Two Tectogrammatical Realizers Side by Side: Case of English and Czech

Jan Ptáček[1]

Abstract. We present a work in progress on a pair of morpho-syntactic realizers sharing the same architecture. We provide description of input tree structures, describe our procedural approach on two typologically different languages and finally present preliminary evaluation results conducted on manually annotated treebank.

1 INTRODUCTION

Natural language generation (NLG) is the process where required information available in digital storage is retrieved, formulated in statements of natural language and conveyed to interested reader or hearer. NLG usually consist of three consecutive steps: (1) content determination, (2) content planning and (3) sentence realization. First two steps regarding communicated content are highly specific across various applications of NLG and thus only the sentence realization has so far emerged as a suitable task worth of general, wide-coverage and reusable solution.

Such of-the-shelf realizer can considerably speed up development of a summarization, reporting, question answering, machine translations (MT) or human-computer dialog applications. While developing our realizer(s) we target especially the last two examples of given use cases.

Two (to some extent) competing approaches have been so far presented in the NLG community. Firstly, so called symbolic realizers share solid background in a well-established linguistic theory and implement a grammar (crafted by hand) operating on structures defined by respective theoretical framework. One, fully specified structure, containing all information needed to assemble resulting sentence and conforming to the grammar at the same time, is sought-after by the means of unification process.

Under the influence of successful employment of statistical methods in MT was the symbolic paradigm extended and gave rise to statistical realizers. Firstly, hybrid symbolic-statistical systems operating in two stages appeared, where hand written grammar rules generated list of candidate sentences and statistical reranking picked the best realization. Later, even the search for candidates follows a grammar automatically extracted from suitable corpora and is guided by a mixture of language models.

While symbolic realizes draw from systemic linguistic school (Fuf/Surge, [1]), from the Meaning-Text Theory (RealPro, [2]) or from Combinatory Categorial Grammars as in case of pure statistical realizer OpenCCG [3] realizer, we have decided

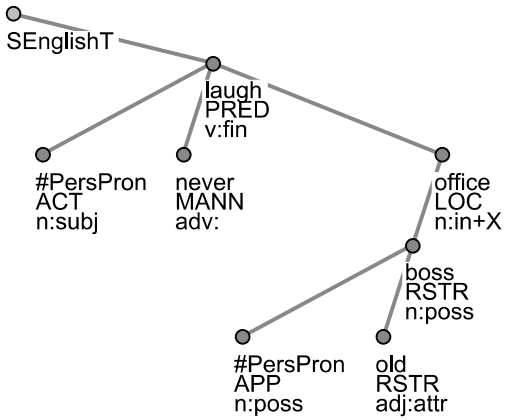

Figure 1. English tectogrammatical representation of the sentence 'She has never laughed in her old boss` office.' (Each node displays its lexical content, and a *functor*: dependency label giving its function. Third line displays a *formeme*: a morphosyntactic form to be used in the surface shape of sentence, that is assigned to each node in the Formeme Selection phase of realization)

to build our Czech realizer in the framework of Functional Generative Description (FGD, [4]) founded in middle 1960's by P.Sgall. Our decision is backed by a considerable number of resources developed for Czech language under the FGD framework, mainly the Prague Dependecy Treebank (PDT 2.0, [5]), tools handling Czech morphology [6], and valency lexicon PDT-Vallex [7].

A treebank in a given language is obviously needed while building a statistical realizer, but proves crucial even for a pure symbolic realizer, because it allows for reliable evaluation on a number of sentences covering various syntactic constructions.

When the work on preparation of PEDT 2.0 [8] has begun, it was a logical step to reuse the architecture of existing Czech sentence realizer and start the building of an English one.

2 INPUT STRUCTURES

The Functional Generative Description is a stratificational dependency framework describing language phenomena on number of layers. We are dealing with the most abstract layer of the description, the tectogrammatical layer. A comparison of tectogrammatical tree structures (t-trees) to input structures of a prominent realizer package – Fuf/Surge follows.

