

Czech-English Dependency-based Machine Translation

Data Preparation for the Starting up Experiments

Martin Čmejrek, Jan Cuřín and Jiří Havelka

Abstract

The MAGENTA system for generating English sentences from tectogrammatical representation was developed during the 2002 Language Engineering workshop organized by CLSP JHU. First, we summarize resources available before, as well as those created during the workshop. Then we describe automatic procedures for the preparation of training and testing data: analytical and tectogrammatical parsing of Czech, Czech-English tectogrammatical transfer based on lexical substitution using word-to-word translation dictionaries enhanced by the information from the Czech-English parallel corpus of WSJ, and conversion of Penn Treebank format into analytical and tectogrammatical representations. These methods are integrated into a complex environment for data preparation and training, and for evaluation of the system.¹

1 Introduction

From July 15 to August 23, 2002, the workshop on Language Engineering took place at the Center for Language and Speech Processing, Johns Hopkins University². MAGENTA is the system resulting from the work of the research group called “Generation in the context of MT” [8]. The system generates English analytical dependency trees from four different input options:

1. English tectogrammatical trees, automatically created from the Penn Treebank;
2. English tectogrammatical trees, human-annotated;
3. English tectogrammatical trees, automatically created from the Penn Treebank, improved by information from the Proposition Bank;
4. So called “Czenglish” tectogrammatical trees, automatically created from the Czech input text. This input option represents an attempt to develop a full MT system based on dependency trees.

The MAGENTA system comprises two independent approaches using dependency trees — statistical and rule-based.

In the sequel, we summarize resources available before (Sections 2–5) as well as those created during the workshop (Section 6). Sections 7–12 describe automatic procedures used for preparation of both training and testing data for all four input options used in the MAGENTA system. Section 13 describes the process of filtering dictionaries used in the transfer procedure.

¹This research was supported by the following grants: MŠMT ČR Grant No. LN00A063 and NSF Grant No. IIS-0121285.

²<http://www.clsp.jhu.edu/ws2002/>

2 The Prague Dependency Treebank

The Prague Dependency Treebank project³ aims at complex annotation of a corpus containing about 1.8M word occurrences (about 80,000 running text sentences) in Czech. The annotation, which is based on dependency syntax, is carried out in three steps: morphological, analytical, and tectogrammatical. The first two have been finished so far, presently, there are about 18,000 sentences tectogrammatically annotated. See [7] and [9] respectively for details of analytical and tectogrammatical annotation.

3 The Penn Treebank

The Penn Treebank project⁴ consists of about 1,500,000 tokens. Its bracketing style is based on constituent syntax, and comprises the surface syntactic structure, various types of null elements representing underlying positions for wh-movement, passive voice, infinitive constructions etc., and also predicate-argument structure markup. The largest component of the corpus consists of about 1 million words (about 40,000 sentences) from the Wall Street Journal newspaper. Only this part of the Penn Treebank corpus was used in the MAGENTA project.

4 The Proposition Bank

The PropBank project adds annotation of basic semantic propositions to the Penn Treebank corpus. For a verb, there is a list of syntactic frames (frameset), which have ever occurred in the annotated data; each position in the frame is associated with a semantic role in the predicate-argument structure of a given verb. The annotation started from the most frequent verbs (all occurrences of one verb are annotated in the same time) and continues to less frequent ones. See [10] for further details.

5 Existing Czech-English parallel corpora

Two considerable resources of Czech-English parallel texts were available before the workshop and were mentioned in previous experiments related to statistical Czech-English MT: Reader's Digest Výběr ([1], [4]), and IBM AIX and AS/400 operating system guides and messages translations [3]. The Reader's Digest Výběr corpus (58,137 sentence pairs) contains sentences from a broad domain, very free translations, while IBM corpus (119,886 sentence pairs) contains domain specific sentences, literal, almost word-by-word translations.

According to the automatic sentence alignment procedure, only 57% sentence pairs from the Reader's Digest Výběr corpus are 1-1 matching sentence pairs, compared to 98% of 1-1 sentence pairs from the IBM corpus.

Both corpora are automatically morphologically annotated by automatic BH tagging tools [6]. None of these corpora contain any syntactic annotation.

6 English to Czech translation of Penn Treebank

MAGENTA system uses syntactically annotated parallel texts as training data. Before the workshop preparation work started, there were no Czech-English parallel data manually syntactically annotated. We decided to translate a considerable part of the existing syntactically

³version 1.0; LDC catalog no.: LDC2001T10, ISBN: 1-58563-212-0, <http://ufal.mff.cuni.cz/pdt>

⁴version 3; LDC catalog no.: LDC99T42, ISBN: 1-58563-163-9

annotated English corpus (Penn Treebank) by human translators rather than to syntactically annotate existing Czech-English parallel texts. The translators were asked to translate each English sentence as a single Czech sentence and also to stick to the original sentence construction when possible. Before the beginning of the workshop, 11,189 WSJ sentences were translated into Czech by human translators (Table 1). The translation project continues, still after the workshop, aiming at translating the whole Penn Treebank.

