Chunking: an unsupervised method to find errors in text

Jonas Sjöbergh
KTH Nada
Stockholm, Sweden
jsh@nada.kth.se

Abstract

We describe a method to use a chunker for grammar checking. Once a chunker is available the method is fully unsupervised, only unannotated text is required for training. The method is very simple, compare the output of the chunker on new texts to the output on known correct text. Rare chunk sequences that occur in the new texts are reported as suspected errors. By automatically modifying the chunk set to be more detailed for common verbs or prepositions more error types can be detected. The method is evaluated on Swedish texts from a few different genres. Our method can be used without modifications on any language, as long as a chunker is available for that language.

1 Introduction

In this paper we use chunking to mean dividing sentences into non-overlapping phrases. The sentence “The red car is parked on the sidewalk.” could be chunked as “[NP The red car] [VC is parked] [PP on the sidewalk]” where NP means noun phrase, VC means verb chain and PP means preposition phrase. Chunking is a relatively well developed field of research. There are chunkers available for several languages that give good results, usually well over 90% accuracy.

We present a method to use a chunker for grammar checking purposes. Once a chunker is available no manual work is required by this method, only a reference corpus of unannotated text is needed.

Automatic grammar checking is traditionally done by manually writing rules describing different error types and then applying these rules to new texts. There are also methods based on statistics or machine learning. Our approach is similar to the method used by Bigert and Knutsson (2002). In essence, they use trigrams of part-of-speech (PoS) tags, comparing them to trigram statistics from a reference corpus. Any sufficiently unlikely trigram is flagged as a suspected error. A few refinements are used to deal with data sparseness, which otherwise lead to quite a few false alarms.

Our method is very similar, comparing chunk n-grams to statistics from a reference corpus. Using chunks gives longer scope, since each chunk can be quite long. It also detects other types of errors, since some error types do not affect the PoS level as much as the chunk level and vice versa. Since the number of chunk types is much lower than the number of PoS tags the data is less sparse. This means that there is less need to add modifications dealing with data sparseness. We instead modify our method in the opposite direction, by enlarging the chunk set automatically for common verbs and prepositions. This detects new error types.

2 Description of the method

Our proposed method is to run the chunker on a reference corpus with known correct text. This training step collects and saves a lot of statistics on what chunk sequences occur in normal language use. When a new text needs to be checked, simply run the chunker on the new text. If it contains chunk sequences that never occurred in the reference texts, these passages are marked as possible errors. It is also possible to use a higher occurrence threshold for accepting chunk the n-grams as correct, for instance if we suspect there to be a few errors in the reference texts.

S. Werner (ed.), *Proceedings of the 15th NODALIDA conference, Joensuu 2005*, Ling@JoY 1, 2006, pp. 180–185
ISBN 952-458-771-8, ISSN 1796-1114. http://ling.joensuu.fi/lingjoy/

The number of different chunk types is generally quite small. Since the reference corpus is automatically annotated, and the only requirement is that it should contain high quality texts, it can be very large. These two facts mean that we can get very good statistics even for quite long chunk sequences. This means that there are few false alarms, since even rare chunk sequences will normally be present in the reference corpus. In our evaluations even 5-grams of chunks give very few false alarms. There were less than 10 false alarms in 10 000 words, with slight variations between different genres.

While it is possible to use the method on long chunk sequences, it is probably better to use shorter sequences. Since there is no detailed error diagnosis available it is not that helpful to get an error report saying there is something wrong with a very long text sequence. Pointing out shorter sequences makes locating the error easier.

While the statistics collected are reliable and thus give few false alarms, there are also quite few correct detections. One example from our tests was 13 correct detections (and 1 false alarm) on data where other grammar checkers detected between 60 and 100 errors (but with more false alarms).

To detect more errors we can modify the chunk set, by automatically modifying the output of the chunker. If we are interested in detecting errors related to verb use we can substitute the chunk used for verb chains with the actual words of the chunk. So for the example “[NP The red car] [VC is parked] [PP on the sidewalk]” we originally get the trigram “NP-VC-PP”. With the new modification we get “NP-is parked-PP”. This allows detection of new error types, for instance wrong verb tense, as in "I want to went home" or "I thought he is nice". Similarly, if we want to find errors related to prepositional use we can do the same for preposition phrases. This detects errors such as "I arrived on Stockholm".

