

# Transformation-Based Learning of Danish Grammar Correction

Daniel Hardt

Department of Computational Linguistics, Copenhagen Business School  
Department of Computing Sciences, Villanova University  
dh@id.cbs.dk

**Paper ID:** 20

**Keywords:** grammar checking, tagging, transformation-based learning, comma, agreement

**Contact Author:** Daniel Hardt

## Abstract

We describe a technique for using the Brill Tagger to learn to identify grammar errors. We have applied this technique to two types of Danish grammar errors: incorrect commas, and incorrect article-noun agreement. The system identifies comma errors with a precision of 91%, while agreement errors are identified with 95% precision, with many of the system errors resulting from deficiencies in the tagger's lexicon, due to the small size of the training corpus. The technique is quite general, and could be directly applied to a large number of grammar checking problems in Danish or other languages.

# Transformation-Based Learning of Danish Grammar Correction

## Abstract

We describe a technique for using the Brill Tagger to learn to identify grammar errors. We have applied this technique to two types of Danish grammar errors: incorrect commas, and incorrect article-noun agreement. The system identifies comma errors with a precision of 91%, while agreement errors are identified with 95% precision, with many of the system errors resulting from deficiencies in the tagger's lexicon, due to the small size of the training corpus. The technique is quite general, and could be directly applied to a large number of grammar checking problems in Danish or other languages.

## 1 Introduction

We describe a general technique for automatically deriving grammar checkers, using the Brill Tagger (Brill, 1994). For a given type of error, error occurrences are systematically generated, and special tags are used to identify the correct and incorrect forms. Then the tagger is trained to learn contexts where errors can be identified. The standard context rule learning system from the Brill Tagger is used. This system can learn context rules involving the preceding and following three words, and their tags.

We have applied this technique to two types of grammar errors in Danish: article-noun agreement and comma placement. The resulting system for article-noun agreement achieves precision of 95%, with many of the remaining errors resulting from omissions in the lexicon. Comma placement is a much more difficult problem, but the system still achieves a precision of over 91%.

## 2 General Technique

We begin with a part of speech tagged corpus and a grammatical problem. We have trained the Brill Tagger on the 290,000 word manually tagged Danish PAROLE corpus (Parole, 1998). We term this the **Base Tagger**. We can define a grammatical problem as a choice among

a set of lexical items, which we call the Confusion Set.<sup>1</sup> For simplicity, we restrict attention to cases where the Confusion Set contains only two items, although generalization to larger sets is straightforward. Thus we can describe a grammar problem as a choice between *lexitem1* and *lexitem2*. We tag every occurrence of *lexitem1* with a unique tag, *ITEM1*, and every occurrence of *lexitem2* with a unique tag *ITEM2*. We systematically introduce errors into the corpus by changing some occurrences of *lexitem1* to *lexitem2*, and some occurrences of *lexitem2* to *lexitem1*.

We then produce training material for the Brill Tagger's context learning system, *Contextual-Rule-Learn*. This system takes two files as input, called *Truth* and *Dummy*. *Truth* is correctly tagged, while *Dummy* is not. The system attempts to find rules which can be used to make *Dummy* resemble *Truth* as much as possible. We begin by making two copies of the tagged corpus. We can leave the *Truth* file unchanged,<sup>2</sup> while making changes of the following form to *Dummy*:

```
lexitem1/ITEM1 -> lexitem2/ITEM2  
lexitem2/ITEM2 -> lexitem1/ITEM1
```

Thus the correct tags are retained in *Truth*, but are not retained in *Dummy*. Note that we are assuming that every occurrence in the original corpus is correct, while any introduced change is incorrect.

Next, we run *Contextual-Rule-Learn* with *Truth* and *Dummy* as input. The context rule learner attempts to learn patterns in which *Dummy* can be changed to more closely resemble *Truth*: in this case, the only differences are cases where tag *ITEM1* should be changed to *ITEM2*, or vice versa. The result is an ordered list of rules which give contexts in which these

---

<sup>1</sup>See work on similar problems in (Golding and Schabes, 1996) and references therein.