For both training and evaluation measured by BLEU metrics, about 500 sentences were retranslated back from Czech into English by 4 different translators (Table 2).

data category	#sentence pairs
training	6,966
devtest ⁵	242
step devtest ⁶	2,737
evaltest ⁵	248
step evaltest ⁶	996

Table 1: English - Czech sentence pairs

data category	#sentences
devtest	259
evaltest	256

Table 2: WSJ sentences retranslated from Czech to English by 4 different translators

7 English Analytical Dependency Trees

Apart from various input options, the tree-to-tree transducer used by the MAGENTA system always generates analytical trees. This section describes the automatic preparation of the output part of the training data from Penn Treebank.

7.1 Marking Heads in English

The concept of the head of a phrase is important when transforming the phrase tree topology into the dependency one. We used Jason Eisner’s scripts for marking head constituents in each phrase.

7.2 Lemmatization of English

The Penn Treebank data contain manually assigned POS tags and this information substantially simplifies lemmatization. The lemmatization procedure just searches the list of all triples of word form, POS tag and lemma extracted from a large corpus, for a triple with a matching word form and POS and chooses the lemma from this triple. A large corpus of English (365M words, 13M sentences) was automatically POS tagged by MXPOST tagger [14] and lemmatized by the *morpha* tool [12]. The resulting list contains 910,216 triples.

Lemmatization procedure makes two attempts to find a lemma:

- first, it tries to find a triple with a matching word form and its (manually assigned) POS;

⁵covered by 4 human retranslations into English

⁶not covered by human retranslations

```

wsj_1700.mrg:5::
(S (NP~SBJ (DT @the the)
  (@NN @aim aim))
  (@VP (MD @would would)
    (@VP~ (@VB @be be)
      (S~-PRD (NP~-SBJ-1 (@-NONE- @* *))
        (@VP (TO @to to)
          (@VP~ (@VB @end end)
            (NP~ (@NP (DT @the the)
              (NN @guerrilla guerrilla)
              (@NN @war war))
            (PP (@IN @for for)
              (NP~ (@NP (@NN @control control))
                (PP (@IN @of of)
                  (NP~ (@NPR (@NNP @Cambodia Cambodia)))))))
          (PP-MNR (@IN @by by)
            (S~-NOM (NP~-SBJ (@-NONE- @*-1 *-1))
              (@VP (@VBG @allowing allow)
                (NP~ (DT @the the)
                  (@NPR (NPN @Khmer Khmer)
                    (@NNP @Rouge Rouge)))
                (NP~ (@NP (DT @a a)
                  (JJ @small small)
                  (@NN @share share))
                (PP (@IN @of of)
                  (NP~ (@NN @power power))))))))))
  (. @. .))

```

Figure 1: Example of a lemmatized sentence with marked heads: “*The aim would be to end the guerrilla war for control of Cambodia by allowing the Khmer Rouge a small share of power.*”. Terminal nodes consist of a sequence of part-of-speech, word form, lemma, and a unique id. The names of the head constituent names start with @. (In the noun phrase *Khmer Rouge* the word *Rouge* was marked as the head by mistake.)

- if it fails, it makes a second attempt with the word form converted to lowercase.

If it fails in both attempts, then it chooses the given word form as the lemma. For technical reasons, a unique identifier is assigned to each token in this step. Figure 1 contains an example of a lemmatized sentence with marked heads.

7.3 Transformation of Phrase Trees into Analytical Representations

The transformation of the lemmatized Penn Treebank phrase trees with marked heads to analytical trees consists of three steps:

1. Structural transformation

The transformation from the phrase tree to the dependency tree is defined recursively:

- Terminal nodes of the phrase are converted to nodes of the dependency tree.
- Constituents of a non-terminal node are converted into separate dependency trees. The root node of the dependency tree transformed from the head constituent becomes the main root. Dependency trees transformed from the left and right siblings of the head constituent are attached to the main root as the left or right children, respectively.

- Nodes representing traces are removed and their children are reattached to the parent of the trace.
- Handling of coordination in PDT is different from the Penn Treebank annotation style and Jason Eisner’s head assigning scripts; in the case of a phrase containing a coordinating conjunction (**CC**), we consider the rightmost **CC** as the head. The treatment of apposition is a more difficult task, since there is no explicit annotation of this phenomenon in the Penn Treebank; constituents of a noun phrase separated by commas (and not containing **CC**) are considered to be in apposition and the rightmost comma is the head.

2. Assignment of analytical functions

The information from the phrase tree and the structure of the dependency tree are both used for analytical function assignment.

- WSJ function tag to analytical function mapping: some function tags of a phrase tree correspond to analytic functions in an analytical tree and can be mapped to them: **SBJ** → **Sb**, **DTV** → **Obj**, **LGS** → **Obj**, **BNF** → **Obj**, **TPC** → **Obj**, **CLR** → **Obj**, **ADV** → **Adv**, **DIR** → **Adv**, **EXT** → **Adv**, **LOC** → **Adv**, **MNR** → **Adv**, **PRP** → **Adv**, **TMP** → **Adv**, **PUT** → **Adv**.
- Assignment of analytical functions using local context: for assigning analytical functions to the remaining nodes, we use simple rules taking into account POS, the name of the constituent headed by a node in the original phrase tree. In the rules this information for the current node, its parent and grandparent can be used. For example, the rule

$$\text{mPOS} = \text{DT} | \text{mAF} = \text{Atr}$$

assigns the analytical function **Atr** to every determiner, the rule

$$\text{mPOS} = \text{MD} | \text{pPOS} = \text{VB} | \text{mAF} = \text{AuxV}$$

assigns the function tag **AuxV** to a modal verb headed by a verb, etc. The attribute **mPOS** representing the POS of the node is obligatory for every rule. The rules are examined primarily in the order of the longest prefix of the POS of the given node and secondarily in the order as they are listed in the rule file. The ordering of rules is important since the first matching rule found assigns the analytical function and the search is finished.

