Towards Case-Based Parsing:
Are Chunks Reliable Indicators for Syntax Trees?

Sandra Kübler
SfS-CL, University of Tübingen
Wilhelmstr. 19
72074 Tübingen, Germany
kuebler@sfs.uni-tuebingen.de

Abstract

This paper presents an approach to the question whether it is possible to construct a parser based on ideas from case-based reasoning. Such a parser would employ a partial analysis of the input sentence to select a (nearly) complete syntax tree and then adapt this tree to the input sentence. The experiments performed on German data from the Tüba-D/Z treebank and the KaRoPars partial parser show that a wide range of levels of generality can be reached, depending on which types of information are used to determine the similarity between input sentence and training sentences. The results are such that it is possible to construct a case-based parser. The optimal setting out of those presented here need to be determined empirically.

1 Introduction

Linguistic similarity has often been used as a bias in machine learning approaches to Computational Linguistics problems. The success of applying memory-based learning to problems such as POS tagging, named-entity recognition, partial parsing, or word sense disambiguation (cf. (Daelemans et al., 1996; Daelemans et al., 1999; Mooney, 1996; Tjong Kim Sang, 2002; Veenstra et al., 2000)) shows that the bias of this similarity-based approach is suitable for processing natural language problems.

In (Kübler, 2004a; Kübler, 2004b), we extended the application of memory-based learning to full scale parsing, a problem which cannot easily be described as a classification problem. In this approach, the most similar sentence is found in the training data, and the respective syntax tree is then adapted to the input sentence. The parser was developed for parsing German dialog data, and it is based on the observation that dialogs tend to be repetitive in their structure. Thus, there is a higher than normal probability of finding the same or a very similar sentence in the training data.

The present paper examines the possibilities of extending the concepts in (Kübler, 2004a; Kübler, 2004b) to unrestricted newspaper text. Since in newspaper text, the probability of finding the same sentence or a very similar one is rather low, the parser needs to be extended to a more flexible approach which does not rely as much on identity between sentences as the original parser.

The paper is structured as follows: Section 2 explains the original parser in more detail, and section 3 describes the treebank used in the investigation. Section 4 investigates whether the chunk sequences used for selecting the most similar sentence in the training data give a reliable estimate of the syntax tree, section 5 investigates properties of tree sets associated with chunk sequences, and section 6 draws conclusions on the architecture of an extended case-based parser.

2 A Memory-Based Parser

The parser in (Kübler, 2004a; Kübler, 2004b) approaches parsing as the task of finding a complete syntax tree rather than incrementally building the tree by rule applications, as in standard PCFGs. Despite this holistic approach to selecting the most similar tree, the parser has a reasonable performance: the first column of Table 1 shows the parser's evaluation on German spontaneous speech dialog data. This approach profits from the fact that it has a more global view on parsing than a PCFG parser. In this respect, the memory-based

	memory-based parser	KaRoPars
labeled recall (syntactic categories)	82.45%	90.86%
labeled precision (syntactic categories)	87.25%	90.17%
F$_1$	84.78	90.51
labeled recall (incl. gramm. functions)	71.72%	
labeled precision (incl. gramm. functions)	75.79%	
F$_1$	73.70	

Table 1: Results for the memory-based parser (Kübler, 2004a; Kübler, 2004b) and KaRoPars (Müller and Ule, 2002; Müller, 2005). The evaluation of KaRoPars is based on chunk annotations only.

parser employs a similar strategy to the one in *Data-Oriented Parsing* (DOP) (Bod, 1998; Scha et al., 1999). Both parsers use larger tree fragments than the standard trees. The two approaches differ mainly in two respects: 1) DOP allows different tree fragments to be extracted from one tree, thus making different combinations of fragments available for the assembly of a specific tree. Our parser, in contrast, allows only one clearly defined tree fragment for each tree, in which only the phrase-internal structure is variable. 2) Our parser does not use a probabilistic model, but a simple cost function instead. Both factors in combination result in a nearly deterministic, and thus highly efficient parsing strategy.