<sup>2</sup>In building the comma error corrector, we do not leave *Truth* unchanged, as described below. Instead, errors are introduced in both files. They are tagged as errors in the *Truth* file, but not in the *Dummy* file. What is crucial is that all tags are correct in *Truth*, while some tags relevant to the error of interest are incorrect in *Dummy*.

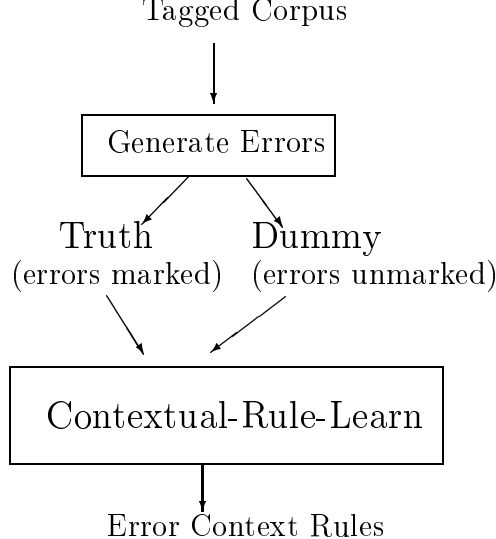


Figure 1: Training the Error Corrector System

changes should be made. We call these rules *Error Context Rules* (see Figure 1).

The tagger can be run using *Error Context Rules*, and we term this system the **Error Corrector**. We now have a system for correcting errors in raw text, consisting of the following processing phases (see Figure 2):

1. **Base Tagger:** Tagger with Part of Speech Rules Learned from Danish Parole Corpus
2. **Item Tagger:** tag *lexitem1* with ITEM1, *lexitem2* with ITEM2
3. **Error Corrector:** Tagger with Error Context Rules
4. **Display**

Using this technique, we can construct an Error Corrector for an arbitrary grammar problem, using the Brill Tagger software unchanged.

### 3 Article Noun Agreement

In Danish, nouns are divided into two genders: common and neuter. Articles and adjectives must agree in gender with nouns. Here, we focus on the agreement of the indefinite article with nouns. The neuter form of the indefinite article in Danish is *et*, while the common form is *en*.<sup>3</sup>

<sup>3</sup>It should be noted that there are several other cases of article-noun gender agreement in Danish which we are not addressing, including definite articles and demonstratives.

Neuter nouns include, for example, *hus(house)*, *kamera(camera)*, *ord(word)*. Common nouns include *mand(man)*, *hund(dog)*, *bil(car)*.

To learn rules for article-noun agreement, we made two copies of the manually tagged PAROLE corpus, where each line is repeated three times. This copy is called *Truth*. The second copy, called *Dummy* is modified as follows. The first line is left unchanged. In the second line, any occurrences of “en” are changed to “et”. In the third line, any occurrences of “et” are changed to “en”. We label each occurrence of “et” with the tag ET, and each occurrence of “en” with the tag EN. Consider the following constructed example, with irrelevant tags removed.

#### Truth

```
En/EN mand har et/ET hus.
En/EN mand har et/ET hus.
En/EN mand har et/ET hus.
```

#### Dummy

```
En/EN mand har et/ET hus.
En/ET mand har et/ET hus.
En/EN mand har et/EN hus.
```

Note that, in the second line of *Dummy*, an EN tag has been changed to an ET tag, and in the third line, an ET tag has been changed to an EN tag. Each sentence of the 290,000 word corpus is treated in this fashion. Then the Context Rule learner of the Brill tagger is run, resulting in *Error Context Rules* for article noun agreement.