While this detects many new errors there are also negative effects. The number of chunk types is no longer small, but actually very large. This means that the statistics from the reference corpus is very sparse, and thus there are a lot of false alarms.

A better approach is to change only those verb chains or preposition phrases that are common, i.e. occur more than some limit number of times in the reference corpus. If the limit is high enough this works quite well. We still have reliable statistics (few false alarms, though higher than originally), but of course many correct detections are also removed.

For noun phrases our chunker produces additional information: is the noun phrase in definite or indefinite form, singular or plural etc. This information can be included in a similar way. Adding this to the chunk set enables us to detect errors such as "these are my the cars", but of course the statistics become more sparse again.

A nice property of our method is that it is simple to tune the number of alarms you get. If you want more alarms (usually both correct detections and false alarms), simply use longer chunk sequences or a lower limit for exchanging verb chains etc.

3 Evaluation

We evaluated our method on Swedish texts. As reference texts we used the Swedish Parole corpus (Gellerstam et al., 2000) (about 16 million chunks) and the KTH News Corpus (Hassel, 2001) (about 10 million chunks). We tested three different genres: newspaper texts, student essays and essays written by non-native speakers learning Swedish. All error reports were manually checked to see if it was a genuine error or a false alarm. The texts were not searched for unreported errors. Compared to other grammar checkers our method is outperformed by state of the art grammar checkers (using manually produced rules for error detection) but similar in performance to other statistical methods.

In the tests in this section we allowed chunk n-grams to span sentence boundaries, we changed the tag for common verb chains and preposition phrases to their corresponding word sequences and we used the extra information available for noun phrases. For chunking we used the output of GTA (Knutsson et al., 2003), which outputs the following chunk types:

- adverb phrase
- adjective phrase
- boundary (clause or sentence)
- infinitive phrase

- noun phrase
- preposition phrase
- verb chain
- outside of phrase (for instance punctuation or interjections)

Noun phrases can also be replaced with a tag indicating the type of noun phrase found, such as a genitive form or a definite and plural form.

Other grammar checkers have also been evaluated on these texts, and some results from those are also presented, in Tables 4, 5 and 6, to give an idea of how good our method is in comparison. The two best grammar checkers evaluated on these texts were the grammar checker in the Swedish version of MS Word 2000 (Arppe, 2000; Birn, 2000) and Granska (Domeij et al., 2000), which are both based mainly on manually constructed rules for different error types. These results were taken from (Sjöbergh and Knutsson, 2005) and reproduced here for convenience.

In Tables 1, 2 and 3 the results using different n-gram lengths and different limits for when to consider a chunk "common" are presented for three different genres.

The test genres was newspaper texts, essays by native speaker students and essays by second language learners. The newspaper texts were taken from the SUC corpus (Ejerhed et al., 1992), which contain almost no errors. The writers also have a very good grasp of the language and use many "advanced" language constructions. The student essays were taken from the written part of the Talbanken corpus (Einarsson, 1976). The essays are argumentative, discussing the views expressed in some other texts the writers have read, and they quote a lot from these texts. The second language learner essays were taken from the SSM corpus (Hammarberg, 1977). The language proficiency varies from writer to writer, some have studied Swedish for only a few months while some have studied several years. These essays are usually quite short.

Looking in the tables, it can be seen that the method is easy to tune to produce few or many error reports. The optimal choice is probably different for different users. Writers with a good grasp of the language using many varied constructions would likely use a very high limit for "common" phrases, while users with limited knowledge of the language (and thus less productive use of their vocabulary) would benefit more from a lower limit. An experienced writer might also benefit more from reports of errors on long chunk sequences, probably being able to assess what is wrong and also capturing error types with longer dependencies. An inexperienced language user would probably be better served with a more precise diagnosis, i.e. use shorter n-grams, to understand what is wrong.

On newspaper texts there were almost no errors to detect. A reasonable performance level to choose on this genre for our method is perhaps 3-grams and the highest limit for "common" chunks. This gives one correct detection and two false alarms. No other grammar checkers were evaluated on this text, but on similar newspaper texts, also 10 000 words (which were not used for our method, since they turned out to be part of the reference corpus in our experiments), gave two detected errors and three false alarms for MS Word 2000, which was the best performing grammar checker on this genre. This is while not counting simple spelling errors, otherwise the best grammar checker was Granska which had 8 correct detections and 35 false alarms (no grammar checker had less than 35 false alarms when counting spelling errors).