3. PDT specific operations

Differences between PDT and Penn Treebank annotation schemes, mainly the markup of coordinations, appositions, and prepositional phrases are handled by this step.

- Coordinations and appositions: the analytical function, which was originally assigned to the head of coordination or apposition respectively is propagated to children nodes with the attached suffix **_Co** or **_Ap** and the head nodes get the analytical function **Coord** or **Apos**.
- Prepositional phrases: the analytical function originally assigned to the preposition node is propagated to its child and the preposition node is labeled **AuxP**.
- Sentences in the PDT annotation style always contain a root node labeled **AuxS**, which, as the only one in the dependency tree, does not correspond to any terminal of the phrase tree; the root node is inserted above the original root. While in the Penn Treebank, the final punctuation is a constituent of the sentence phrase, in the analytical tree, it is moved under the sentence root node.

After these rearrangements modifying the local context of some nodes, the analytical function assignment procedure attempts to label the remaining empty positions.

data category	#sentences
training	42,697
devtest	248
step devtest	3,384
evaltest	249
step evaltest	1,416

Table 3: Penn Treebank sentences automatically converted into Analytical and Tectogrammatical representation

8 English Tectogrammatical Dependency Trees

The transformation of Penn Treebank phrase trees into Tectogrammatical representation reuses the preprocessing (marking heads and lemmatization) described in Sections 7.1 and 7.2, and consists of the following three steps:

1. Structural Transformation - the topology of the tectogrammatical tree is derived from the topology of the PTB tree, and each node is labeled with the information from the PTB tree. In this step, the concept of head of a PTB subtree plays a key role;
2. Functor Assignment - a functor is assigned to each node of the tectogrammatical tree;
3. Grammateme Assignment - morphological (e.g. Tense, Degree of Comparison) and syntactic grammatemes (e.g. TWHEN_AFT(er)) are assigned to each node of the tectogrammatical tree. The assignment of the morphological attributes is based on PennTreebank tags and reflects basic morphological properties of the language. The syntactic grammatemes capture more specific information about deep syntactic structure. At the moment, there are no automatic tools for the assignment of the latter ones.

The whole procedure is described in detail in [15].

In order to gain a “gold standard” annotation, roughly 1,000 sentences have been annotated manually (see Table 4). These data are assigned morphological grammatemes (the full set of values) and syntactic grammatemes, and the nodes are reordered according to topic-focus-articulation.

data category	#sentences
training	561
devtest	248
step devtest	199
evaltest	249
step evaltest	0

Table 4: Penn Treebank sentences manually assigned Tectogrammatical representations