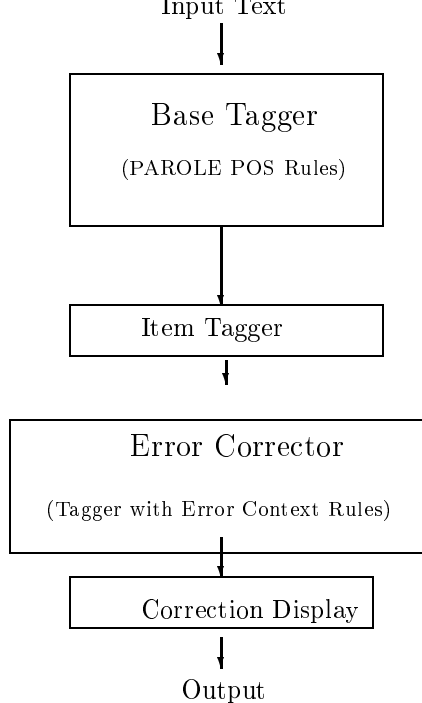


Figure 2: Error Correction System

The following are the first ten rules learned

1. ET  $\rightarrow$  EN if one of the three following tags is N(common-sing)
2. EN  $\rightarrow$  ET if one of the three following tags is N(neuter-sing)
3. ET  $\rightarrow$  EN if one of the three following tags is Adj(common-sing)
4. ET  $\rightarrow$  EN if the next tag is N(common-sing)
5. EN  $\rightarrow$  ET if one of the two following tags is N(neuter-sing-genitive)
6. EN  $\rightarrow$  ET if one of the two following tags is N(neuter-plural)
7. EN  $\rightarrow$  ET if the next tag is Adj(neuter-sing)
8. ET  $\rightarrow$  EN if the next tag is N(common-sing-genitive)
9. EN  $\rightarrow$  ET if the next tag is N(neuter-sing)
10. ET  $\rightarrow$  EN if one of the two following tags is Pronoun(common-sing)

These rules describe conditions under which the first tag is changed to the second.

### 3.1 Resulting System

As described in Section 2, we construct an agreement checker in a very simple way: input is first tagged by the Base Tagger, and then the Item Tagger tags all occurrences of “en” with EN, and all occurrences of “et” with ET. Next, errors are corrected using the *Error Corrector*, and finally errors are displayed: these are simply cases where an “en” word has an ET tag, or vice versa. Error Display shows errors in this form: en(et) – for “en” corrected to “et”, or et(en) – for “et” corrected to “en”.

For illustration, here is a sample run:

#### Input

```

En mand har et hus .
Et mand har et hus .
En mand har en hus .
  
```

#### Output

```

En mand har et hus .
et(en) mand har et hus .
En mand har en(et) hus .
  
```

### 3.2 Results and Analysis

The system was tested on a file of 162876 words, extracted from the Bergenholtz corpus (Bergen-

holtz, 1988). The file contained 527 occurrences of “et” and 1098 of “en”. Two new files are created, both consisting of two concatenated copies of the original file. One new file is called “truth”, and is left unchanged. Errors are generated in the other file as follows: in the first half, all occurrences of “en” are changed to “et”, and in the second half, all occurrences of “et” are changed to “en”. This gives a total of 1625 differences, or errors.

The test resulted in 1454 proposed corrections from et-en or en-et. Of these, 1375 correctly identified errors, while 79 were incorrect. This gives a precision of 95%. Since there were a total of 1625 errors, the recall is 85%.

Many system errors can be directly linked to limitations resulting from the small size of the original training corpus (ie., the 290,000 word Danish PAROLE corpus). As part of the training process, the tagger builds a lexicon consisting of every word appearing in the training corpus, together with a list of all the possible tags for each word. This is a crucial piece of information for the agreement checker, since the gender of the noun can be directly inferred from the list of possible tags for any given noun. The most frequent cause of error appears to be where the noun does not appear in the lexicon. Below is a list of the first 5 incorrectly labeled *en* errors:<sup>4</sup>

- et(en) vidunder *a wonder*
- et(en) øjeblik *a moment's*
- et(en) statsapparat *a state apparatus*
- et(en) ordentligt møgfald *a big scolding*
- et kvæk : et(en) kvæk *a croak*

The nouns in each of these error cases (*vidunder*, *øjeblik*, *statsapparat*, *møgfald*, and *kvæk*) are missing from the system’s lexicon. Thus, it appears that an increase in the training data would be the simplest way to improve for the system, since it would result in an expansion of the lexicon. On the other hand, some errors involve rather difficult, somewhat idiosyncratic constructions, such as the following:

et provokerende “husets-herre-venter-på-at-blive-opvartet”-attitude

<sup>4</sup>These are cases where the system incorrectly changed *et* to *en*.