Since the complete tree structure in the memory-based parser is produced in two steps (retrieval of the syntax tree belonging to the most similar sentence and adaptation of this tree to the input sentence), the parser must rely on more information than the local information on which a PCFG parser suggests the next constituent. For this reason, we suggested a backing-off architecture, in which each modules used different types of easily obtainable linguistic information such as the sequence of words, the sequence of POS tags, and the sequence of chunks. Chunk parsing is a partial parsing approach (Abney, 1991), which is generally implemented as cascade of finite-state transducers. A chunk parser generally gives an analysis on the clause level and on the phrase level. However, it does not make any decisions concerning the attachment of locally ambiguous phrases. Thus, the German sentence in (1a) receives the chunk annotation in (1b).

(1) a. In der bewußten Wahrnehmung des
In the conscious perception of the
Lebens sieht der international
life discerns the internationally
angesehene Künstler den Ursprung aller
distinguished artist the origin of all
Kreativität.
creativity.

’The internationally recognized artist discerns the origin of all creativity in the conscious perception of life.’

b. [PC In der bewußten Wahrnehmung des Lebens] [VCL sieht] [NC der international angesehene Künstler] [NC den Ursprung] [NC aller Kreativität].

NCs are noun chunks, PC is a prepositional chunk, and VCL is the finite verb chunk. While for the chunks to the right of the verb chunk, no attachment decision could be made, the genitive noun phrase *des Lebens* could be grouped with the PC because of German word order regularities, which allow exactly one constituent in front of the finite verb.

It can be hypothesized that the selection of the most similar sentence based on sequences of words or POS tags works best for dialog data because of the repetitive nature of such dialogs. The strategy with the greatest potential for generalization to newspaper texts is thus the usage of chunk sequences. In the remainder of this paper, we will therefore concentrate on this approach.

The proposed parser is based on the following architecture: The parser needs a syntactically annotated treebank for training. In the learning phase, the training data are chunk parsed, the chunk sequences are extracted from the chunk parse and fitted to the syntax trees; then the trees are stored in memory. In the annotation phase, the new sentence is chunk parsed. Based on the sequence of chunks, the group of most similar sentences, which all share the same chunk analysis, is retrieved from memory. In a second step, the best sentence from this group needs to be selected, and the corresponding tree needs to be adapted to the input sentence.

The complexity of such a parser crucially depends on the question whether these chunk sequences are reliable indicators for the correct syntax trees. Basically, there exist two extreme possibilities: 1) most chunk sequences are associated with exactly one sentence, and 2) there is only a small number of different chunk sequences, which are each associated with many sentences. In the first case, the selection of the correct tree based on a chunk sequence is trivial but the coverage of the parser would be rather low. The parser would encounter many sentences with chunk sequences which are not present in the training data. In the second case, in contrast, the coverage of chunk sequences would be good, but then such a chunk sequence would correspond to many different trees. As a consequence, the tree selection process would have to be more elaborate. Both extremes would be extremely difficult for a parser to handle, so in the optimal case, we should have a good coverage of chunk sequences combined with a reasonable number of trees associated with a chunk sequence.

The investigation on the usefulness of chunk sequences was performed on the data of the German treebank TüBa-D/Z (Telljohann et al., 2004) and on output from KaRoPars, a partial parser for German (Müller and Ule, 2002). But in principle, the parsing approach is valid for languages ranging from a fixed to a more flexible word order. The German data will be described in more detail in the following section.

3 The German Data

3.1 The Treebank TüBa-D/Z

The TüBa-D/Z treebank is based on text from the German newspaper 'die tageszeitung', the present release comprises approx. 22 000 sentences. The treebank uses an annotation framework that is based on phrase structure grammar enhanced by a level of predicate-argument structure. The annotation scheme uses pure projective tree structures. In order to treat long-distance relationships, TüBa-D/Z utilizes a combination of topological fields (Höhle, 1986) and specific functional labels (cf. the tree in Figure 5, there the extraposed relative clause modifies the subject, which is annotated via the label *ON-MOD*). Topological fields described the main ordering principles in a German sentence: In a declarative sentence, the position of the finite verb as the second constituent and of the remaining verbal elements at the end of the clause is fixed. The finite verb constitutes the *left sentence bracket* (LK), and the remaining verbal elements the *right sentence bracket* (VC). The left bracket is preceded by the *initial field* (VF), between the two verbal fields, we have the unstructured *middle field* (MF). Extraposed constituents are in the *final field* (NF).