M. Elhadad characterizes the realizer input in [1] as "skeletal, partially lexicalized thematic tree specifying the semantic roles, open-class lexical items and top-level syntactic category of each constituent."

[1] Charles University in Prague, ÚFAL, Malostranské nám. 25, 11800, Czech Republic. Email: ptacek@ufal.mff.cuni.cz.

T-trees also fit the same definition, but where Fuf/Surge fills the description of skeletal tree with an Attribute Value Matrix (as known from HPSG), FGD makes use of a rooted dependency tree.

Where thematic roles in Fuf/Surge are those specified for verbs by Levin [9], FGD uses the notion of a *functor*; a value from unified set of 68 labels, describing the function of individual argument in predicate-argument structure. Example values of the functor include: Actor/Bearer, Patient, Addressee, Origin, Effect, Appurtenance, Cause, Comparison, Manner, Temporal Adjuncts and other circumstantial relations. FGD characterizes the functor as syntactico-semantic relation; the rationale is to capture major semantic differences but still have a closed set of distinctive values. Fuf/Surge – using more specific thematic roles – then faces a difficult problem of handling verbs that do not fit in any element of Levin's hierarchy and the realizer is able to process them only through a shallower and less semantic mode of operation.

T-trees do not include representation of closed-class words, again similarly as Fuf/Surge input structures.

The role of top-level syntactic categories used by Fuf/Surge (i.e. clause, np, pers_pron, etc.) is in FGD fulfilled by semantic part of speech attribute (c.f. details in [5]).

To summarize: the information captured within the tectogrammatical description of a sentence is comparable to an input specifications of widely deployed Fuf/Surge realizer; with the exception of thematic roles, where the corresponding functor attribute offers not so fine-grained, but more versatile set of values.

2.1 Prague Dependency Treebank

In this Section, we give a brief quantitative characteristic of both treebanks that serve as test beds for our realizers.

The PDT 2.0 data consists of 7,110 manually annotated textual documents, containing altogether 115,844 sentences with 1,957,247 tokens (word forms and punctuation marks). 45% of the data is annotated on the tectogrammatical layer (i.e. 3,165 documents, 49,431 sentences, 833,195 tokens).

The texts in electronic form have been provided by the Institute of the Czech National Corpus and consists of annotated non-abbreviated articles from two Czech major daily newspapers, one business journal and one scientific journal.[3]

2.2 Prague English Dependency Treebank

The PEDT 2.0 treebank comprises one half of a parallel Prague Czech-English Dependency Treebank focused mainly as a linguistic resource for the purposes of machine translation.

Once released, the PEDT 2.0 will contain 49,208 sentences containing over 1.2 million tokens of Wall Street Journal part of the Penn Treebank [10], annotated in a scheme of PDT 2.0, adapted for English. The PEDT 2.0 treebank is still work in progress. Of the English part, 12.029 sentences (305.666 words, i.e. 24%) has final annotation of dependency and functor relations. These data are automatically brought to full tectogrammatical representation and serve as a training set for

[3] Documents from the scientific journal Vesmír are not annotated on the highest, most semantic, tectogrammatical layer.

```
sentence s5 {
  be -> { I.ACT and.CONJ }
  be.ant
  and -> { Poland.LOC Netherlands.LOC
           Argentina.LOC Argentine.LOC
           UK.LOC USA.LOC usa.LOC cz.LOC
           EU.LOC }
  usa ["United States"]
  cz  ["Czech Republic"]
}
```

Figure 2. Dott notation of a tectogrammatical tree structure, one from the determiners regression test set, that gets realized by the current version of English realizer as "I was in Poland, in the Netherlands, in Argentina, Argentine, in the UK, in the USA, in the United States, in the Czech Republic and in the EU."

presented English realizer. However we have not yet assessed the quality of the semi-automatic annotation we work with.

2.3 Tectogrammatical Dot Notation *(dott)*

As DeVault et al [11] reports, the input structures of nowadays available realizers are overly complex and while abstract from the linguistic point of view, they still require too much linguistic expertise for a common application developer to assemble them.

Unfortunately, the usage of tectogrammatical trees as an input structure seems not to overcome common difficulties associated with deploying a general realizer in a particular NLG application.