The best performing grammar checker on essays written by non-native speakers was Granska, which detected 411 errors and made 13 false alarms, which is a lot more than our method detects, see Table 2. However, most of the 411 detected errors are simple spelling errors, which our method will generally ignore (since the chunker ignores them). If only grammatical errors and hard spelling errors (resulting in another existing word) are counted, the best performing grammar checker detects 118 errors, making 8 false alarms. Using 4-grams of chunks and a limit of 5 000 occurrences for "common" chunks, our method performs similarly, with 98 correct detections and 12 false alarms. MS Word 2000, which is tuned for high precision, detected 58 errors with only 3 false alarms on this text, not counting spelling errors.

There are very many errors in these essays, most of which were not detected by any of the grammar checkers. Since there are so many errors and since even trigrams of chunks can span quite a lot of text, finding n-grams of chunks with errors in them might be thought to be too easy. While it is true that there are many errors in the texts, just pointing out random chunk n-grams does not perform that well. When checking 50 random chunk trigrams, 27 would be counted as correctly detected errors and 23 as false alarms. Our method performs better than this.

On the student essays MS Word 2000 detected 14 errors with 3 false alarms and Granska detected 31 with 13 false alarms. When including spelling errors MS Word detects 38 errors with 31 false alarms and Granska 48 errors and still 13 false alarms. A reasonable performance level for our method here is 24 correct detections with 13 false alarms, which is not so good.

Our method performs quite poorly on these essays. This is mainly caused by them differing a lot from the reference domain, while still using correct language constructions. The main problem is that there are a lot of “short quotes” which is rare in the reference texts and thus give rise to many unseen chunk n-grams. There are also longer quotes from “odd” genres, such as old bible translations and law books, which while not erroneous are not examples of the more common use of Swedish, and thus lead to more false alarms.

4 Discussion

Many of the unseen n-grams are caused by chunker errors, i.e. the unseen chunk n-gram is not the n-gram we should find. Usually the reason the chunker makes an error is because there is an error in the text, which means that this is not a problem. The method correctly signals an error, though the reason might not be the one we expect.

As mentioned regarding the poor performance on the native speaker student essays, our method has trouble with texts from domains that differ too much from the domains the reference texts come from. In general this is not a great concerns, since it is cheap to add more data to the reference texts. All we need is raw text, although with relatively few errors.

Increasing not only the number of domains covered by the reference texts, but also just increasing the size of the reference data is gen-

N	Limit	Correct	False
3	500	4	63
3	5000	2	7
3	50000	1	2
4	500	14	179
4	5000	6	52
4	50000	3	19
5	500	26	279
5	5000	17	144
5	50000	15	74

Table 1: Evaluation results on 10 000 words of newspaper texts, taken from the SUC corpus. There are very few errors in these texts, which leads to poor accuracy.

N	Limit	Correct	False
3	500	75	20
3	5000	24	2
3	50000	5	1
4	500	223	43
4	5000	98	12
4	50000	33	6
5	500	315	60
5	5000	199	27
5	50000	108	13

Table 2: Evaluation results on 10 000 words of second language learner essays from the SSM corpus. With many errors to detect, it is easy to get quite high precision. Most errors in the text go undetected, though.

N-gram length	Limit	Correct	False
3	500	26	47
3	5000	12	11
3	50000	6	1
4	500	93	138
4	5000	42	47
4	50000	24	13
5	500	174	233
5	5000	118	138
5	50000	68	69

Table 3: Evaluation results on 10 000 words of native speaker student essays from the written part of the Talbanken corpus. Frequent use of quotations leads to many false alarms.

	MS Word	Granska
All detected errors	10	8
All false positives	92	35
Detected spelling errors	8	6
False positives	89	20
Detected grammar errors	2	2
False positives	3	15

Table 4: Evaluation of two state of the art grammar checking methods on proofread newspaper texts, 10 000 words. Table 1 shows our method on similar data.

	MS Word	Granska
All detected errors	392	411
All false positives	21	13
Detected spelling errors	334	293
False positives	18	5
Detected grammar errors	58	118
False positives	3	8

Table 5: Evaluation of two state of the art grammar checking methods on second language learner essays, 10 000 words. Table 2 shows our method on the same data.

