* versus *It is an expensive speaker*). Of course, if we use a similar algorithm with bottom-up traversal, these cases will be handled correctly, but errors will be introduced in the opposite cases – when the correct lemma or formeme can be determined only from the parent, but not from children (e.g. *according to the speaker* versus *buy a speaker*).

Not only that both the types of cases (parent/children are important for translation) are frequent, but sometimes we need to know the parent as well as the children to choose the correct translation. The child-parent dependencies are chained in the tree, so we need to find the combination of lemmas and formemes that results in the maximal global probability of the whole tree. Hidden Markov Tree Models provide a theoretical background for the tree-modified Viterbi algorithm, which can efficiently find the global maximum.

### 5.2. Description of HMTM

**Related work**
Hidden Markov Models (HMM, see Chapter 9 in Manning and Schütze (1999))[9] belong to the most successful techniques in Computational Linguistics. There are many modifications of HMM: arc-emission versus state-emission, epsilon-emission, HMM with Gaussian distribution of emission function etc. Hierarchical Hidden Markov Models, which are used for Information Extraction (Skounakis et al., 2003), make use of tree structures, but they still primarily work with linearly organized observations/states.

Hidden Markov Tree Models (HMTM) were introduced by Crouse et al. (1998), and used in applications such as image segmentation, signal classification, denoising and image document categorization. More information about HMTM can be found in Diligenti et al. (2003) and in Durand et al. (2004). The latter article contains also a detailed explanation of the tree-modified Viterbi algorithm. Parts of this Section are based on Žabokrtský and Popel (2009), where HMTM are introduced for dependency-based MT, and on Popel (2009).

**Formal definition**
Suppose that

- $V = \{1, \dots, |V|\}$ is a set of tree nodes, $r \in V$ is the root node and $\rho : V \setminus \{r\} \rightarrow V$ is a function determining the parent node of each non-root node.
- $\mathbf{X} = (X_1, \dots, X_{|V|})$ is a sequence of random variables taking values from a state space $S$. Random variable $X_v$ is understood as a *hidden state* of the node $v$ and $P(X_v|X_{\rho(v)})$ is called *transition probability*.
- $\mathbf{Y} = (Y_1, \dots, Y_{|V|})$ is a sequence of *observable symbols* taking values from an alphabet $K$. $P(Y_v|X_v)$ is called *emission probability*.

We further introduce the following notation:

- $\mathtt{subtree} : V \rightarrow 2^V$ is a function mapping a node $v$ to a set of all nodes of the subtree rooted in $v$, i.e. $\mathtt{subtree}(v) = \{w \in V : \exists w = z_1, \dots, z_n = v, \forall i \in \{1 \dots n-1\} \quad \rho(z_i) = z_{i+1}\}$.
- $\mathbf{X}(v)$ is a sequence of hidden states of the subtree rooted in $v$, i.e. $\mathbf{X}(v) = \{X_w : w \in \mathtt{subtree}(v)\}$. Hence $\mathbf{X} = \mathbf{X}(r) = \{X_r, \mathbf{X}(w) : \rho(w) = r\}$.
- Analogously, $\mathbf{Y}(v)$ is a sequence of symbols of the subtree rooted in $v$.

Similarly to stationary first-order state-emitting HMM, we formulate three independence assumptions for HMTM:

[9] To avoid any terminological confusion, we should note that by HMM we mean only Hidden Markov *Chain* Models.

1. **stationary property** (analogy to time invariance property of HMM)
   $\forall v, w \in V \setminus \{r\} : P(X_v|X_{\rho(v)}) = P(X_w|X_{\rho(w)})$ &
   $\forall v, w \in V : P(Y_v|X_v) = P(Y_w|X_w)$,
   i.e. transition and emission probabilities are independent of nodes.
2. **tree-Markov property** (analogy to limited horizon property of HMM)
   $\forall v \in V \setminus \{r\}, \forall w \in V \setminus \mathit{subtree}(v) : P(\mathbf{X}(v)|X_{\rho(v)}, X_w) = P(\mathbf{X}(v)|X_{\rho(v)})$,
   i.e. given $X_{\rho(v)}$, all hidden states of the subtree rooted in $v$ are conditionally independent of any other nodes.[10]
3. **state-emission property**
   $\forall v, w \in V : P(Y_v|X_v, X_w, Y_w) = P(Y_v|X_v)$,
   i.e. given $X_v$, $Y_v$ is conditionally independent of any other nodes.