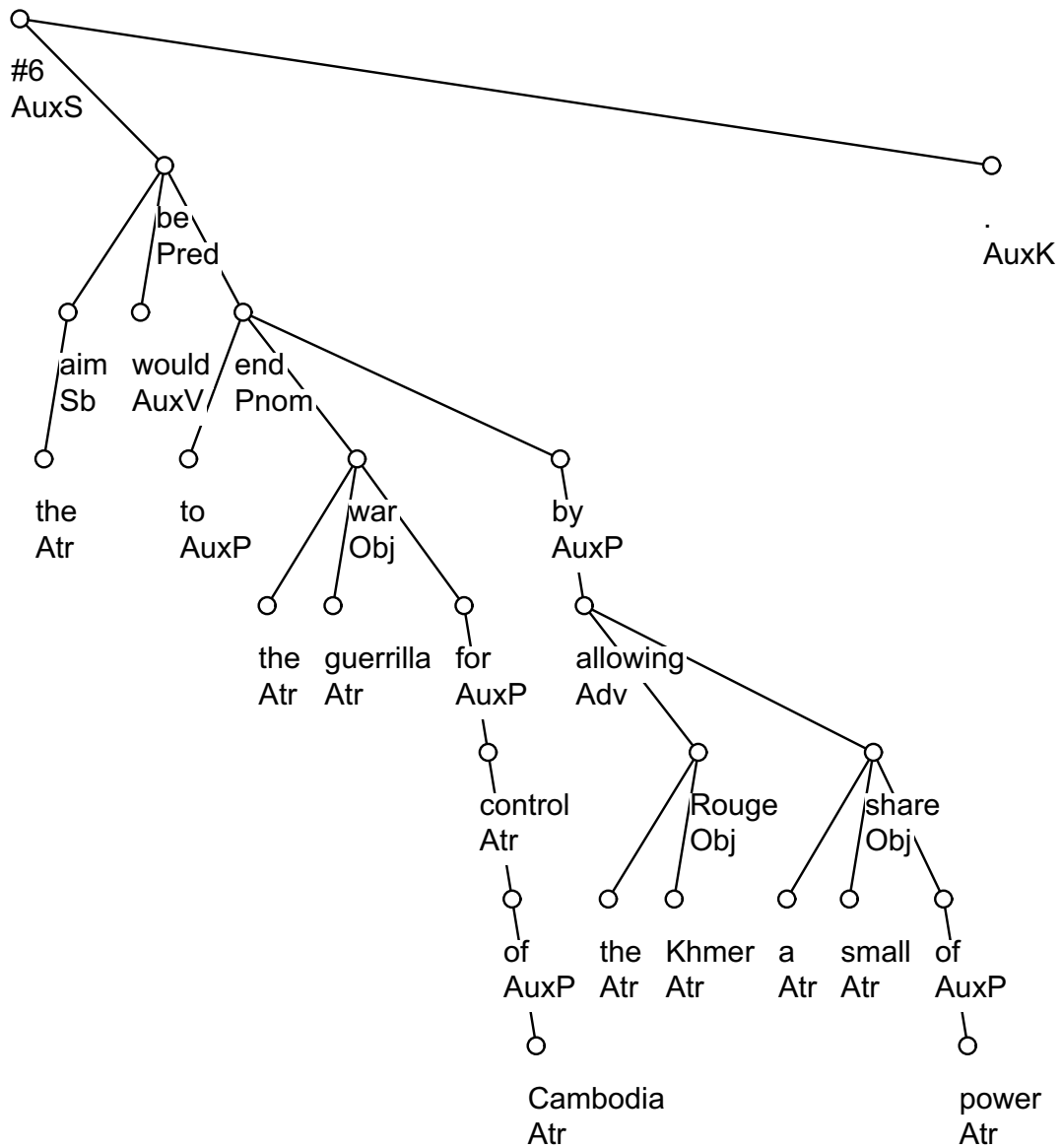


Figure 2: Example of an analytical tree automatically converted from Penn Treebank: “*The aim would be to end the guerrilla war for control of Cambodia by allowing the Khmer Rouge a small share of power.*” (In the noun phrase *Khmer Rouge* the word *Rouge* was marked as head by mistake.)

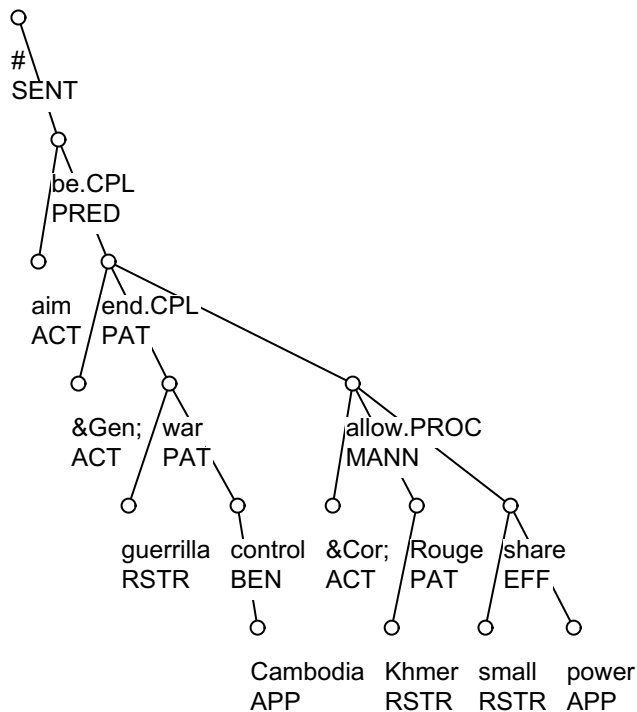


Figure 3: Example of a tectogrammatical tree automatically converted from Penn Treebank: “The aim would be to end the guerrilla war for control of Cambodia by allowing the Khmer Rouge a small share of power.” Lexical symbol &Cor; stands for the tectogrammatical counterpart of the subject, and &Gen; for the general participant.