	MS Word	Granska
All detected errors	38	48
All false positives	31	13
Detected spelling errors	24	17
False positives	28	0
Detected grammar errors	14	31
False positives	3	13

Table 6: Evaluation of two state of the art grammar checking methods on essays written by native speakers, 10 000 words. Table 3 shows our method on the same data.

erally a good idea. Since it is so cheap it is a good way to reduce the number of false alarms. False alarms are generally caused by rare language constructions being used, which is mitigated by a larger reference corpus. A larger reference corpus also gives a richer chunk set for a fixed limit on occurrences for “common” chunks, which can also lead to more correct error detections.

Our method detects many different types of errors. Some error types detected by our method are considered “hard” to detect by manually writing rules to detect them, and are thus not very well covered by traditional grammar checkers. This is one reason our method detected errors that no other evaluated grammar checker found.

The main error types detected by our method in these experiments was missing or erroneously placed commas, word order errors, spelling errors resulting in other existing words and using the wrong preposition for a certain verb. Other error types with fewer detections were verb tense errors, split compounds, missing words, missing sentence breaks, agreement errors, other inflectional errors, repeated words, unconventional use of fixed expressions and idioms and simply using a different word from the one intended (only from the learners).

Since our method detects many different types of errors but does not give a detailed diagnosis it is perhaps hard for a user to understand what is wrong. One way to mitigate this is to create different versions of the grammar checker. One version could for instance use the changing of chunk tags for verb chains and thus detect mostly verb related errors, while another might change the chunk tags for noun phrases and thus find noun related errors. They will likely all detect some error types related to chunk sequences in general, but those that only one version detects could be given a slightly more detailed diagnosis.

5 Conclusions

While our method did not perform as well as the best available grammar checkers it did produce useful results. The main strength of the method is that it is very cheap to use. All you need is a chunker and unannotated text.

Another strength of the method is that it is very easy to tune how many detections you want from the method. Usually both the number of correct detections and the number of false alarms are affected, so if you want to detect many errors and are prepared to accept more false alarms or if you want almost no false alarms at the cost of leaving some errors undetected, the same method can still be used.

Another strong point is that it detects errors that no other method detects and which are often hard to try to tackle with other methods. Together with the possibility to tune the method for very few false alarms means that our method can successfully be used in combination with other grammar checkers. With almost no new false alarms even quite few new detections can be useful, especially since it is so cheap to create a grammar checker using our method.

Acknowledgments

We thank Viggo Kann and Ola Knutsson for contributing useful ideas and helpful suggestions.

This work has been funded by The Swedish Agency for Innovation Systems (VINNOVA).

References

Antti Arppe. 2000. Developing a grammar checker for Swedish. In T. Nordgård, editor, *Proceedings of Nodalida '99*, pages 13–27. Trondheim, Norway.

Johnny Bigert and Ola Knutsson. 2002. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Proceedings of Romand 2002, Robust Methods in Analysis of Natural language Data*, pages 10–19.

Juhani Birn. 2000. Detecting grammar errors with lingsoft's Swedish grammar checker. In T. Nordgård, editor, *Proceedings of Nodalida '99*, pages 28–40. Trondheim, Norway.

Richard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. 2000. Granska – an efficient hybrid system for Swedish grammar checking. In *Proceedings of Nodalida '99*, pages 49–56, Trondheim, Norway.

Jan Einarsson. 1976. Talbankens skriftspråkskonkordans. Lund University.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå Corpus project. Technical report, Department of General Linguistics, University of Umeå (DGL-UUM-R-33), Umeå, Sweden.

Martin Gellerstam, Yvonne Cederholm, and Torgny Rasmark. 2000. The bank of Swedish. In *Proceedings of LREC 2000*, pages 329–333, Athens, Greece.

Björn Hammarberg. 1977. Svenskan i ljuset av invandrares språkfel. *Nysvenska studier*, 57:60–73.

Martin Hassel. 2001. Internet as corpus - automatic construction of a Swedish news corpus. In *Proceedings of Nodalida 2001*, Uppsala, Sweden.

Ola Knutsson, Johnny Bigert, and Viggo Kann. 2003. A robust shallow parser for Swedish. In *Proceedings of Nodalida 2003*, Reykjavik, Iceland.

Jonas Sjöbergh and Ola Knutsson. 2005. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In *Proceedings of RANLP 2005*, pages 506–512, Borovets, Bulgaria.