Checking treebank consistency to find annotation errors

Kaarel Kaljurand

May 2, 2004

Contents

1 Introduction

This report describes a tool which helps checking the consistency of syntactic annotation in treebanks. Such annotation usually groups words in a sentence into phrases (e.g. NP, VP, etc), often also specifying the syntactic functions (e.g. subject, object etc) of the words in the phrases.

Most of the annotation work is done manually, the number of different annotators can be large, their familiarity with the underlying linguistic theory can be different, and the theory itself is likely to be modified as new texts (of different nature and unpredicted challenges) are introduced. Therefore, the treebank can come out erroneous. Errors in the annotation can be signalled by the inconsistency of the annotation. E.g. if the same sequence of words is annotated differently in different sentences, then it might mean that the annotator has made a mistake.

Our intention here is not to provide one single mechanism for treebank consistency checking. Instead, we describe a tool that can be used in many different ways which could reveal inconsistencies in the treebank. The exact usage has to be determined by the linguist in charge of checking the treebank. Note that the approach presented here is likely to be a bit specific to checking the NEGRA corpus, as this corpus has been our main test suite.

2 Related work

There is surprisingly little literature on consistency checking of treebanks.

[Brants and Skut, 1998] describe an annotation strategy where most of the annotation decisions are made by an underlying statistical parser (namely: detection and naming of phrasal structures and specifying the syntactic functions of the words). Human annotator has to specify only some of the hierarchical structure of the sentence, starting from bottom up, or select the annotation from

a list that the parser has recommended (in case the automatic decision is not statistically reliable). In addition, the parser is constantly learning from the growing annotated corpus by modifying its statistical information. As this way of annotation can still lead to errors (since the human annotator has still several decisions to make), two people are employed to annotate the same sentences, so that an inter-annotator agreement could be calculated.

This is clearly an elegant approach, which loses the need for automatic final checking of the annotation. (If such checking is possible then it should be incorporated into the annotation process.) Still, it is not quite clear how the approach would work for annotation of small corpora with tagset which is not fully determined (e.g. starting to build a corpus for a less studied language).

[Kordoni, 2003] report their experiences from the work on English treebank of spontaneous speech data. They summarize the methods of arriving at a consistently annotated treebank, by mentioning stylebooks, automatic partial parsers and inter-annotator agreement. The partial parsing approach proved to be the most useful as the treebank was annotated by only one annotator. They also (shortly) describe an automatic consistency checker that, for a given phrase type, extracts all strings of words in the corpus that had been annotated with this phrase type.

[Dickinson and Meurers, 2003] have successfully detected syntactic annotation errors in Penn Treebank. Their method is to index all possible word n-grams in the treebank, mapping them to their phrasal category name (or `NIL` if the n-gram is not annotated as a phrase). Detecting an n-gram which is annotated differently in the same context, signals an annotation error. The reliability of such discovery is bigger the longer the context is.

One of the weaknesses of this method could be its lexical nature. Only the annotation of word n-grams is checked, which means that consistency problems can be found only in very large treebanks. Also, several types of inconsistency are never discovered, e.g. the consistency of grouping phrase categories (e.g. NP and VP) is not checked. Also, the nature of the inconsistency (how many different categories were applied to the same string of words? what is their frequency distribution?) is not analyzed.

3 Consistency checking

The basic idea of the sentence annotation in a treebank is to group lexical items. (At least in NEGRA treebank) each group can be described by the group name (phrase category) and its syntactic function in a higher-level group. Each lexical item participates in a group together with its POS-tag and syntactic function.

Treebank annotation and exploration tools such as Annotate[1] and TIGERSearch[2] focus on the sentences in the treebank. The user can go through the treebank one sentence at a time (possibly filtering out sentences that don't match his interests by formulating a search query) and the full details of the sentence structure are presented to him. The problem with such way of treebank exploration is that an overview of how a string of words (or syntactic functions) has been annotated throughout the treebank, is missing. E.g. the annotator can see that syntactic labels `MO` and `HD` can be grouped under phrase label `VP`, but whether or not it is always the case, is not explicitly presented (see Fig. 1)

Given a fully annotated treebank, the basic idea behind the consistency checking that we present here, is to restructure the treebank in a way that the focus is taken away from the sentences and given to the groups. Groups can be formed by either words, POS-tags or syntactic function labels. For each such group a set of its possible annotations in the treebank is extracted (see Table. 1).