Let $v_1, \ldots, v_n$ be children of the root $r$, then using the tree-Markov property and mathematical induction we get:

$$\begin{aligned}
P(\mathbf{X}) &= P(X_r, \mathbf{X}(v_1), \ldots, \mathbf{X}(v_n)) \\
&= P(X_r)P(\mathbf{X}(v_1), \ldots, \mathbf{X}(v_n)|X_r) \\
&= P(X_r)P(\mathbf{X}(v_1)|X_r)P(\mathbf{X}(v_2), \ldots, \mathbf{X}(v_n)|X_r, \mathbf{X}(v_1)) \\
&= P(X_r)P(\mathbf{X}(v_1)|X_r)P(\mathbf{X}(v_2), \ldots, \mathbf{X}(v_n)|X_r) \\
&= P(X_r)P(\mathbf{X}(v_1)|X_r) \ldots P(\mathbf{X}(v_n)|X_r) \\
&= P(X_r) \prod_{v \in V \setminus \{r\}} P(X_v|X_{\rho(v)})
\end{aligned} \tag{1}$$

Using the state-emission property and mathematical induction we get:

$$\begin{aligned}
P(\mathbf{Y}|\mathbf{X}) &= P(Y_r|\mathbf{X})P(\mathbf{Y}(v_1), \ldots, \mathbf{Y}(v_n)|\mathbf{X}(v_1), \ldots, \mathbf{X}(v_n), X_r, Y_r) \\
&= P(Y_r|X_r)P(\mathbf{Y}(v_1), \ldots, \mathbf{Y}(v_n)|\mathbf{X}(v_1), \ldots, \mathbf{X}(v_n)) \\
&= \prod_{v \in V} P(Y_v|X_v)
\end{aligned} \tag{2}$$

From Equations 1 and 2 we can deduce the following factorization formula:

$$P(\mathbf{Y}, \mathbf{X}) = P(Y_r|X_r)P(X_r) \cdot \prod_{v \in V \setminus \{r\}} P(Y_v|X_v)P(X_v|X_{\rho(v)}) \tag{3}$$

[10] Our formulation of the tree-Markov property differs from the one used in Diligenti et al. (2003), which could be rewritten as
$\forall v, w, z \in V, \rho(w) = \rho(z) = v \implies P(\mathbf{X}(w)|X(v), \mathbf{X}(z)) = P(\mathbf{X}(w)|X(v))$,
i.e. given $X_{\rho(w)}$, the subtree of $w$ is conditionally independent of its sibling subtrees.
Such assumption is too weak to be used in the last two lines of Equation 1, where we need $P(X_v|X_{\rho(v)}, X_{\rho(\rho(v))}) = P(X_v|X_{\rho(v)})$.
On the other hand, the formulation used in Žabokrtský and Popel (2009) is unnecessarily strong:
$\forall v \in V \setminus \{r\}, \forall w \in V : P(X_v|X_{\rho(v)}, X_w) = P(X_v|X_{\rho(v)})$,
i.e. given $X_{\rho(v)}$, $X_v$ is conditionally independent of any other nodes.