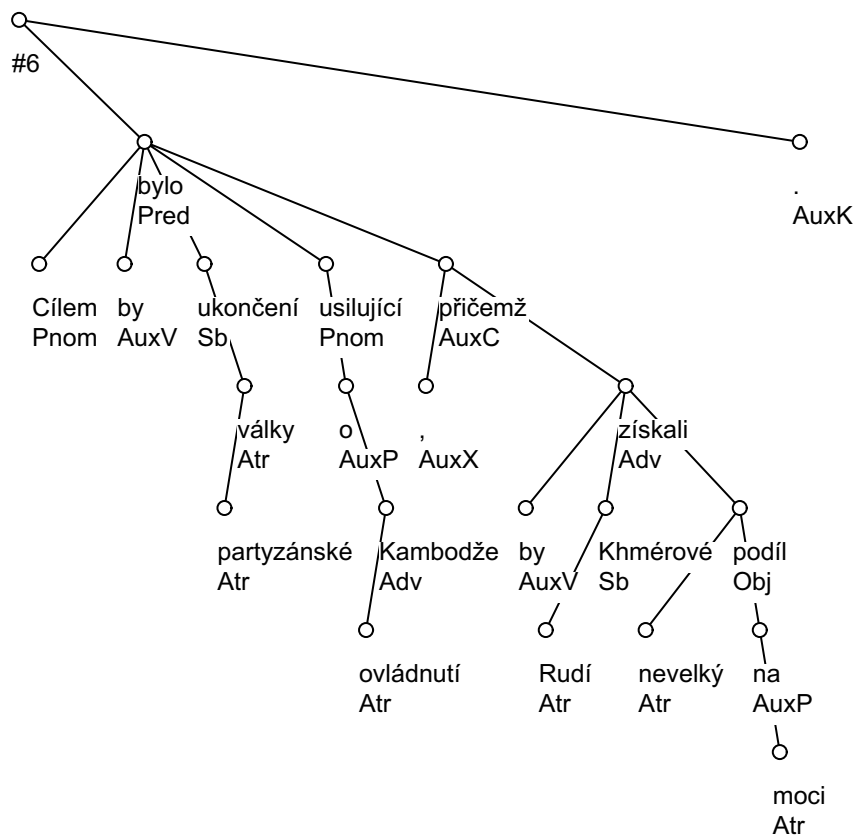


Figure 4: Example of a Czech analytical tree automatically parsed from input text: “*Cílem by bylo ukončení partyzánské války usilující o ovládnutí Kambodže, přičemž by Rudí Khmerské získali nevelký podíl na moci.*” (As a result of the automatic procedure, this tree contains some errors in attachment and analytical function assignment.)

9 Part-of-Speech Tagging and Lemmatization of Czech

The Czech translations of Penn Treebank were automatically tokenized and morphologically tagged, each word form was assigned a basic form - *lemma* by [6] tagging tools.

10 Analytical parsing of Czech

Czech analytical parsing consists of a statistical dependency parser for Czech [5] and a module for automatic analytical functor assignment [16]. For efficiency reasons, sentences longer than 60 words were excluded from the corpus in this step.

Figure 4 contains an example of a Czech analytical tree.

11 Tectogrammatical parsing of Czech

During the tectogrammatical parsing of Czech, the analytical tree structure is converted into the tectogrammatical one. These transformations are described by linguistic rules [2]. Then, tectogrammatical functors are assigned by C4.5 classifier [16].

Figure 5 contains an example of a Czech tectogrammatical tree.

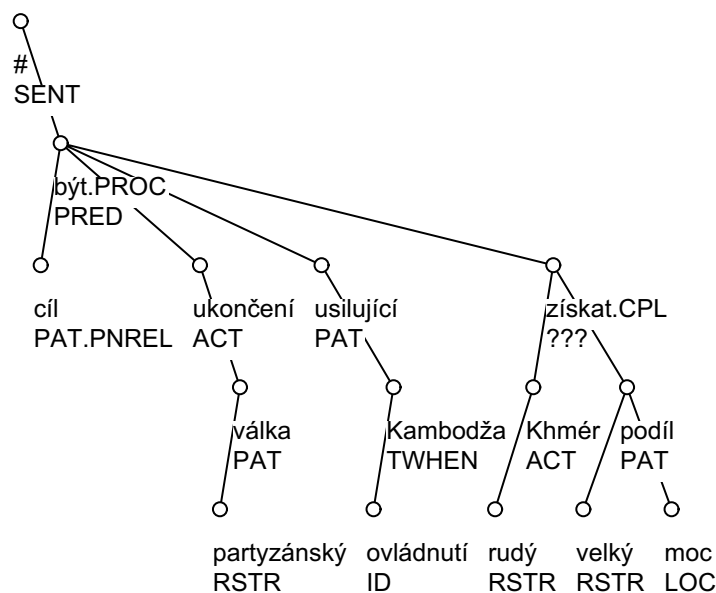


Figure 5: Example of a Czech tectogrammatical tree automatically converted from the analytical one: “*Cílem by bylo ukončení partyzánské války usilující o ovládnutí Kambodže, přičemž by Rudí Khmérové získali nevelký podíl na moci.*” (The incorrect structure from the analytical tree in Figure 4 persists.)

12 Tectogrammatical Lexical Transfer - “Czenglish” tectogrammatical representation

In this step, tectogrammatical trees automatically created from Czech input text are transferred into so called “Czenglish” tectogrammatical trees. The transfer procedure itself is a lexical replacement of the trlemma attribute of autosemantic nodes by its English equivalent found in the Czech-English probabilistic dictionary. Because of multiple translation possibilities, the output structure is a forest of “Czenglish” tectogrammatical trees represented in a packed-tree format [11].

12.1 Translation equivalent replacement algorithm (for 1-1, 1-2 entry-translation mapping).

For each Czech tectogrammatical tree (TGTree) do:

1. Start at the root
2. In the dictionary, find translation equivalents for ”trlemma” of this node
3. If there is only one translation, add the appropriate TN-tags to this node, continue with step 9
If there is more than one translation:
4. Change the current node into OR_node
5. For each child of the current node create a new ID_node, set the parent of the child to this ID_node

6. Create new WORD_node for each translation, set parents of the new nodes to the OR_node
7. If there is a two-word translation, create a new node for the dependent word and set its parent to the appropriate WORD_node created in 6)
8. For each ID_node created in step 5 set multiple parents to all WORD_nodes created in step 6
9. Backtrack to the next node in TGTree and continue with step 2

Figure 6 contains an example of the “Czenglish” tectogrammatical packed-tree.