[1] `http://www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html`
[2] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/`

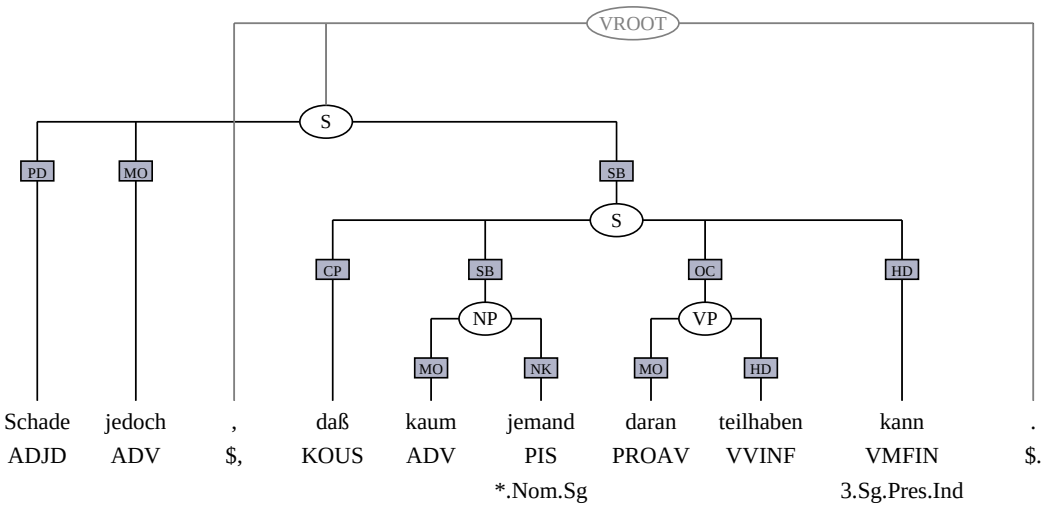

Figure 1: Annotation example from NEGRA treebank displayed by TIGERSearch

Phrase	Frequency	Location (sentence ID)
AP	69	11,24,24,27,41,...
VP	21	5,41,83,95,150,...
AVP	20	52,91,91,110,114,...

Table 1: Grouping of MO and HD in 500 sentences from NEGRA treebank

When checking the annotation of a group, we must also take its context into account. As the context determines the linguistic nature of the group, the annotation of the group should vary less the longer the context. Violation of this rule can signal annotation errors.

The restructured data can be further sorted, filtered and visualized to highlight the possible annotation errors. Groups which have corpus evidence of being annotated by only a single category should be filtered out, since we have no reason to doubt their annotation. Groups for which the annotation varies, should be sorted by the nature of the variation.

We use a measure which evaluates the skewness of the frequency distribution of all the categories applied to the group in the treebank. It is defined as:

$$skewvalue(g) = \begin{cases} \sum_{c \in C} (f_c - mean(f_C))^2 & if\ |C| > 1 \\ -1 & if\ |C| = 1 \end{cases} \quad (1)$$

where g is a group, C is a set of categories used for g throughout the treebank, f_c is the frequency of category c, $mean(f_C)$ is the statistical mean of the frequencies of all the categories, $|C|$ is the number elements in set C.

Groups whose skew value is -1 have been annotated consistently, we have no ground to doubt their annotation. Other values could signal an error. Groups which get a high skew value have been usually annotated by one category, the other categories occur seldom. This could lead to discovering typos. On the other hand, groups which receive a value which is close to 0, have been annotated with

several categories with equal frequency. This could lead to a problem with a bad tagset (it is not clear for the annotator which category to choose, so he chooses at random).

The skew value presented here is still experimental. One of the problems with the high skew value is that a group annotated perfectly throughout the corpus can get a high value, simply because, in the language in general, the functions/meanings of a linguistic unit (e.g. string of words) have a skewed frequency distribution.

4 Implementation

The described approach to consistency checking was implemented in Perl. It is available on the web[3].