The tree for sentence (1a) is shown in Figure 1. The syntactic categories are shown in circular nodes, the function-argument structure as edge labels in square boxes. Inside a phrase, the function-argument annotation describes head/non-head relations; on the clause level, directly below the topological fields, grammatical functions are annotated. The prepositional phrase (PX) is marked as a verbal modifier (V-MOD), the noun phrase *der international angesehene Künstler* as subject (ON), and the complex noun phrase *den Ursprung aller Kreativität* as accusative object (OA). The topological fields are annotated directly below the clause node (SIMPX): the finite verb is placed in the left bracket, the prepositional phrase constitutes the initial field, and the two noun phrases the middle field.

3.2 Partially Parsed Data

KaRoPars (Müller and Ule, 2002) is a partial parser for German, based on the finite-state technology of the TTT suite of tools (Grover et al., 1999). It employs a mixed bottom-up top-down routine to parse German. Its actual performance is difficult to determine exactly because it employed manually written rules. The figures presented in Table 1 result from an evaluation (Müller, 2005) in which the parser output was compared with treebank structures. The figures in the Table are based on an evaluation of chunks only, i.e. the annotation of topological fields and clause boundaries was not taken into account.

The output of KaRoPars is a complex XML representation with more detailed information than is needed for the present investigation. For this reason, we show a condensed version of the parser output for sentence (1a) in Figure 2. The figure shows only the relevant chunks and POS tags, the complete output contains more embedded chunks, the n-best POS tags from different taggers, morphological information, and lemmas. As can be seen from this example, chunk boundaries often do not coincide with phrase boundaries. In the present case, it is clear from the word ordering constraints in German that the noun phrase *des*