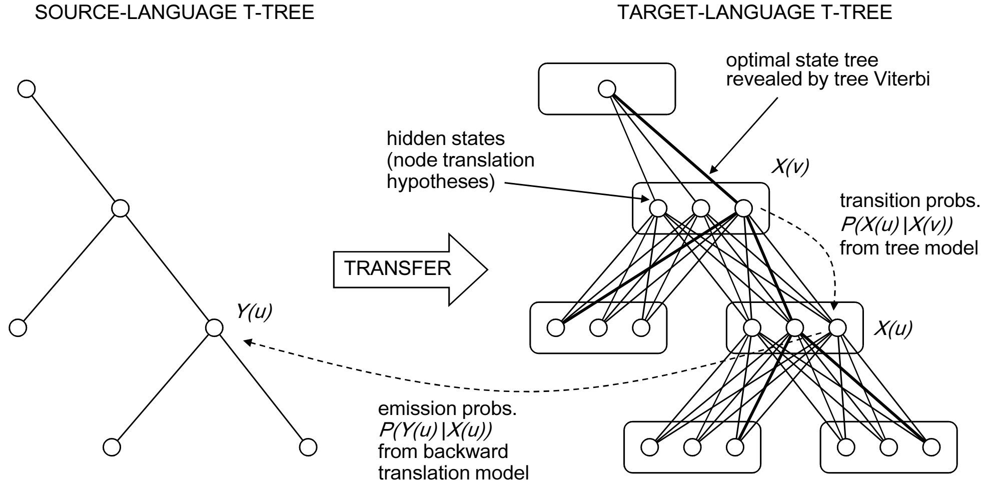
SOURCE-LANGUAGE T-TREE
TARGET-LANGUAGE T-TREE
optimal state tree revealed by tree Viterbi
hidden states (node translation hypotheses)
X(v)
transition probs. P(X(u)|X(v)) from tree model
TRANSFER
Y(u)
X(u)
emission probs. P(Y(u)|X(u)) from backward translation model

*Figure 1. Scheme of the tectogrammatical transfer as a task for HMTM.*

We see that HMTM (analogously to HMM, again) is defined by the following parameters:[11]

- $P(X_v|X_{\rho(v)})$ – transition probabilities between the hidden states of two tree-adjacent nodes,[12]
- $P(Y_v|X_v)$ – emission probabilities.

### 5.3. Application of HMTM in MT

**How to estimate emission and translation probabilities?**

When using HMTM in MT, labels of the source-language nodes can be interpreted as observable symbols and labels of the target-language nodes can be interpreted as hidden states (see Figure 1). In the case of TectoMT transfer, a label of a node is a pair of lemma and formeme. Therefore, the hidden states space (S) is the Cartesian product of lemmas and formemes possible for the target language and the alphabet of observable symbols (K) is the Cartesian product of lemmas and formemes possible for the source language.

HMTM emission probabilities can be estimated from the "backward" (source given target) node-to-node translation model. This node-to-node translation model can be

[11] As follows from the stationary property, the parameters are independent on the node $v$.

[12] The need for parametrizing also $P(X_r)$ (prior probabilities of hidden states in the root node) can be avoided by adding an artificial root whose state is fixed.

further estimated by factorization to the lemma translation dictionary and formeme translation dictionary.

HMTM transition probabilities can be estimated from the target-language tree model.

The decomposition into *translation model* and *language model* proved to be extremely useful in Statistical Machine Translation since Brown et al. (1993). It allows to compensate for the lack of parallel resources by the relative abundance of monolingual resources.

**Limitations of HMTM**

There are several limitations implied by the definition of HMTM, which we have to consider before applying it to MT.

The first limitation is merely a technical detail. The set of hidden states and the alphabet of observable symbols are supposed to be finite. This assumption can be easily fulfilled by introducing an artificial symbol/state for unknown tokens. However, in practice we are able to consider only a limited number of possible hidden states for each node, so the trick with an artificial symbol is not actually needed.

More serious limitations are induced by the three independence assumptions:

- **stationary property**
  We assume that the position of a node in a tree cannot influence its translation and emission probabilities. For example, this property would be violated if some words should be translated differently when being children of the main clause verb (i.e. grandchildren of the technical root).[13] According to our observations, such a dependence on the level of a node (i.e. distance from the root) is not a substantial issue.
  Another violation of the stationary property can be a dependency on word order. For example, some words should be translated differently when being at the beginning of the sentence.[13] These cases are also not a substantial problem.[14]
- **tree-Markov property**
  This assumption concerns only the target-language tree model. The conditional dependency (in the probabilistic sense) of a node on its parent corresponds well to the intuition behind dependency relations (in the linguistic sense) in dependency trees. However, there are special linguistic phenomena that violate this assumption. These phenomena are addressed in the manual for English tec-

[13] ...and this difference could be determined neither from the source node nor from the target-side parent node.