Experiments with machine translation using syntactico-semantic transfer on the tectogrammatical layer carried out by Bojar [12] and Žabokrtský [13] suggest that it is indeed very demanding for a statistical component to arrive at a decently coherent t-tree after the transfer step.[4]

Our take to a remedy for this input complexity problem is a *tectogrammatical dot syntax* (dott) coupled with preprocessing phase that infers values of all unspecified tectogrammatical atributes from local context of each node. It goes beyond another widely used technique, i.e. introduction of default values that take place when an important attribute is not present in an input structure.

From our initial experiment with English realizer it seems that we are able to reliably deduce the semantic part of speech atribute (i.e. counterpart of Fuf/Surge top-level syntactic category) from the lexical content, dependency structure, and the functor alone.

To do so, we use simple POS statistics gathered from British National Corpus mixed with a bi-node POS+functor model from completed part of PEDT2.0. In near future we plan to investigate Hidden Markov Models adapted to tree structures that are already successfully used in the image processing domain [14].

Outlined simplification of the input structure is coupled with a plain text format, illustrated in Figure 2, and we get a simple notation of syntactico-semantic structures that has already

[4] To support our argument we quote the evaluation from Section 4, where we were able to reach BLEU score of 0.34 on automatically parsed Czech data, while the best English-to-Czech MT system using tectogrammatics scores 0.09 on the Workshop of Machine Translation 2008 e-test data.

proved to be efficient while we have assembled regression suit of tests for the English realizer.

Because the syntax is inspired by a popular tool for noting graphs in so called *dot syntax,* we call the notation *dott* – which stands for 'dot tectogrammatical' (adjective in postnominal position in Latin manner).

3 SENTENCE REALIZATION

In out current implementation, we do not adhere to the unification or pure statistical solution of the realization problem as usual, instead our approach is procedural.

We have decomposed each generation into sequence of several linguistically motivated steps:

Czech System	English System
1. Formeme Selection,	1. Formeme Selection,
2. Agreement,	2. Agreement,
3. Functional Words	3. Inflection,
4. Inflection,	4. Word Ordering,
5. Word Ordering,	5. Functional Words,
6. Punctuation,	6. Punctuation,
7. Vocalization.	7. Determiners.

Such decomposition brings some positives and also negatives in comparison to mentioned symbolic and pure statistical systems. Our grammar of Czech is 'hardwired'; written in the Perl programming language. It is not isolated and reusable as in the case of Fuf/Surge realizer, nor bidirectional as with OpenCCG. On the other hand, procedural design results in swift run-time and quick prototyping.

In our system, the input tectogrammatical tree is gradually changing – in each step, new node attributes and/or new nodes are added. After the last step, the resulting sentence is obtained simply by concatenating word forms which are already filled in the individual nodes, the ordering of which is also already specified.

We discuss selected interesting and non-trivial steps in following paragraphs providing more details.

3.1 Formeme Selection

Formeme selection phase is where the syntactic shape of the sentence is grounded. The input tree is traversed in a depth-first fashion, and a suitable morphosyntactic form is selected from the repertoire of forms available in Czech or English.

Several types of information are used when deriving the value of the new formeme attribute. A valency lexicon is consulted: if the governing node of the current node has a nonempty valency frame, and the valency frame specifies constraints on the surface form for the functor of the current node, then these constraints imply the set of possible formemes. In case of verbs, it is also necessary to specify which diathesis should be used (active, passive, reflexive passive, etc.; depending on the type of diathesis, the valency frame from the lexicon undergoes certain transformations).

The English system uses automatically collected valency dictionary from the completed part of PEDT 2.0.

3.2 Word Ordering

Ordering of nodes in the input t-tree structures can be used to express information structure of the sentences, and does not directly mirror the ordering in the surface shape of the sentence. In the case of Czech system, the word order of the output sentence is reconstructed using syntactic rules (e.g., adjectival attribute goes in front of the governing noun) and topic-focus articulation. Special treatment is required for clitics: they should be located in the 'second' position in the clause (Wackernagel position); if there are more clitics in the same clause, rules for specifying their relative ordering are used (for instance, the clitic by always precede short reflexive pronouns).

