Using a Treebank Grammar for the Syntactical Annotation of German Lexical Phrases

Marcin Junczys-Dowmunt[*], **Filip Graliński**[†]

[*]Adam Mickiewicz University
Międzychodzka 5, 60-371 Poznań, Poland
junczys@amu.edu.pl

[†]Poleng sp. z o. o.
Głogowska 82/7, 60-741 Poznań, Poland
filip.gralinski@poleng.pl

Abstract

The aim of this paper is to investigate whether a treebank grammar can be used to automatically classify and annotate German phrases contained in a MT lexicon. Phrases from the lexicon appear in their citation form and may differ structurally from the phrase tokens found in the corpus. We describe the grammar extraction process for a formalism called Tree-Generating Binary Grammar and evaluate the performance of subsets of the obtained grammar on a set of four types of lexical phrases.

1. Introduction

This paper is the first in a series of articles describing our experiments with the integration of a treebank extracted grammar into a fully fledged machine translation system. The system we refer to throughout this article is the POLENG system described by (Jassem, 2006), which at present is being extended with an additional language pair — namely German-Polish. In this paper we will show how this extension process can be sped up by using a treebank grammar for a lexicon annotation task.

Apart from single word entries, the POLENG translation lexicon includes pairs of multi-word units that cannot be easily translated by translating its elements compositionally. As in (Jassem and Lison, 2001) we will call such a multi-word unit a *lexical phrase*.

A typical example from our test set is the phrase pair *reinen Wein einschenken / wyłożyć kawę na ławę*. Both equivalents are idiomatized. As we will see in later examples[1] this is not neccessarily the case, since the German phrases are translations of Polish phrasal lexicon entries.

In the POLENG system lexical phrases of the target language are annotated with the syntactical functions they can perform in a sentence. Apart from that, their structure and their heads need to be identified to ensure correct handling in the target language generation process where reorderings of the phrase elements or morphological adjustments are perfomed.

During previous work on other language pairs, lexical phrases have been annotated partly by hand and partly by a parser employing handwritten grammars. For German no such grammar is available for our parser. Designing a grammar by hand is a time-consuming task, we resolve to extract a German grammar from a publicly available treebank — the TüBa-D/Z that consists of annotated sentences from newspaper articles.[2]

The grammar formalism we use is called Tree-Generating Binary Grammar (TgBG) (Graliński, 2006). In order to account for the quantitative information provided by the treebank, we extend the formalism with a probabilistic model.

We will investigate whether a treebank grammar can be used to parse lexical phrases that are provided in their citation form and therefore differ in their morphological structures from the phrase tokens[3] that can be found in newspaper texts.

For instance, infinitive constructions with complements, like *reinen Wein einschenken* seem to pose a problem, since when integrated in a sentence, they span two distinct topological fields[4] and may be hard to identify. Also there may not be many stand-alone phrases of this kind in a general text corpus. Other types of phrases will be described in section 6. where we will also show how we dealt with the problem of parsing those phrases in their citation form.

The necessary preprocessing steps performed on the treebank data as well as the grammar rule generation processes are illustrated. At the end of the article we provide an evaluation of the annotation system obtained on a set of bilingual phrase pairs where the Polish part has already been annotated.

2. The Treebank (TüBa-D/Z)

The third release of the TüBa-D/Z (Telljohann et al., 2006) — the Tübingen Treebank of Written German — comprises 27 125 syntactically annotated sentences with 473 747 tokens in total. The corpus is based on sentences taken from the daily issues of a German newspaper (*die*

[1] Disregarding their Polish source phrases, we will give only the German phrases.

[2] We chose the TüBa-D/Z over another known German treebank — the Tiger Treebank — since TüBa trees are more similar to the trees produced by our parser and probably require less transformation steps than those from the Tiger Treebank.

[3] A phrase that is actually used in a form of communication and does probably not occur in its citation form.