[14] PDT-style tectogrammatical nodes have an attribute `deepord`, which specifies the so-called *deep word order* for the purpose of communicative dynamism. TectoMT tectogrammatical trees use this attribute for surface word order. Nevertheless, if there were a reason, the attribute could be incorporated to the source node's label to circumvent the violation of the stationary property.

togrammatical annotation (Cinková et al., 2006) in Sections: Non-dependency edges, Dual dependency and Ambiguous dependency.

Predicative complements have the so-called dual dependency – on a verb and on a semantic noun, but only the former is represented by a tree edge.[15] In the following examples[16] we mark the predicative complement with an underline; its second dependency is always the subject (*He*). *He spoke of him as of his father. He left whistling. He lives alone.*

Although not considered a dual dependency, copula constructions also violate the assumption. For example, in sentences *He is a speaker.* and *It is a speaker.* we can disambiguate the sense of the object (*speaker*) based on the subject (*He* or *It*), but these nodes are siblings, so that the probabilistic dependency cannot be directly used in HMTM.

A possible solution to circumvent these violations and hopefully improve the translation quality is to incorporate the secondary dependencies into the labels of source nodes to be handled by the translation model.

- **state-emission property**

  This property can be weakened to "arc-emission property":
  given $X_v$ and $X_{\rho(v)}$, $Y_v$ is conditionally independent of any other nodes, i.e.
  $\forall v, w \in V : P(Y_v|X_v, X_{\rho(v)}, X_w, Y_w) = P(Y_v|X_v, X_{\rho(v)})$
  A factorization formula, analogical to Equation 3, can be then proved:

$$P(\mathbf{Y}, \mathbf{X}) = P(Y_r|X_r)P(X_r) \cdot \prod_{v \in V \setminus \{r\}} P(Y_v|X_v, X_{\rho(v)})P(X_v|X_{\rho(v)}) \qquad (4)$$

  With this generalization we can condition emission probabilities (i.e. translation model) on the parent node. Another (actually equivalent) method how to use a richer translation model, without the need of weakening the state-emission property, is to incorporate the needed attributes to the labels of target-side nodes.

The most limiting assumption from the MT viewpoint was not expressed explicitly yet:

- **isomorphism presumption**

  The source-language tree and the target-language tree are required to be isomorphic. In other words, only node labeling can be changed in the HMTM transfer step. This assumption concerning the tree isomorphism is problematic. As we have shown in Section 3, there are cases when it is not possible to translate a sentence correctly without violating the isomorphism presumption. On the other hand, only 8% of all translation errors in our annotation experiment were caused

[15] The latter dependency relation is indicated by the attribute `compl.rf`.

[16] We present English examples, but since the violations concern the target-language tree model, it would be more accurate to present Czech equivalents.

by such cases. Possible solutions to the problem are discussed in Popel (2009, p. 65).