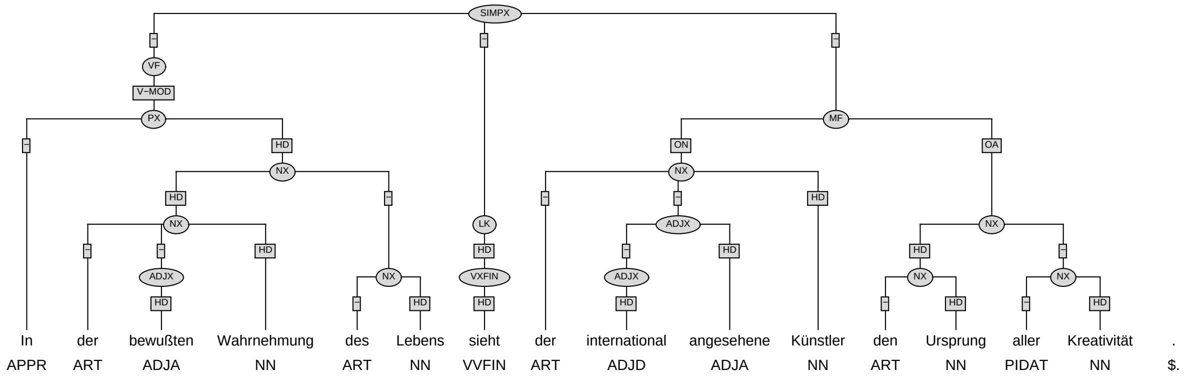

Figure 1: The TüBa-D/Z tree for sentence (1a).

```
<s broken="no">
  <cl c="V2">
    <ch fd="VF" c="PC" prep="in">
      <ch c="PC" prep="in">
        <t f="In"><P t="APPR"></P></t>
        <ch nccat="noun" hdnoun="Wahrnehmung" c="NC">
          <t f="der"><P t="ART"></P></t>
          <t f="bewußten"><P t="ADJA"></P></t>
          <t f="Wahrnehmung"><P t="NN"></P></t></ch></ch>
      <ch nccat="noun" hdnoun="Leben" c="NC">
        <t f="des"><P t="ART"></P></t>
        <t f="Lebens"><P t="NN"></P></t></ch></ch>
    <ch finit="fin" c="VCLVF" mode="akt">
      <t f="sieht"><P t="VVFIN"></P></t></ch>
    <ch nccat="noun" hdnoun="Künstler" c="NC">
      <t f="der"><P t="ART"></P></t>
      <t f="international"><P t="ADJD"></P></t>
      <t f="angesehene"><P t="ADJA"></P></t>
      <t f="Künstler"><P t="NN"></P></t></ch>
    <ch nccat="noun" hdnoun="Ur=Sprung" c="NC">
      <t f="den"><P t="ART"></P></t>
      <t f="Ursprung"><P t="NN"></P></t></ch>
    <ch nccat="noun" hdnoun="Kreativität" c="NC">
      <t f="aller"><P t="PIDAT"></P></t>
      <t f="Kreativität"><P t="NN"></P></t></ch></cl></s>
```

Figure 2: The KaRoPars analysis for sentence (1a). For better readability, the words and the chunk types are displayed in bold.

Lebens needs to be attached to the previous phrase. In the treebank, it is grouped into a complex noun phrase while in the KaRoPars output, this noun phrase is the sister of the prepositional chunk *In der bewußten Wahrnehmung*. Such boundary mismatches also occur on the clause level.

4 Chunk Sequences as Indicators for Syntax Trees

The complexity of the proposed parser depends on the proportion of chunk sequences versus syntax trees, as explained in section 2. A first indication of this proportion is given by the ratio of chunk sequence types and tree types. Out of the 22 091 sentences in the treebank, there are 20 340 different trees (types) and 14 894 different chunk sequences. This gives an average of 1.37 trees per chunk sequence. At a first glance, the result indicates that the chunk sequences are very good indicators for selecting the correct syntax tree. The negative aspect of this ratio is that many of these chunk sequences will not be part of the training data. This is corroborated by an experiment in which one tenth of the complete data set of chunk sequences (test set) was tested against the remainder of the data set (training set) to see how many of the test sequences could be found in the training data. In order to reach a slightly more accurate picture, a ten-fold setting was used, i.e. the experiment was repeated ten times, each time using a different segment as test set. The results show that on average only 43.61% of the chunk sequences could be found in the training data.

(2) Schon trifft sich die Mannschaft erst am
Already meets REFL the team only on the
Spieltag.
game day.
'So the team only meets on the day of the game.'

In a second experiment, we added more information about chunk types, namely the information from the fields *nccat* and *finit* in the XML representation to the chunk categories. Field *nccat* contains information about the head of the noun chunk, whether it is a noun, a reflexive pronoun, a relative pronoun, etc. Field *finit* contains information about the finiteness of a verb chunk. For this experiment, sentence (2) is represented by the chunk sequence "NC:noun VCL NC:refl PC NC:noun PC AVC NC:noun VCR:fin". When using such chunk sequences, the ratio of sequences found in the training set decreases to 36.59%.

In a third experiment, the chunk sequences were constructed without adverbial phrases, i.e. without the one category that functions as adjunct in a majority of the cases. Thus sentence (3) is represented by the chunk sequence "NC VCL NC NC" instead of by the complete sequence: "NC VCL NC AVC AVC AVC NC". In this case, 54.72% of the chunk sequences can be found. Reducing the information in the chunk sequence even further seems counterproductive because every type of information that is left out will make the final decision on the correct syntax tree even more difficult.

(3) Wer gibt uns denn jetzt noch einen Auftrag?
Who gives us anyhow now still an order?
'Who will give us an order anyhow?'

All the experiments reported above are based on data in which complete sentences were used. One possibility of gaining more generality in the chunk sequences without losing more information consists of splitting the sentences on the clause level.