*a provocative “the-house’s-master-being-served”-attitude*

Here, the *et* is incorrectly left unchanged by the system (the original text had *en*). The system attempts to find the expression *husets-herre-venter-på-at-blive-opvartet* in its lexicon, and fails.

Overall, this analysis suggests that the technique described here, despite its simplicity, is remarkably successful in producing a solution to the article-noun agreement checking problem. It appears that many system errors will be corrected by simply producing a reasonably complete lexicon. Other remaining errors result from idiosyncratic constructions that will perhaps remain beyond the scope of currently available techniques for automatic detection.

## 4 Comma Correction

Here, we attempt to construct a system that will identify incorrect commas, i.e., commas that should be deleted. We apply essentially the same approach as described above for the agreement problem, although some details in the training materials differ, as described below.

We produced the training file by tagging 600,000 words of text from the Bergenholtz corpus, using the Base Tagger. We converted the tags to the Reduced Parole Tag Set. This was done to facilitate the learning of generalizations such as “no comma between a preposition and a noun”. In the original tag set, there are 23 tags for common nouns, because of differences in number, gender, etc. In the reduced tagset, there are just two: N (common noun), and N\_GEN (genitive noun). Other categories have similarly reduced numbers of tags. Note that the reduced tagset could not have been used for the agreement problem, since information about noun gender would be lost.

Additional commas were introduced at random positions in the training file. These additional commas are considered errors, and are labeled with the tag BC (bad comma) in the *Truth* file, while the original commas are tagged GC (good comma). The *Dummy* file is identical to *Truth*, except that all commas are tagged GC. Thus what the system learns is contexts in which a comma’s tag should be changed from GC to BC, and in this way marked as an error. The

list of such contexts is produced by the learner as an ordered list of rules, specifying when the comma tag should be changed. It is important to note that these rules are ordered, so that a decision specified by a rule early on the list will sometimes be reversed by a rule later on the list.

In all, 166 *Error Context Rules* for commas were produced. The first 13 rules are shown below:

1. GC → BC if one of the three following tags is End-of-sentence
2. GC → BC if one of the two previous tags is Beginning-of-sentence
3. GC → BC if the next tag is Preposition
4. GC → BC if one of the two following tags is Verb(Infinitive)
5. GC → BC if the previous tag is Conjunction
6. BC → GC if the previous tag is Interjection
7. GC → BC if the previous tag is Preposition and the following tag is N
8. GC → BC if one of the two previous tags is Subordinating Conjunction
9. GC → BC if the previous tag is Pronoun and the following tag is N
10. GC → BC if the previous tag is Verb(past) and the following tag is Pronoun(personal)
11. BC → GC if one of the next two tags is Subordinating Conjunction
12. GC → BC if the previous word is *er* (is)

The first two rules state that a comma is marked bad ("BC") if it is within 3 words of the end of a sentence, or within 2 words of the beginning of the sentence. These rules were learned because there were comparatively few correct commas in these environments in the truth file, and a large number of incorrect commas in these environments. However, the system soon learns that these rules are overly general. For example, the sixth rule states that a comma is correct if preceded by an interjection (INTERJ). This occurs typically near the beginning or end of a sentence, as in the following example from the training corpus:

```
Naa/INTERJ ,/GC I/PRON_PERS sidder/V_PRES stadig/RGU
Well , you sit still
og/GC hygger/V_PRES jer/PRON_PERS ./XP
and enjoy yourselves.
```

Rule 7 doesn't permit commas between prepositions and nouns, and Rule 8 doesn't permit commas near the beginning of a subordinate clause. This is related to the fact that a comma typically introduces a subordinate clause in Danish. This fact is partially captured in Rule 11, which permits commas just before subordinating conjunctions. Rule 9 disallows commas between a Pronoun and Noun. In the Parole corpus, there is no category for Determiner, and words like "the" and "a" are tagged as pronouns.