The tool expects its input to be in NEGRA format (see [Brants, 1997]). The output is in simple text format: it lists all the groups in the treebank together with their skew value and all possible annotations in a given context (see Fig. 2)

```
NE#NE#[NN]
        8
                5       MPN     26,125,129,281,399
                1       MTA     359
```

Figure 2: Output example. The group which is formed by words with POS-tags `NE` and `NE` has been annotated in two different ways in the context of the word with POS-tag `NN`. Phrase category label `MPN` is used 5 times (in sentences 26, 125, etc) and phrase category label `MTA` is used once (in sentence 359). The skew value calculated for the group is 8.

The tool expects its input to be in STDIN and writes the output to STDOUT. The nature of the output is determined by the commandline parameters.

- **key** determines the nature of the groups. It can be 'word' (for groups of words to be output), 'edge' (for edge labels), 'pos' (for POS-tags) and 'morph' (for morphological tags). Defaults to 'word'.
- **value** determines the nature of the group identifier. It can be 'node' (for node label) or 'edge' (for edge label). Defaults to 'node'.
- **lc** determines the amount of left context in which the group annotation consistency is measured. It can be any natural number, although values which would cross the sentence border do not make sense. Defaults to '0'.
- **rc** is analogical to **lc**, determining the amount of right context. Defaults to '0'.
- **context** determines the nature of the context. It can be 'word' (for context words to be output), 'edge' (for edge labels), 'pos' (for POS-tags) and 'morph' (for morphological tags). It makes sense only if either **lc** or **rc** is larger than '0'. Defaults to the same value as **key**.

Applying the program to file 'tb.negra' (a subset of NEGRA corpus) in the following way

```
consistency.pl --context pos --lc 1 --rc 2 < tb.negra > tb.cons
```

[3] `http://psych.ut.ee/~kaarel/Programs/Treebank/ConsistencyChecking`

produces a file ‘tb.cons’ which lists all the word-groups in ‘tb.negra’ with their phrase-annotation in the context of one POS-tag to the left and two POS-tags to the right (in case the group covers the whole sentence, then no context elements can be added).

```
...
FRANKFURT#A.#M.
        0
                1       MPN     454
                1       ROOT    454
...

[KOUS]#einer#von#ihnen#[ADJD]#[VAFIN]
        -1
                1       NP      235
...
```

From this output segment the linguist can decide that in sentence 454 the string “FRANKFURT A.M.” has been grouped twice (this, of course, might have been intentional). The string “einer von ihnen” has been annotated only once in the context of POS-tags `KOUS`, `ADJD`, `VAFIN`.

5 Conclusions and future work

The work presented here is a simple approach to consistency checking of syntactic annotation in treebanks that can possibly be extended.

As we check only the consistency of annotation of groups, we do not discover problems with a lack of structure (i.e. ungrouped strings). E.g. if a string of sentence elements “A B C” is annotated with different structures $[[AB]C]_S$, $[A[BC]]_S$ and $[ABC]_S$ the output of the tool would not reveal this. Similarly, when part of the string participates in another structure, e.g. $[XA][BC]$, it will not be indexed.

Currently the tool has been only applied to small segments of well-checked treebanks (e.g. $\sim$500 sentences from NEGRA treebank). Therefore its usefulness has to be still evaluated.

References

[Brants, 1997] Brants, Thorsten, 1997. The NeGra Export Format. Claus report #98, Saarland University, Computational Linguistics, Saarbrücken.

[Brants and Skut, 1998] Brants, Thorsten and Wojciech Skut, 1998. Automation of treebank annotation. In *New Methods in Language Processing (NeMLaP-98)*. Sydney, Australia.

[Dickinson and Meurers, 2003] Dickinson, Markus and W. Detmar Meurers, 2003. Detecting inconsistencies in treebanks. In *The Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*. Växjö, Sweden.

[Kordoni, 2003] Kordoni, Valia, 2003. Strategies for annotation of large corpora of multilingual spontaneous speech data. In *Workshop on Multilingual Corpora: Linguistic Requirements and Technical Perspectives held at Corpus Linguistics 2003*. Lancaster.