### 5.4. Tree-modified Viterbi algorithm

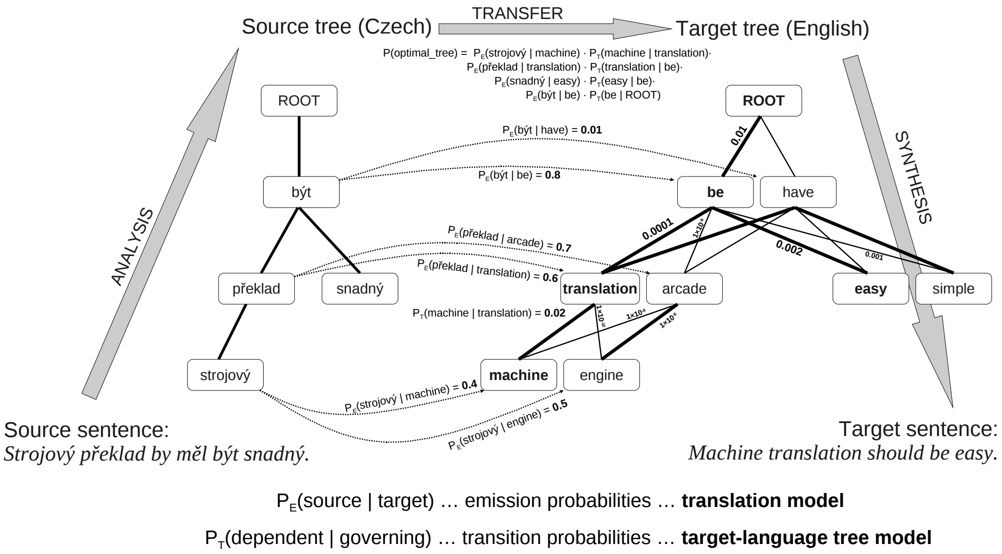
TRANSFER
Source tree (Czech)
Target tree (English)
P(optimal_tree) = $P_E$(strojový | machine) · $P_T$(machine | translation)·
$P_E$(překlad | translation) · $P_T$(translation | be)·
$P_E$(snadný | easy) · $P_T$(easy | be)·
$P_E$(být | be) · $P_T$(be | ROOT)
ANALYSIS
SYNTHESIS
ROOT
být
překlad
snadný
strojový
ROOT
be
have
translation
arcade
easy
simple
machine
engine
$P_E$(být | have) = **0.01**
$P_E$(být | be) = **0.8**
$P_E$(překlad | arcade) = **0.7**
$P_E$(překlad | translation) = **0.6**
$P_T$(machine | translation) = **0.02**
$P_E$(strojový | machine) = **0.4**
$P_E$(strojový | engine) = **0.5**
0.01
0.0001
$1\times10^{-8}$
0.002
0.001
$1\times10^{-10}$
$1\times10^{-8}$
$1\times10^{-8}$
Source sentence:
*Strojový překlad by měl být snadný.*
Target sentence:
*Machine translation should be easy.*
$P_E$(source | target) … emission probabilities … **translation model**
$P_T$(dependent | governing) … transition probabilities … **target-language tree model**

*Figure 2. A simplified example of the tectogrammatical transfer as a task for HMTM. The actual translation direction is English-to-Czech, but for better illustration of the target-side t-tree, we display the Czech-to-English direction in the figure.*

Naturally the question arises how to restore the most probable hidden tree labeling $\widehat{\mathbf{X}}$ given the observed tree labeling $\mathbf{Y}$ (and given the tree topology, of course). Using the factorization formula from Equation 3, we can write:

$$\begin{aligned}\widehat{\mathbf{X}} &= \arg\max_{\mathbf{X}} P(\mathbf{X}|\mathbf{Y}) \\ &= \arg\max_{\mathbf{X}} P(\mathbf{X},\mathbf{Y}) \\ &= \arg\max_{\mathbf{X}} P(Y_r|X_r)P(X_r) \cdot \prod_{v\in V\setminus\{r\}} P(Y_v|X_v)P(X_v|X_{\rho(v)})\end{aligned} \tag{5}$$

Similarly to the classical Viterbi algorithm, we can use dynamic programming to achieve an efficient implementation – $\mathcal{O}(|V| \cdot K^2)$ for $|V|$ nodes and $K$ states considered for every node.

However, we cannot start at the root node and perform top-down traversal, which would be the most straightforward analogy to the classical Viterbi algorithm. Instead, the tree-modified Viterbi algorithm starts at leaf nodes and continues upwards, storing in each node for each state and each its child the optimal downward pointer to the hidden state of the child. When the root is reached, the optimal state tree is retrieved by downward recursion along the pointers from the optimal root state. Downward pointers are marked by bold edges in Figure 2.

In practice, HMTM serves us as an inspiration, though for pragmatic reasons the implementation differs in some aspects from the theory. Apart from usual practices like computing probabilities in logarithmic space and smoothing transition probabilities, we use a factorization of the translation model into two channels: lemmas and formemes. Moreover, we use a forward translation model (target given source) in addition to the backward translation model (source given target), because it proved to have a positive effect on the translation quality. The emission probability is computed as a weighted average of the models.