(4) Ganz abgesehen davon, daß man dann schon
Totally irrespective of it, that one then already
mal alle die Geschlechtsgenossinnen kennt, mit
once all the fellow females knows, with
denen man nach der Trennung über den Kerl
whom one after the break-up about the twerp
ablästern kann, weil sie ja genau
slander can, because they already exactly
wissen, wie mies er eigentlich ist.
know, how bad he really is.
'Completely irrespective of the fact that one already knows all the other females with whom one can slander the twerp after the break-up because they already know what a loser he is.'

Thus, the complex sentence in (4) translates into 5 different clauses, i.e. into 5 different chunk sequences:

1. SubC NC:noun AVC AVC AVC NC:noun NC:noun VCR:fin
2. PC NC:noun PC PC VCR:fin
3. SubC NC:noun AVC AJVC VCR:fin
4. SubC AJVC NC:noun AVC VCR:fin
5. AVC VCR:fin PC

The last sequence covers the elliptical matrix clause *ganz abgesehen davon*, the first four sequences describe the subordinated clauses; i.e. the first sequence describes the subordinate clause *daß man dann schon mal alle die Geschlechtsgenossinnen kennt*, the second sequence covers the relative clause *mit denen man nach der Trennung über den Kerl ablästern kann*. The third sequence describes the subordinate clause introduced by the conjunction *weil*, and the fourth sequence covers the subordinate clause introduced by the interrogative pronoun *wie*.

On the one hand, splitting the chunk sequences into clause sequences makes the parsing task more difficult because the clause boundaries annotated during the partial parsing step do not always coincide with the clause boundaries in the syntax trees. In those cases where the clause boundaries do not coincide, a deterministic solution must be found, which allows a split that does not violate the parallelism constraints between both structures. On the other hand, the split into clauses allows a higher coverage of new sentences without extending the size of the training set. In an experiment, in which the chunk sequences were represented by the main chunk types plus subtypes (cf. experiment two) and were split into clauses, the percentage of unseen sequences in a tenfold split was reduced from 66.41% to 44.16%. If only the main chunk type is taken into account, the percentage of unseen sequences decreases from 56.39% to 36.34%.

The experiments presented in this section show that with varying degrees of information and with different ways of extracting chunk sequences, a range of levels of generality can be represented. If the maximum of information regarded here is used, only 36.59% of the sequences can be found. If, in contrast, the sentences are split into chunks and only the main chunk type is used, the ratio of found sequences reaches 63.66%. A final decision on which representation of chunks is optimal, however, is also dependent on the sets of trees that are represented by the chunk sequences and thus needs to be postponed.

5 Tree Sets

In the previous section, we showed that if we extract chunk sequences based on complete sentences and on main chunk types, there are on average 1.37 sentences assigned to one chunk sequences. At a first glance, this results means that for the majority of chunk sequences, there is exactly one sentence which corresponds to the sequence, which makes the final selection of the correct tree trivial. However, 1261 chunk sequences have more than one corresponding sentence, and there is one chunk sequence which has 802 sentences assigned. We will call these collections *tree sets*. In these cases, the selection of the correct tree from a tree set may be far from trivial, depending on the differences in the trees. A minimal difference constitutes a difference in the words only. If all corresponding words belong to the same POS class, there is no difference in the syntax trees. Another type of differences in the trees which does not overly harm the selection process are differences in the internal structure of phrases. In (Kübler, 2004a), we showed that the tree can be cut at the phrase level, and new phrase-internal structures can be inserted into the tree. Thus, the most difficult case occurs when the differences in the trees are located in the higher regions of the trees where attachment information between phrases and grammatical functions are encoded. If such cases are frequent, the parser needs to employ a detailed search procedure.

The question how to determine the similarity of trees in a tree set is an open research question. It is clear that the similarity measure should abstract away from unimportant differences in words and phrase-internal structure. It should rather concentrate on differences in the attachment of phrases and in grammatical functions. As a first approximation for such a similarity measure, we chose a measure based on precision and recall of these parts of the tree. In order to ignore the lower levels of the tree, the comparison is restricted to nodes in the tree which have grammatical functions.

(5) Der Autokonvoi mit den Probenbesuchern
The car convoy with the rehearsal visitors
fährt eine Straße entlang, die noch heute
travels a street down, which still today
Lagerstraße heißt.
Lagerstraße is called.