In the case of English system we also use a set of syntactic rules to force the subj-verb-obj order for declarative sentences. We continue to look for further word ordering principles.

3.3 Vocalization

Vocalization is a Czech phonological phenomenon: the vowel *-e* or *-u* is attached to a preposition if the pronunciation of the prepositional group would be difficult without the vowel (e.g., *ve výklenku* instead of **v výklenku*).

We use a C5.0 classifier trained on data from Czech National Corpus. We report accuracy yielded on a test set consisting of 200.000 instances of preposition randomly sampled from the corpus.

system	**acc**
baseline (all unvocalized)	85.24 %
manually assembled rule based system [15]	94.86 %
C5.0 classifier	**98.53 %**

4 PRELIMINARY EVALUATION

As an important finding of Langkilde-Geary [16] shows, evaluation results of various surface realizers is almost frailly dependent on the level of detail a particular input structure provides.

As detailed in Section 2, we process input structures that lie on the more abstract end of realizators spectra. This will hold even more after the input in *dott* notation will became a main input for both our realizers. However, 1-reference BLUE scores [17] reported in this section for Czech system, were obtained realizing fully specified tectogrammatical trees from the PDT 2.0 treebank. The English input is more underspecified, as of the time being the manual annotation of PEDT 2.0 data consists only of functors and of the dependency structure. It resembles the information we expect to be obligatory in the *dott* format. Though, the performance of the English system as of now is limited because we lack the data to train on.

realizer	**baseline**	**e-test**	**d-test**
English	0.12 BLEU	0.38 BLEU	
Czech	0.01 BLEU	0.48 BLEU	0.47 BLEU
AutoCzech		0.34 BLEU	

We find the stability of scores reached by Czech realizer on development data as well on evaluation data as encouraging. We suppose that this is mainly due to the use of externally compiled

valency lexicon in the formeme selection phase. However, this is matter of further investigation.

The last system captioned as AutoCzech shows performance on input data that were automatically parsed. This poses an upper bound for any machine translation system using our realizer as last step in the pipeline.

5 RELATED WORK

To put our results in context with other existing systems, we need to introduce another measure commonly reported for realizers in addition to BLEU score.

A *coverage* is obtained as a ratio of sentences for which the realizer has produced strings to number of all inputs in the test set. The need for such a measure stems from the fact, that both the unification and statistical search approach can reach dead ends either because of needed rules missing in the grammar or because of exceeding given time limit (usually 15s) per sentence in case of the search. Our procedural approach always produces a result when run on the PDT or completed part of PEDT treebank; it takes less then two seconds per sentence generated on our development workstation.

Following numbers compare quality of output of English realizers only, as we are not aware of any Czech realizer evaluated on treebank of comparable size as PDT.

OpenCCG reports 0.6615 BLEU score, but on development data and 0.5768 BLEU on Section 23 of Penn Treebank with 94.5% coverage. A more specific input already containing surface syntactic forms, features for some auxiliary words and topic markings helps a realizer of Cahill and van Genabith [18] to reach **0.6651** BLEU score also on Penn Treebank Section 23 with 98.5% coverage.

The evaluation of Fuf/Surge package is not so straightforward. While Callaway [19] reports BLEU of 0.7350 but only with 49.5% coverage, he has than conducted an error analysis on the training section of PTB and reported that "after a period of several months" of hand-tweaking of the rule set that he has reached a BLEU score of **0.9321** at 98.7% coverage. However, the level of detail of automatically constructed input for the realizer is unknown to us.

6 CONCLUSIONS & FUTURE WORK

As of time being, we are working on reimplementation of Czech realizer in order to bring it into the same tree structure API as the English one uses to simplify further parallel development. Then we will introduce support of the *dott* notation also in the Czech version. Another feature of the Czech realizer in work is an experiment where final string of words is being enriched by pitch accents and boundary tones marks to be used by the Arctic Text-to-Speech system [20] we interface. Last but not least is outlined use of HMM adapted for tree structures, that we plan to employ both in *dott* processing and also during the formeme selection phase.