## 6. Conclusions

We have implemented several improvements of English-Czech translation system TectoMT. In order to do so, we annotated 250 sentences produced by the baseline system and identified the most prominent errors and their sources. According to the error analysis, the assumption of isomorphism between the source and target tectogrammatical trees causes only 8% of errors. This facilitates the utilization of Hidden Tree Markov Model based transfer phase, which proved to be one of the most helpful modifications we have done.

We have achieved an improvement over the baseline 0.0659 BLEU (3.9735 NIST). Our new version of TectoMT reaches 0.0981 BLEU (4.7157 NIST). Although these results are still lower than those of the state-of-the-art English-Czech MT systems, our system is rapidly evolving and we see a great potential for further improvements.

## 7. Acknowledgement

The work on this project was supported by the grants MSM0021620838, MŠMT ČR LC536, and 1ET101120503.

## Bibliography

Bojar, Ondřej, Silvie Cinková, and Jan Ptáček. Towards English-to-Czech MT via Tectogrammatical Layer. *Prague Bulletin of Mathematical Linguistics*, 90, 2008a. ISSN 0032-6585.

Bojar, Ondřej, Miroslav Janíček, Zdeněk Žabokrtský, Pavel Češka, and Peter Beňa. CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008b. ELRA.

Bojar, Ondřej, David Mareček, Václav Novák, Martin Popel, Jan Ptáček, Jan Rouš, and Zdeněk Žabokrtský. English-Czech MT in 2008. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 125–129, Athens, Greece, March 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W09/W09-0x22`.

Brown, Peter E., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 1993. URL `http://acl.ldc.upenn.edu/J/J93/J93-2003.pdf`.

Cinková, Silvie, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Annotation of English on the tectogrammatical level. Technical Report 35, ÚFAL MFF UK, 2006.

Crouse, Matthew, Robert Nowak, and Richard Baraniuk. Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing*, 46(4):886–902, 1998.

Cuřín, Jan, Martin Čmejrek, Jiří Havelka, Jan Hajič, Vladislav Kuboň, and Zdeněk Žabokrtský. Prague Czech-English Dependency Treebank, Version 1.0. Linguistics Data Consortium, Catalog No.: LDC2004T25, 2004.

Diligenti, Michelangelo, Paolo Frasconi, and Marco Gori. Hidden tree Markov models for document image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25: 2003, 2003.

Durand, Jean-Baptiste, Paulo Goncalvès, and Yann Guédon. Computational methods for hidden Markov tree models - An application to wavelet trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560, 2004.

Hajič, Jan. *Disambiguation of Rich Inflection – Computational Morphology of Czech*. Charles University – The Karolinum Press, Prague, 2004.

Hopkins, Mark and Jonas Kuhn. Machine Translation as Tree Labeling. In *Proceedings of SSST, NAACL-HLT*, pages 41–48, 2007.

Koehn, Philipp and Christof Monz. Manual and automatic evaluation of machine translation between European languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121, 2006.

Manning, Christopher D. and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP)*, pages 523–530, Vancouver, BC, Canada, 2005.

Popel, Martin. Ways to Improve the Quality of English-Czech Machine Translation. Master's thesis, ÚFAL, MFF UK, Prague, Czech Republic, 2009.

Rouš, Jan. Probabilistic translation dictionary. Master's thesis, ÚFAL, MFF UK, Prague, Czech Republic, 2009.

Simard, Michel, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. Translating with non-contiguous phrases. In *Proceedings of HLT-EMNLP*, pages 755–762, October 2005.

Skounakis, Marios, Mark Craven, and Soumya Ray. Hierarchical Hidden Markov Models for Information Extraction. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 427–433. Morgan Kaufmann, 2003.

Spoustová, Drahomíra, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. The Best of Two Worlds: Cooperation of Statistical and Rule-Based Taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing, ACL 2007*, pages 67–74, Praha, 2007.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. Error Analysis of Machine Translation Output. In *Proceedings of the Fifth International Language Resources and Evaluation (LREC'06)*, pages 697–702, Genoa, Italy, May 2006.

Žabokrtský, Zdeněk 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.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL*, 2008.