’The convoy of the rehearsal visitors’ cars travels down a street that is still called Lagerstraße.’

For example, Figure 5 shows the tree for sentence (5). The matrix clause consists of a complex subject noun phrase (GF: ON), a finite verb phrase, which is the head of the sentence, an accusative noun phrase (GF: OA), a verb particle (GF: VPT), and an extraposed relative clause (GF: ON-MOD). Here the grammatical function indicates a long-distance relationship, the relative clause modifies the subject. The relative clause, in turn, consists of a subject (the relative pronoun), an adverbial phrase modifying the verb (GF: V-MOD), a named entity predicate (EN-ADD, GF: PRED), and the finite verb phrase. The comparison of this tree to other trees in its tree set will then be based on the following nodes: NX:ON VXFIN:HD NX:OA PTKVC:VPT R-SIMPX:ON-MOD NX:ON ADVX:V-MOD EN-ADD:PRED VXFIN:HD. Precision and recall are generally calculated based on the number of identical constituents between two trees. Two constituents are considered identical if they have the same node label and grammatical function and if they cover the same range of words (i.e. have the same yield). For our comparison, the concrete length of constituents is irrelevant, as long as the sequential order of the constituents is identical. Thus, in order to abstract from the length of constituents, their yield is normalized: All phrases are set to length 1, the yield of a clause is determined by the yields of its daughters. After this step, precision and recall are calculated on all pairs of trees in a tree set. Thus, if a set contains 3 trees, tree 1 is compared to tree 2 and 3, and tree 2 is compared to tree 3. Since all pairs of trees are compared, there is no clear separation of precision and recall, precision being the result of comparing tree A to B in the pair and recall being the result of comparing B to A. As a consequence only the $F_{\beta=1}$-measure, a combination of precision and recall, is used.

As mentioned above, the experiment is conducted with chunk sequences based on complete sentences and the main chunk types. The average F-measure for the 1261 tree sets is 46.49%, a clear indication that randomly selecting a tree from a tree set is not sufficient. Only a very small number of sets, 62, consists of completely identical trees, and most of these sets contain only two trees.

The low F-measure can in part be explained

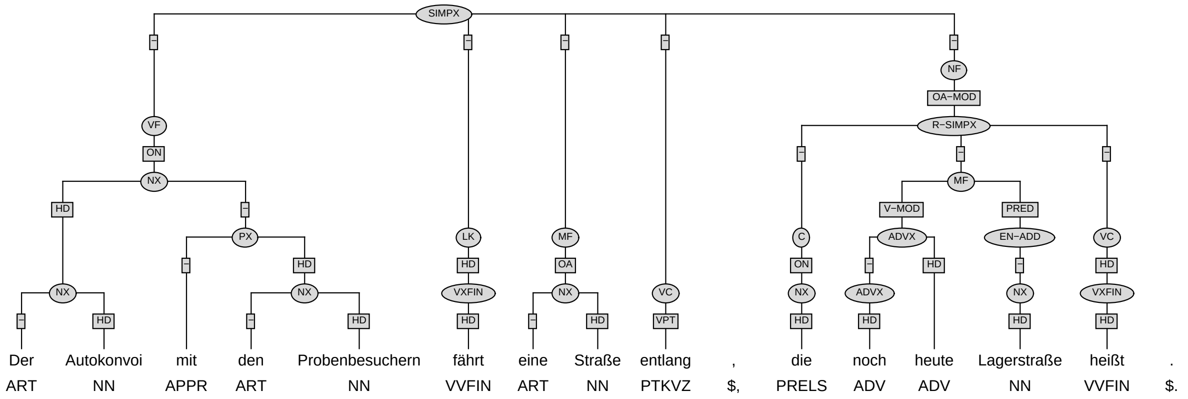

Figure 3: The TüBa-D/Z tree for sentence (5).

by the relatively free word order of German: In contrast to English, the grammatical function of a noun phrase in German cannot be determined by its position in a sentence. Thus, if the partial parser returns the chunk sequence "NC VCL NC NC", it is impossible to tell which of the noun phrases is the subject, the accusative object, or the dative object. As a consequence, all trees with these three arguments will appear in the same tree set. Since German additionally displays case syncretism between nominative and accusative, a morphological analysis can also only provide partial disambiguation. As a consequence, it is clear that the selection of the correct syntax tree for an input sentence needs to be based on a selection module that utilizes lexical information.