[4] See section 2. and 4.3.

tageszeitung). In order to embrace the syntactical peculiarities of the German language, the developers chose a hybrid annotation scheme that integrates the theory of topological fields with a hierarchical constituent analysis.

Topological fields provide a descriptive framework for the flat top-level sentence structure while the hierarchical structure of the syntactic units located within the boundaries of the fields is described by phrase structure trees.

The part-of-speech tags used in the treebank are taken from the Stuttgart-Tübingen tagset (STTS) (Schiller et al., 1995). Additional morphological tags give information about the inflectional features of the tokens. Syntactical categories and topological fields are indicated by node labels. Edge labels represent the grammatical functions of lexical entries, phrases, topological fields, and clauses, indicating phrase heads or semantic roles, etc. Secondary edge labels are used to resolve ambiguities and indicate some types of long distance dependencies.

3. The POLENG MT System

POLENG (commercial name: Translatica) is a rule-based machine translation system with modules for Polish-to-English, English-to-Polish, Polish-to-Russian, Russian-to-Polish, and Polish-to-German translation (Jassem, 2006). The grammar rules for Polish, English, and Russian, transfer rules as well as the bilingual lexicons were created manually.[5] German is the first POLENG language for which a treebank extracted grammar is applied.

3.1. Tree-Generating Binary Grammars

The grammar formalism called Tree-Generating Binary Grammar is a theoretical idealization of the grammatical framework utilised in the POLENG system (Graliński, 2006). A TgBG rule is composed of three parts: a *production*, a *tree operation*, and an *attribute expression*. Productions are simply CFG-like rewriting rules. Tree operations specify how to assemble a new tree using trees referenced by the symbols of the right-hand side of the production. Tree nodes are labeled with distinct *category symbols* rather than production symbols. It is also possible to assign attributes to tree nodes. Attribute expressions are used to set these attributes and check attribute conditions (such as agreement conditions, see section 5.1.).

3.1.1. Tree-Generating Capabilities

TgBG (with the repertoire of tree operations described in (Graliński, 2006)) is weakly equivalent to CFG although, due to tree operations, it has a greater strong generative power: with tree operations called *left/right attachments* it is possible to generate flat syntax trees (to some extent similar to dependency trees), whereas with *left/right inserts* some discontinuous constructions can be handled.

Subtrees can be labeled with *syntactic roles* such as *subject*, *object*, *modifier*. One special syntactic role called *head* is distinguished.

3.1.2. Attributes and Attribute Expressions

TgBG attributes can have atomic values. The special value `any` is used to specify that any value is acceptable. The following operators are used in attribute expressions of sample TgBG rules given in this paper:

- `,` — conjunction,
- `:=` — simple assignment,
- `=` — simple equality,
- `==` — enhanced equality (`A == B` is true iff `A = B` or `A` is equal to `any` or `B` is equal to `any`),
- `:==` — enhanced assignment (`A :== B` is equivalent to `A := B` if `A` is equal to `any` and equivalent to `A == B`, otherwise),
- `setscore` v — setting the score-value v associated with the rule (see section 3.2.).

By default the attributes of the newly formed node are copied from its head.

3.2. Probabilistic TgBG

So far (i.e. for parsing Polish, English and Russian in the POLENG system) the best syntax tree is determined by summing *scores* assigned to TgBG rules in a heuristic manner (no probabilistic model for parsing has been applied).

In order to use a treebank extracted grammar in an effective manner, probabilities (logarithms of probabilities, strictly speaking) are used instead of heuristic scores. This way we obtain a Probabilistic TgBG (just as we obtain a Probabilistic CFG by assigning probabilities to CFG rules (Manning and Schütze, 1999)).

The probability for a treebank extracted TgBG rule can be calculated automatically by dividing the number of occurrences of constructions described by the rule by the number of occurrences of constructions of a given category.