For practical reasons such as time efficiency and integration with the Tree-to-tree transducer, a simplified version, taking into account only the first most probable translation was used during the time of the workshop. Also 1-2 translations were handled as 1-1 — two words in one trlemma attribute.

13 Czech-English Word-to-Word Translation Dictionaries

13.1 Manual Dictionary Sources

There were three different sources of Czech-English manual dictionaries available, two of them were downloaded from the Web (WinGED, GNU/FDL), and one was extracted from the Czech/English EuroWordNet. See dictionaries parameters in Table 5.

13.2 Dictionary Filtering

For a subsequent use of these dictionaries for a simple transfer from the Czech to the English tectogrammatical trees (see Section 12) a relatively huge number of possible translations for each entry⁵ had to be filtered. The aim of the filtering is to exclude synonyms from the translation list, i.e. to choose one representative per meaning.

First, all dictionaries are converted into a unified XML format (See description of steps *a8822*, *b8822* in Table 6) and merged together preserving information about the source dictionary (*c8822*, *d8822*).

This merged dictionary consisting of entry/translation pairs (Czech entries and English translations in our case) is enriched by the following procedures:

- Frequencies of English word obtained from large English monolingual corpora are added to each translation (*e8822*). See description of the corpora in Section 7.2.
- Czech POS tag and stem are added to each entry using the Czech morphological analyzer (*f8822*, [6]).
- English POS tag is added to each translation (*g8822*). If there is more than one English POS tag obtained from the English morphological analyzer [14], the English POS tag is “disambiguated” according to the Czech POS in the appropriate entry/translation pair.

We select few relevant translations for each entry taking into account the sum of the weights of the source dictionaries (see dictionary weights in Table 5), the frequencies from English monolingual corpora, and the correspondence of the Czech and English POS tags (*j8822*).

⁵For example for WinGED dictionary it is 2.44 translations per entry in average, and excluding 1-1 entry/translation pairs even 4.51 translations/entry.