#### 4.1 Resulting System

We build a comma correction system using our general technique: the Base Tagger is run, followed by the comma Error Corrector, which is the tagger together with the tagger with the comma Error Context Rules. Comma errors are those which are tagged with BC in the output. Here is a sample run of the system, with different comma positions in the sentence *Det er godt, at du kom* (It is good, that you came):

##### Input

```
Det er godt, at du kom.
Det er godt at, du kom.
Det er godt at du, kom.
Det, er godt at du kom.
Det er, godt at du kom.
```

##### Output

```
Det er godt , at du kom .
Det er godt at ,/BC du kom .
Det er godt at du ,/BC kom .
Det ,/BC er godt at du kom .
Det er ,/BC godt at du kom .
```

Of the five different comma positions, only the first is correct in Danish. The system correctly labels all the other alternatives as incorrect (BC).

#### 4.2 Results and Analysis

The system was tested with a file of distinct text from the Bergenholtz corpus, containing 14,044

words. The file contains 869 commas. 389 additional commas were introduced in random positions, as errors. The system marked 327 commas as errors, of which 299 actually were errors. This gives a precision of 91.4% and a recall of 76.9%.

Here is a list of the first 10 examples where the system incorrectly marked a comma as an error:

1. Hulgaard/EGEN ,/BC Århus/EGEN
2. mener/VPRES ,/BC vi/PRONPERS
3. mener/VPRES ,/BC han/PRONPERS
4. menneskemassen/VPRES ,/BC der/UNIK
5. 17-13/NUM ,/BC Norris-Paulsen/N
6. morderiske/VPRES ,/BC  
psykopatiske/VINF
7. Sørensen/EGEN ,/BC Århus/EGEN
8. nabokommunen/N ,/BC på/SP
9. systemet/N ,/BC kan/VPRES
10. de/PRONDEMO aktive/ADJ ,/BC servicefunktionerne/N

In items 1 and 7 a line break was incorrectly placed immediately before the text in question. Items 4 and 6 involve mistagging: *menneskemassen* (“mass of people”) and *morderiske* (“murderous”) are both nouns, mistagged as verbs. Item 10 is an interesting case *De aktive, servicefunktionerne* (the active, serviceworkers). The comma is marked incorrect because of the following rule:

- GC → BC if the previous tag is ADJ and the next tag is N

This is normally correct; commas don’t tend to appear between an ADJ and a N. Here, however, “the active” is a complete NP, on par with, eg, “the rich”, and “serviceworkers” is a separate NP.

## 5 Conclusions and Future Directions

We have shown how the Brill Tagger can be used to automatically derive grammar checkers for two important problems in Danish grammar: article-noun agreement, and comma placement.

The technique is quite general, and can readily be applied to any grammar checking problem that can be cast as a choice among a set of lexical items. Traditional grammar books (eg, (Jacobsen and Jørgensen, 1991)) contain extensive lists of such topics: we have identified at least a dozen such grammar problems in Danish that would be amenable to the technique described here. We are also planning to apply the approach to problems in English grammar.

We suspect that many grammar problems are similar to the article-noun agreement problem, and can be successfully addressed using The Brill tagger learning system. We are exploring ways in which the Brill tagger might be modified to make it more effective as a tool for building grammar checkers. In particular, the Brill tagger learns rules in a greedy fashion that always maximizes the success rate of the system overall, while we suspect that precision is relevant to the evaluation of rules.

## References

- Henning Bergenholtz. 1988. Et korpus med dansk almensprog. *Hermes*.
- Eric Brill. 1994. A report of recent progress in transformation-based error-driven learning. In *DARPA Workshop*.
- Andrew R. Golding and Yves Schabes. 1996. Combining trigram-based and feature-based methods for context-sensitive spelling correction. In *Proceedings of the 34th Annual meeting of the Association for Computational Linguistics*.
- Henrik Glaberg Jacobsen and Peter Stray Jørgensen. 1991. *Politikens Håndbog i Nudansk*. Politikens Forlag.
- Parole. 1998. [http://coco.ihl.ku.dk/~parole/par\\_eng.htm](http://coco.ihl.ku.dk/~parole/par_eng.htm).