3.3. The Parser

The TgBG parser used in the POLENG system implements a variant of the bottom-up CKY algorithm (Graliński, 2006). This way a shared-packed forest for a given input sentence can be obtained. The probability of a syntax tree is calculated by summing logprobs of subtrees.

4. Tree Processing

Theory-neutrality has been one of the TüBa developers' main goals as it allows to use the treebank for the extraction of various grammar formalisms. Nevertheless, theory-neutrality is also the source of several problems when adopting the data for a specific formalism.

For the needs of TgBG we are required to perform extensive preprocessing steps on the corpus data to be able to generate rules with appropriate attributes. We decided to transform the structure of the original trees iteratively into trees that are similar to those generated by the TgBG-parser, which can be easily converted into a set of TgBG rules.

[5] The POLENG MT lexicons were based on human-readable dictionaries.

4.1. Tagset and Morphology Conversion

There are noticeable differences between the STTS and the set of nonterminal symbols used in the TgBG, which are determined by the annotation scheme of the POLENG German dictionary. The STTS incorporates syntactical information in the tag name, e.g. PIAT (attributively used indefinite pronoun) and PIS (substitutively used indefinite pronoun). In the TüBa each inflectional word is additionally annotated with a set of morphological features represented by a cluster of single character abbreviations.

The POLENG tagset consists of basic part-of-speech tags with subcategorizing flags, e.g. pron:i (indefinite pronoun), and a set of morphological and syntactical features. Each pair consisting of a STTS-tag and a morphological feature set is converted to a pair consisting of a nonterminal symbol and an *attribute value matrix* (AVM) — a set of feature-value pairs that represent morphological features and subcategorization information consistent with the dictionary and parser of the POLENG system.

4.2. Morphological Propagation

Each set of feature-value pairs obtained by conversion is attached to its corresponding leaf in the original tree. The ancestor nodes of the leaves can be interpreted as empty AVMs that will be filled in the following steps.

Nodes that are marked as phrasal heads propagate their features to the parent nodes they are attached to. If there is no head among a node's children, some chosen feature-value pairs for that node — like case or person — are computed by means of simple set operations. This is justified by the fact that most headless phrases in the treebank contain appositions or coordinating conjunctions.

A special case of headless phrases are topological fields, the features of which are obtained according to a dependency structure we discuss in the next subsection. If no value can be computed we allow for underspecified values (by assigning the special value `any` to an attribute), which can specified during futher processing steps by grammatical agreement, see 5.1. for more details.

4.3. Reconstruction of Basic Dependency Trees

Topological fields are not isolated in a sentence, they influence each other with respect to type and number of constituents they include. E.g. in most cases the LK contains the finite verb while its complements or modifiers can appear in the VF or MF.[6] In another sentence the MF may contain complements of a verbal element from the VC, which again can be a verbal complement of the finite verb in LK.

TüBa marks information of this kind by edge labels, e.g. OA for an accusative object or V-MOD for a verb modifier. Where this information is not explicitly annotated, it must be extracted heuristically.[7]

We extract the verb frames from each clause and reconstruct a basic dependency tree for each finite verb. By doing so, we collect edge labels that are spread over the whole tree into a single compact structure and make implicit dependencies explicit (see Figure 1). These dependencies are used to add and propagate information about complements to the level of the topological fields by adding special feature-value pairs.

4.4. Re-attachment of Punctuation

In TüBa punctuation is tagged, but it is not attached to the syntactic tree. We decided to re-attach punctuation at phrase level since we observed the significant influence of punctuation on the probability distribution of certain types of rules. Many appositions in the corpus contain brackets or quotations marks, which result in unexpected grammar rules when the punctuation is ignored.

The re-attachment is performed automatically according to two simple heuristics: Single punctuation marks (commas, dashes etc.) are attached as children to the lowest common parent node of the immediate left and right neighbours of the punctuation mark. For paired punctuation marks (brackets, quotation marks etc.) we identify two parent nodes according to the previous method and attach both marks to the lower node.