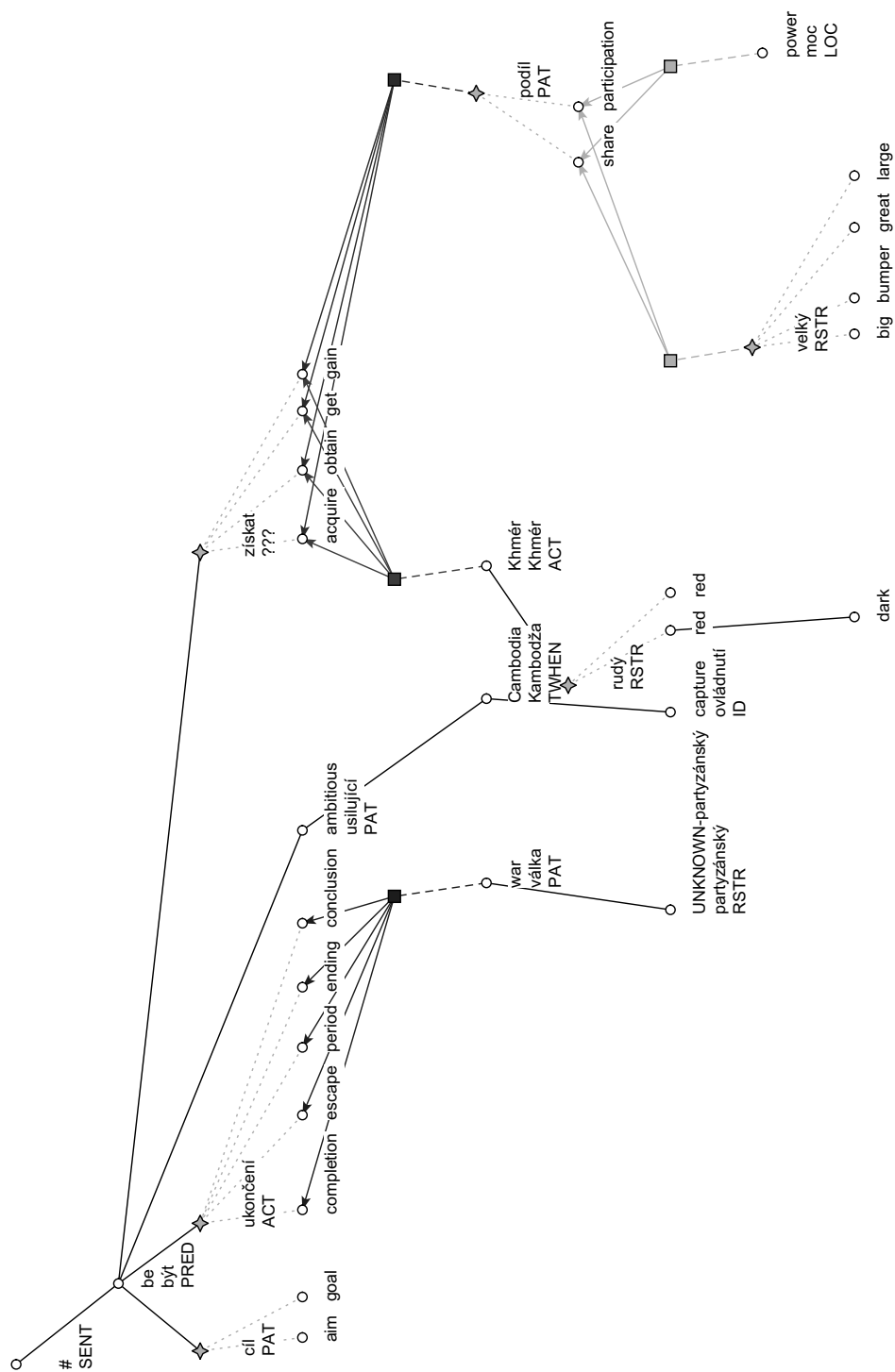


Figure 6: Example of a packed tree representation of a forest of Czeenglish tectogrammatical trees resulting from the sentence: “Cílem by bylo ukončení partyzánské války usilující o ovládnutí Kambodže, přičemž by Rudí Khmérové získali nevelký podíl na moci.”

<i>Dictionary</i>	<i># entries</i>	<i># translations</i>	<i>Weight</i>
EuroWordNet	12,052	48,525	3
GNU/FDL	12,428	17,462	3
WinGED	16,296	39,769	2
<i>merged</i>	33,028	87,955	—

Table 5: Dictionary parameters and weights

13.3 Scoring Translations Using GIZA++

To make dictionaries more sensitive to a specific domain, which is in our case the domain of financial news, and because of the use of stochastic methods in the subsequent stages (such as transduction of English tectogrammatical trees to English analytical trees), it would help to have the translations somehow weighted.

By extending this dictionary by the training part of the Czech-English parallel corpus (7,412 sentences from WSJ) and by running GIZA++ training (translation models 1-4, see [13]) on it (*steps a8824-e8824*), we obtained a probabilistic Czech-English dictionary. As a result, the entry/translation pairs seen in the parallel corpus become more probable. For entry/translation pairs not seen in the parallel text, the probability distribution among translations is uniform. The translation is “GIZA++ selected” if its probability is higher than a threshold, which is set to 0.10 in our case.

The final selection (*step l8822*) contains translations selected by both dictionary and GIZA++ selectors. In addition, translations not covered by the original dictionary can be included into the final selection, if they were newly discovered in the parallel corpus by GIZA++ training and their probability is significant (higher than the most probable translation so far).

The translations of the final selection are used in the transfer (*steps h8801 or i8801*). See the sample of the dictionary in Figure 7.

14 Conclusion and further development

The system described, in conjunction with the MAGENTA system, comprises the whole way from the Czech plain text sentence to the English one. It integrates the latest results in analytical and tectogrammatical parsing of Czech, experiments with existing word-to-word dictionaries combined with those automatically obtained from a parallel corpus, lexical transfer, and conversions between Penn Treebank and Prague Dependency Treebank annotation style.

Since the MAGENTA system was designed as a language-independent tool, we plan to reverse the direction of translation — from English to Czech. We intend to implement parsing of English sentences into analytical and tectogrammatical representations and English-Czech lexical transfer.

References

- [1] Yaser Al-Onaizan, Jan Cuřín, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. The statistical machine translation. Technical report, 1999. NLP WS’99 Final Report.
- [2] Alena Böhmová. Automatic procedures in tectogrammatical tagging. *The Prague Bulletin of Mathematical Linguistics*, 76, 2001.
- [3] Jan Cuřín and Martin Čmejrek. Automatic translation lexicon extraction from czech-english parallel texts. *The Prague Bulletin of Mathematical Linguistics*, 71:47–57, 1999.
- [4] Jan Cuřín and Martin Čmejrek. Automatic extraction of terminological translation lexicon from czech-english parallel texts. *Internationa-*