To summarize: we have presented two realizer systems under development. While the Czech system is considered stable and gives reasonably high BLEU scores, the evaluation shows that there is still room for improvement for the English system.

ACKNOWLEDGEMENTS

This work was funded in part by the Companions project (www.companions-project.org), IST-FP6-034434 and by national grants 52408/2008, 7643/2007 and 1ET101120503.

REFERENCES

[1] M. Elhadad and J. Robin. *An overview of SURGE: A reusable comprehensive syntactic realization component*. In Proceedings of the 8th International Language Generation Workshop. (1996)

[2] B. Lavoie and O. Rambow. *RealPro – a fast, portable sentence realizer*. In Proceedings of ANLP'97. (1997)

[3] M. White, R. Rajkumar, and S. Martin. *Towards broad coverage surface realization with CCG.* In Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation UC-NLG+MT, (2007).

[4] P. Sgall, E. Hajičová and J. Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Charles University. (1986).

[5] M. Mikulová, A. Bémová, J. Hajič, E. Hajičová, J. Havelka, V. Kolářová, M. Lopatková, P. Pajas, J. Panevová, M. Razímová, P. Sgall, J. Štěpánek, Z. Urešová, K. Veselá, Z. Žabokrtský and L. Kučová. *Annotation on the tectogrammatical layer in the Prague Dependency Treebank.* Annotation manual. Technical report TR–2005–28. Prague: ÚFAL MFF, Charles University in Prague, (2005).

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

[7] J. Hajič, J. Panevová, Z. Urešová, A. Bémová, V. Kolářová – Řezníčková and P. Pajas. *PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation.* In Proceedings of The Second Workshop on Treebanks and Linguistic Theories. Vaxjo University Press, Vaxjo. (2003)

[8] J. Šindlerová, L. Mladová, J. Toman and S. Cinková. An Application of the PDT-scheme to a Parallel Treebank. In Proc. TLT'07. (2007)

[9] B. Levin. *English verb classes and alternations: a preliminary investigation.* University of Chicago Press. (1993)

[10] M. Marcus, B. Santorini and M. Marcinkiewicz. *Buildinf a large annotated corpus of english: the Penn Treebank.* Computional Linguistics, 19(2). (1993)

[11] D. DeVault, D. Traum and R. Artstein. *Practical Grammar-Based NLG from Examples.* In Proceedings of Fifth International Natural Language Generation Conference. Ohio. (2008)

[12] O. Bojar and J. Hajič. Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. In Proceedings of the Third Workshop on Statistical Machine Translation, Columbus. (2008).

[13] Z. Žabokrtský, J. Ptáček and P. Pajas. TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In Proceedings of the Third Workshop on Statistical Machine Translation, pages 167–170, Columbus, Ohio, (2008).

[14] J.B. Durand, P. Goncalves and Y. Guédon. *Computational methods for hidden Markov tree models - An application to wavelet trees.*In IEEE Transactions on Signal Processing. (2004)

[15] V. Petkevič. (ed) Linguistic Problems of Czech, Vocalization of Prepositions. Report for the JRP PECO 2824 project. (1995)

[16] I. Langkilde-Geary. An empirical verification of coverage and correctness for a general-purpose sentence generator. In Proc. INLG. (2002)

[17] K. Papineni, S. Roukos, T. Ward and W.J. Zhu. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th nnual Meeting of ACL (2001).

[18] A. Cahill and J. van Genabith. Robust PCFG-based generation using automatically acquired LFG approximation. In Proc. SMT Workshop at ACL'07. (2007)

[19] C. Callaway, Evaluating *Coverage for Large Symbolic NLG Grammars*. In Proc. IJCAI-03, (2003)

[20] J. Matoušek, D. Tihelka and J. Romportl. Current state of Czech text–to–speech system ARCTIC. In Lecture notes in computer science. Vol. 4188. Berlin, Heidelberg: Springer–Verlag. (2006).