In the graphical representation of a tree, this procedure is equivalent to attaching the punctuation to the lowest horizontal edge that is crossed by a vertical edge drawn from the mark, or one of the marks.

5. Rule Generation

The generation of the context free body of a TgBG rule and its tree operation is straightforward, therefore we focus on the generation of the attribute expressions only.

5.1. Generating Attribute Expressions

We distinguish between three general types of attribute expressions:

1. attributes expressions that model relations between a parent node and its children. These can describe phenomena like morphological propagation;

2. attribute expressions that describe relations between the child nodes of a parent node only. They can be used for checks of morphological agreement or for the specification of underspecified values;

3. attribute expressions that characterize single child nodes. They can restrict features to certain values.

All types of expressions are generated from the AVMs allocated to the given nodes.

[6] In the theory of topological fields, German sentences are categorized according to the position of the finite verb. The most common topological fields are: C (Konjunkt); LK (Linke Klammer); VC (Verbkomplex); VF (Vorfeld); MF (Mittelfeld); NF (Nachfeld). German sentences differ in regard to their field configuration, but not in regard to the word order regularities within the fields. For more detailed information on the theory of topological fields see (Telljohann et al., 2006)

[7] For instance, if the VC contains more than one element, then the unmarked element is probably a verbal complement of the element marked as head. There is no special edge label indicating this fact. Possible ambiguities are resolved by secondary edges when there are more than one non-head elements.

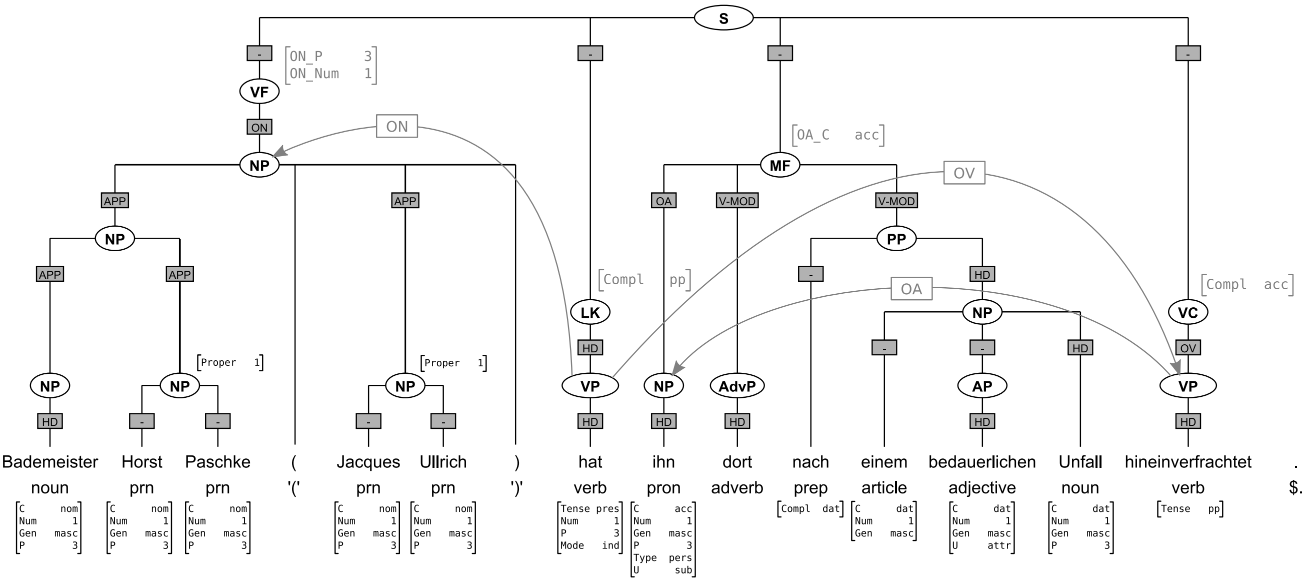

Figure 1: Transformed Tüba tree (s470) with converted label names, AVMs, re-attached punctuation and basic dependency structure (light grey arrows)

We apply a small set of handwritten rules to impose restrictions on the relations that can arise between two nodes. That way we can tell the rule generation algorithm that e.g. the two parts of a nominal conjunction must agree in case, but do not need to agree in gender or number. If the given AVMs contain the features mentioned in the rule, a corresponding argument expression is added to the argument body.