- tional Journal of Corpus Linguistics*, 6(Special Issue):1–12, December 2001.
- [5] Jan Hajič, Eric Brill, Michael Collins, Barbora Hladká, Douglas Jones, Cynthia Kuo, Lance Ramshaw, Oren Schwartz, Christopher Tillmann, and Daniel Zeman. Core Natural Language Processing Technology Applicable to Multiple Languages. Technical Report Research Note 37, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, 1998.
- [6] Jan Hajič and Barbora Hladká. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of COLING-ACL Conference*, pages 483–490, Montreal, Canada, 1998.
- [7] Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Uřešová, Alla Bémová, Jan Štěpánek, Petr Pajas, and Jiří Kárník. *A Manual for Analytic Layer Tagging of the Prague Dependency Treebank*. Prague, Czech Republic, 2001. English translation of the original Czech version, http://shadow.ms.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/aman_en.pdf.
- [8] Jan Hajič, Martin Čmejrek, Bonnie Dorr, Yuan Ding, Jason Eisner, Daniel Gildea, Terry Koo, Kristen Parton, Gerald Penn, Dragomir Radev, and Owen Rambow. Natural language generation in the context of machine translation. Technical report, 2002. NLP WS’02 Final Report — in print.
- [9] Eva Hajičová, Jarmila Panevová, and Petr Sgall. A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank. Technical Report TR-2000-09, ÚFAL MFF UK, Prague, Czech Republic, 2000.
- [10] Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding semantic annotation to the penn treebank. In *In Proceedings of the Human Language Technology Conference*, San Diego, California, 2002.
- [11] I. Langkilde. Forest-based statistical sentence generation. In *Proceedings of NAACL’00*, Seattle, WA, 2000.
- [12] G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223, 2001.
- [13] F. J. Och and H. Ney. Improved statistical alignment models. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hongkong, China, October 2000.
- [14] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, University of Pennsylvania, May 1996. ACL.
- [15] Zdeněk Žabokrtský and Ivona Kučerová. Transforming penn treebank phrase trees into (praguan) tectogrammatical dependency trees. *The Prague Bulletin of Mathematical Linguistics*, 78, 2002.
- [16] Zdeněk Žabokrtský, Petr Sgall, and Džeroski Sašo. Machine learning approach to automatic functor assignment in the prague dependency treebank. In *Proceedings of LREC 2002 (Third International Conference on Language Resources and Evaluation)*, volume V, pages 1513–1520, Las Palmas de Gran Canaria, Spain, 2002.

```

<e>zesílit<t>V
    [FSG] <tr>increase<trt>V<prob>0.327524
    [FSG] <tr>reinforce<trt>V<prob>0.280199
    [FSG] <tr>amplify<trt>V<prob>0.280198
    [G] <tr>re-enforce<trt>V<prob>0.0560397
    [G] <tr>reenforce<trt>V<prob>0.0560397

<e>výběr<t>N
    [FSG] <tr>choice<trt>N<prob>0.404815
    [FSG] <tr>selection<trt>N<prob>0.328721
    [G] <tr>option<trt>N<prob>0.0579416
    [G] <tr>digest<trt>N<prob>0.0547869
    [G] <tr>compilation<trt>N<prob>0.0547869
    [] <tr>alternative<trt>N<prob>0.0519888
    [] <tr>sample<trt>N<prob>0.0469601

<e>selekce<t>N
    [FSG] <tr>selection<trt>N<prob>0.542169
    [FSG] <tr>choice<trt>N<prob>0.457831

<e>rozšířit<t>V
    [FSG] <tr>widen<trt>V<prob>0.20402
    [FSG] <tr>enlarge<trt>V<prob>0.20402
    [G] <tr>expand<trt>V<prob>0.138949
    [G] <tr>extend<trt>V<prob>0.130029
    [G] <tr>spread<trt>V<prob>0.0822508
    [] <tr>step<trt>V<prob>0.0516784
    [] <tr>let<trt>X<prob>0.0459122
    [] <tr>stretch<trt>V<prob>0.0427784
    [] <tr>larger<trt>V<prob>0.040804
    [] <tr>broaden<trt>V<prob>0.040804
    [] <tr>ground-handling<trt>N<prob>0.0136013
    [] <tr>make larger<trt>V<prob>0.01
    [] <tr>let_out<trt>V<prob>0.01
    [] <tr>reconsider<trt>V<prob>0.00515253

    [S] ... dictionary weight selection
    [G] ... GIZA++ selection
    [F] ... final selection

```

Figure 7: A sample of the Czech-English dictionary used for the transfer.

step	functionality summary
8801 – Czech data	
a8801	tokenization of Czech WSJ files
b8801	morphology & tagging
c8801	preprocessing necessary for Collins’ parser
d8801	statistical dependency parser for Czech
e8801	analytical function assignment
f8801	rule based conversion of analytical representation into tectogrammat- ical representation
g8801	C 4.5 based assignment of tectogrammatical functors
h8801	lexical transfer into “Czenglish” packed forest representation
i8801	simplified lexical transfer into “Czenglish”, first translation
8802 – English data	
a8802	marking heads in Penn Treebank trees
b8802	lemmatization of Penn Treebank
c8802	conversion of Penn Treebank trees into analytical trees
d8802	conversion of Penn Treebank trees into tectogrammatical trees
8822 – Czech-English Dictionary Filtering	
a8822	creating unified SGML format of dictionaries from various input for- mats
b8822	filtering out garbage
c8822	conversion into XML
d8822	preparing dictionary for POS annotation
e8822	adding frequencies from large monolingual corpus to English transla- tions
f8822	morphological analysis of Czech entries
g8822	morphological analysis of English translations
h8822	merging temporary dictionaries from steps e8822, f8822 and g8822 into one XML dictionary
i8822	converting the whole dictionary (without any filtering criteria) to a parallel plain text corpus to be used as GIZA++ training data
j8822	selecting translations according to dictionary weights and converting the selected sub-dictionary to a parallel plain text corpus to be used as GIZA++ training data
k8822	merges results of GIZA++ dictionary training (e8824) with XML dictionary.
l8822	selecting translations for transfer according to dictionary weights and GIZA++ translation probabilities
m8822	stores Czech-English translation dictionary to be used for transfer (h8801, i8801)
8824 – Czech-English Probabilistic Dictionary Training	
a8824	creating parallel corpus from Czech tectogrammatical trees (g8801) and English tectogrammatical trees (d8802)
b8824	creating plain text parallel corpus of trlemmas for GIZA++ training
c8824	extending training corpus by corpus obtained from i8822 or j8822
d8824	GIZA++ training, model 4
e8824	converting GIZA++ output into XML

Table 6: Summary of used scripts