Another source of differences in the trees are errors in the partial analysis. In the tree set for the chunk sequence "NC VCL AVC PC PC VCR", there are sentences with rather similar structure, one of them being shown in (6). Most of them only differ in the grammatical functions assigned to the prepositional phrases, which can serve either as complements or adjuncts. However, the tree set also contains sentence (7).

(6) Die Brüder im wehrfähigen Alter
The brothers in the fit for military service age
seien schon vor der Polizeiaktion in die
had already before the police operation into the
Wälder geflohen.
woods fled.

'Those brothers who are considered fit for military service had already fled into the woods before the police operation.'

(7) Das gilt auch für den Umfang, in dem
This holds also for the extent, to which
Montenegro attakkiert wird.
Montenegro attacked is.

'This is also true for the extent to which Montenegro is being attacked.'

In sentence (7), the relative pronoun was erroneously POS tagged as a definite determiner, thus allowing an analysis in which the two phrases *in dem* and *Montenegro* are grouped as a prepositional chunk. As a consequence, no relative clause was found. The corresponding trees, however, are annotated correctly, and the similarity between those two sentences is consequently low.

The low F-measure should not be taken as a completely negative result. Admittedly, it necessitates a rather complex tree selection module. The positive aspect of this one-to-many relation between chunk sequences and trees is its generality. If only very similar trees shared a tree set, then we would need many chunk sequences. In this case, the problem would be moved towards the question how to extract a maximal number of different partial parses from a limited number of training sentences.

6 Consequences for a Case-Based Parser

The experiments in the previous two sections show that the chunk sequences extracted from a partial parse can serve as indicators for syntax trees. While the best definition of chunk sequences can only be determined empirically, the results presented in the previous section allow some conclusions on how the parser must be designed.

6.1 Consequences for Matching Chunk Sequences and Trees

From the experiments in section 4, it is clear that a good measure of information needs to be found for an optimal selection process. There needs to be a good equilibrium between a high coverage of different chunk sequences and a low number of trees per chunk sequence. One possibility to reach the first goal would be to ignore certain types of phrases in the extraction of chunk sequences from the partial parse. However, the experiments show that it is impossible to reduce the informativeness of the chunk sequence to a level where all possible chunk sequences are present in the training data. This means that the procedure which matches the chunk sequence of the input sentence to the chunk sequences in the training data must be more flexible than a strict left-to-right comparison. In (Kübler, 2004a; Kübler, 2004b), we allowed the deletion of chunks in either the input sentence or the training sentence. The latter operation is uncritical because it results in a deletion of some part of the syntax tree. The former operation, however, is more critical, it either leads to a partial syntactic analysis in which the deleted chunk is not attached to the tree or to the necessity of guessing the node to which the additional constituent needs to be attached and possibly guessing the grammatical function of the new constituent. Instead of this deletion, which can be applied anywhere in the sentence, we suggest the use of Levenshtein distance (Levenshtein, 1966). This distance measure is, for example, used for spelling correction: Here the most similar word in the lexicon is found which can be reached via the smallest number of deletion, substitution, and insertion operations on characters. Instead of operating on characters, we suggest to apply Levenshtein distance to chunk sequences. In this case, deletions from the input sequence could be given a much higher weight (i.e. cost) than insertions. We also suggest a modification of the distance to allow an exchange of chunks. This modification would allow a principled treatment of the relative free word order of German. However, if such an operation is not restricted to adjacent chunks, the algorithm will gain in complexity but since the resulting parser is still deterministic, it is rather unlikely that this modification will lead to complexity problems.