5.2. Example Rules

The following example illustrates morphological agreement in a simple TgBG rule that can be extracted from the sentence represented in Figure 1. It states that a noun phrase (np) consists of an article, an adjective phrase (ap) and a noun (being head of the phrase). The expression contained within percent signs is a specification of a tree operation: the subtrees representing the article, the adjective phrase and the noun will become the children of a new node labeled with category NP. The attribute expression takes care for the agreement in case (C), number (Num) and gender (Gen) between the head noun and the other elements of the phrase (see section 5.1., type 2) and ensure that the adjective phrase contains an attributively used (attr) adjective (type 3). The last expression is a special instruction that encodes the logarithmic probability of the rule.

```
np = article ap noun* %NP[article,ap,noun]%
  C :== ap.C,
  C :== article.C,
  Num :== ap.Num,
  Num :== article.Num,
  Gen :== ap.Gen,
  Gen :== article.Gen,
  ap.U == attr,
  setscore -2.64397688264577
```

The two rules below show how attribute expressions establish consistency checks across the boundaries of topological fields. The AVMs responsible for that rules were generated using the dependency tree described in section 4.1. In the first rule, new attributes are created to store values of a distinguished phrase (type 1) — the subject in this case. The last expression makes sure that only noun phrases in nominative case (nom) can trigger this rule (type 3).

```
vf = np %VF[np:on]%
  ON_Num := np.Num,
  ON_P := np.P,
  np.C == nom,
  setscore -0.71053048605525
```

The newly created attributes are used in the next rule to create the connection between the head of the sentence — the finite verb in the LK-field — and the subject in the VF-field by checking for agreement in number (Num) and person (P). The remaining attributes concern other fields found in the sentence depicted in Figure 1.

```
s = vf lk mf vc %S[vf,lk,mf,vc]%
  lk.Compl == vc.Tense,
  lk.Num == vf.ON_Num,
  lk.P == vf.ON_P,
  vc.Compl == mf.OA_C,
  setscore -2.27476688751752
```

6. Evaluation

The annotation system was tested on a set of four thousand German phrases that are equivalents of Polish words and lexical phrases (taken from the Polish-to-German dictionary). These Polish counterparts have already been classified and annotated partly by an automatic system and partly by hand. We use this information to compare the output of the German annotation tool with the annotation of the Polish phrases. The annotation system is supposed to be able to discriminate between phrases that behave syntactically like nouns, verbs, adjectives, or sentences. That means that we need to be able to find the head of a given phrase in order to classify it as a whole. For instance the German phrase *nicht gewohnt, früh aufzustehen* looks like a verbal phrase, but within a sentence it functions as a predicative adjective, since its head is the participle *gewohnt* and not the infinitive *aufzustehen*. Sentences need to be parsed completely in order to distinguish them from verbal phrases, e.g. *das Buch kann ich nirgendwo bekommen* also features an infinitive construction, but this infinitive is a verbal complement of the finite verb, thus the whole phrase is a sentence.

For phrases that are not sentences, another problem arises. After the head of the phrase has been identified, we need to annotate phrase elements that maintain agreement or government relations of any kind with the head. For noun phrases this would be, e. g., attributively used adjectives, or for verbal phrases — various kinds of complements.

For example, let us consider the German phrase *ein mit einem Rahmen umgebener Text*. Not only should this phrase be classified as a noun phrase with *Text* as its head, but also it should be determined that *ein* and *umgebener* are inflected (as elements of this phrase), whereas *mit*, *einem* and *Rahmen* are fixed words.

The POLENG Polish-to-German dictionary contains over 120,000 German lexical phrases that need to be classified this way. We chose at random 4000 phrases, 1000 from each group, for our evaluation set. We skipped phrases with words that are missing in our German lexicon (as we have not used any module for recognition of unknown German words so far). In our experiments we compared the head categories assigned by the annotation system with the head categories of their Polish counterparts.

6.1. Results

In this section we present the precision measures obtained for our annotation task. In order to increase parser efficiency we use a empirically determined *cut-off probability* of 0.0015 for the TgBG rules. This step results in a deficient probability distribution of the probabilistic model, but does not seem to pose a problems in practise. As a consquence our set of over 10,000 TgBG rules generated from the treebank is reduced to a subset of approximately 700 rules[8]. Using more rules does not seem to improve the annotation precision but slows down the parser. The results are given in Figure 2.

T	NoC	NoCC	%
APs	1000	767	76.70%
NPs	1000	968	96.80%
infinitive VPs	1000	932	93.20%
sentences	1000	903	90.30%
overall	4000	3570	89.25%

Figure 2: Results of the annotation experiment (*T* – phrase type, *NoC* – number of classified phrases, *NoCC* – number of correctly classified phrases)

Returning to the problem of infinitive constructions mentioned in the introduction, it shows that a certain type of German subclauses, the so called *erweiterte Infinitiv,* has a syntactic structure which is similar to that of infinitive phrases in their citation form, though it very rarely appears as a stand-alone sentence, e.g. *Huck und Reinicke boten Jugendlichen an,* ***mit ihnen ein Bühnenprojekt zu realisieren***[9]. Similarly participle constructions that can be interpreted as phrases syntactically equivalent to adjectives are parsed as short sentences (and reinterpreted heuristically as adjective phrases) thanks to another interesting syntactical structure. It shows that lots of headlines of the newspaper articles featured in the corpus consist of such isolated participle phrases, e.g. *Endlich getraut*[10]. Still the recognition of adjective lexical phrases is the most error-prone, which is due to the structural diversity of phrases that can be interpreted that way. It seems, however, that this can be overcome with some additional heuristic rules analyzing the configuration of a phrase.

The numbers prove that these clauses found in the corpus can be reliably used for the recognition and annotation of lexical phrases, especially noun phrases and infinitive verbal phrases.

7. Conclusions

Our experiments show that it is possible to use a parser corpus-trained to annotate German lexical phrases in a MT lexicon, even though the corpus was composed of newspaper sentences rather than lexical phrases. The results are promising for the further development of a universal German parser the more so because no (probabilistic) German POS-tagger has been used so far.

8. References

Graliński, Filip, 2006. Some methods of describing discontinuity in polish and their cost-effectiveness. *Lecture Notes in Artificial Intelligence, Text, Speech and Dialogue, 9th International Conference, TSD 2006*, 4188.

Jassem, Krzystof and Maciej Lison, 2001. Classification, storage and processsing of lexical phrases in polish-english machine translation. *Technologia Mowy i Języka*, 5.

Jassem, Krzysztof, 2006. *Przetwarzanie tekstów polskich w systemie tłumaczenia automatycznego POLENG*. Poznań: Wydawnictwo Naukowe UAM.

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

Schiller, Anne, Simone Teufel, and Christine Thielen, 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report.

Telljohann, Heike, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister, 2006. Stylebook for the Tübingen treebank of written German (TüBa-D/Z).

[8] Which is remarkably close to the number of rules used in the hand written grammars of the other POLENG languages.

[9] Sentence no. 428 in the TüBa-D/Z, the bold faced subclause is the *erweiterte Infinitiv*.

[10] Sentence no. 616 in the TüBa-D/Z.