6.2 Consequences for the Tree Selection

As explained in section 5, there are chunk sequences that correspond to more than one syntax tree. Since differences in the trees also pertain to grammatical functions, the module that selects the best tree out of the tree set needs to use more information than the chunk sequences used for selecting the tree set. Since the holistic approach to parsing proposed in this paper does not lend itself easily to selecting grammatical functions separately for single constituents, we suggest to use lexical co-occurrence information instead to select the best tree out of the tree set for a given sentence. Such an approach generalizes Streiter's (2001) approach of selecting from a set of possible trees based on word similarity. However, an approach based on lexical information will suffer extremely from data sparseness. For this reason, we suggest a soft clustering approach based on a partial parse, similar to the approach by Wagner (2005) for clustering verb arguments for learning selectional preferences for verbs.

7 Conclusion and Future Work

In this paper, we have approached the question whether it is possible to construct a parser based on ideas from case-based reasoning. Such a parser would employ a partial analysis (chunk analysis) of the sentence to select a (nearly) complete syntax tree and then adapt this tree to the input sentence.

In the experiments reported here, we have shown that it is possible to obtain a wide range of levels of generality in the chunk sequences, depending on the types of information extracted from the partial anaylses and on the decision whether to use sentences or clauses as basic segments for the extraction of chunk sequences. Once a robust method is implemented to split trees into subtrees based on clauses, chunk sequences can be extracted on the clause level rather than from complete sentences. Consequently, the tree sets will also reach a higher cardinality. However, a tree selection method based on lexical information will be indispensable even then. For this tree selection, a method for determining the similarity of tree structures needs to be developed. The measure used in the experiments reported here, F_1, is only a very crude approximation, which serves well for an initial investigation, but which is not good enough for a parser depending on such a similarity measure. The optimal combination of chunk sequences and tree selection methods will have to be determined empirically.

References

Steven Abney. 1991. Parsing by chunks. In Robert Berwick, Steven Abney, and Caroll Tenney, editors, *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, Dordrecht.

Rens Bod. 1998. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, Stanford, CA.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In Eva Ejerhed and Ido Dagan, editors, *Proceedings of the 4th Workshop on Very Large Corpora*, pages 14–27, Copenhagen, Denmark.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–43. Special Issue on Natural Language Learning.

Claire Grover, Colin Matheson, and Andrei Mikheev. 1999. TTT: Text Tokenization Tool. Language Technology Group, University of Edinburgh.

Tilman Höhle. 1986. Der Begriff ”Mittelfeld”, Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany.

Sandra Kübler. 2004a. *Memory-Based Parsing*. John Benjamins, Amsterdam.

Sandra Kübler. 2004b. Parsing without grammar–using complete trees instead. In Nicolas Nicolov, Ruslan Mitkov, Galia Angelova, and Kalina Boncheva, editors, *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, Current Issues in Linguistic Theory. John Benjamins, Amsterdam.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710.

Raymond J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing EMNLP*, pages 82–91, Philadelphia, PA.

Frank Henrik Müller and Tylman Ule. 2002. Annotating topological fields and chunks–and revising POS tags at the same time. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 695–701, Taipei, Taiwan.

Frank Henrik Müller. 2005. *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. Ph.D. thesis, Seminar für Sprachwissenschaft, Universität Tübingen. Version of 16th Nov. 2005.

Remko Scha, Rens Bod, and Khalil Sima’an. 1999. Memory-based syntactic analysis. *Journal of Experimental and Theoretical Artificial Intelligence*, 11:409–440. Special Issue on Memory-Based Language Processing.

Oliver Streiter. 2001. Recursive top-down fuzzy match, new perspectives on memory-based parsing. In *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation, PACLIC 2001*, Hong Kong.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2004. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235, Lisbon, Portugal.

Erik F. Tjong Kim Sang. 2002. Memory-based named entity recognition. In *Proceedings of CoNLL-2002*, pages 203–206. Taipei, Taiwan.

Jorn Veenstra, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities, Special Issue on Senseval, Word Sense Disambiguation*, 34(1/2):171–177.

Andreas Wagner. 2005. *Learning Thematic Role Relations for Lexical Semantic Nets*. Ph.D. thesis, Universität Tübingen.