N-gram based Statistical Grammar Checker for Bangla and English

Md. Jahangir Alam, Naushad UzZaman and Mumit Khan
Center for Research on Bangla Language Processing, BRAC University, Dhaka, Bangladesh
jahangir_bu@yahoo.com, naushad@bracu.ac.bd, mumit@bracu.ac.bd

Abstract

This paper describes a statistical grammar checker, which considers the n-gram based analysis of words and POS tags to decide whether the sentence is grammatically correct or not. We employed this technique for both Bangla and English and also described limitation in our approach with possible solutions.

1. Introduction

Grammar checker determines the syntactical correctness of a sentence. Grammar checking is mostly used in word processors and compilers. Grammar checking for application like compiler is easier to implement because the vocabulary is finite for programming languages but for a natural language it is challenging because of infinite vocabulary.

Three methods are widely used for grammar checking in a language; syntax-based checking, statistics-based checking and rule-based checking. In syntax based grammar checking [1], each sentence is completely parsed to check the grammatical correctness of it. The text is considered incorrect if the syntactic parsing fails. In statistics-based approach [2], POS tag sequences are built from an annotated corpus, and the frequency, and thus the probability, of these sequences are noted. The text is considered incorrect if the POS-tagged text contains POS sequences with frequencies lower than some threshold. The statistics based approach essentially learns the rules from the tagged training corpus. In rule-based approach [3], the approach is very similar to the statistics based one, except that the rules must be handcrafted.

However, one of the most widely used grammar checkers for English, Microsoft Office Suite grammar checker, is also not above controversy [4]. It demonstrates that work on grammar checker in real time is not very easy task; so starting the implementation for language like Bangla grammar checker is a major feat.

Bangla being spoken by more than 200 million peoples [5], no significant work is done on grammar checking of Bangla text. This paper describes an ongoing statistical grammar checker based on n-gram analysis of words and Part-Of-Speech (POS) tags. We showed the performance of this grammar checker for English and Bangla.

2. Why statistical approach?

A statistical approach does not need language resources like handcrafted grammatical rules, except for perhaps a tagged corpus to train the language model (LM). Given the scarcity of language resources for Bangla, statistical approach may be the only reasonable one for the foreseeable future.

3. Methodology

In statistical approach we can simply measure the probability of a sentence using n-gram analysis. For example using bigram probability of the sentence “He is playing.” is,

P (“He is playing.”) = P (He | <start>) * P (is | He) * P (playing | is) * P (. | playing)

Now if any of these three words are not in the training corpus (used to train the LM) then the probability of the sentence will become zero because of multiplication. So if we consider the words in this statistical method then we need a huge corpus that must contain all the words of the language.

To solve this problem, we can use part-of-speech (POS) tags rather than individual words. Difference is, earlier we checked which words are more probable to come after any word and now we will be checking which POS tags are more probable to come after any tags. When we use the tags then the words are variable.

Take the previous example sentence “He is playing.” again. After tagging, the sentence becomes “He/pps is/bez playing/vbg ./.”. Now we can use the tag sequence to calculate the probability of the sentence.

P (pps bez vbg .) = P (pps | <start>) *P(bez | pps) * P(vbg | bez) * P(. | vbg)

The grammar checker we are proposing works as follows:

1. Assign tag for each word of a sentence.
2. Use n-gram (in our case, n=3; i.e. trigram) analysis (LM) to determine the probability of the tag sequence.
3. If the probability is above some threshold then the sentence is considered grammatically correct. In our model if probability is greater then zero then it considers the sentence as correct. Probability of a sequence becomes zero when two or more consecutive tags cannot be fit together (or in other word they are incompatible). This model does not employ any smoothing techniques yet.

At first we need a POS tagger, which will automatically tag the words or we need to tag the words (of a sentence) manually. Then use a trigram model (which looks two previous tags) to determine the probability of the tag sequence and finally make the decision of grammatical correctness based on the probability of the tag sequence. For example, using the Brown [6] corpus and Brill's tagger [7], calculations for the sentence "He saw the book on the table." are,

He/pps saw/vbd the/at book/nn on/in the/at table/nn ./.
P (pps | None None) = 0.0635486169593
P (vbd | None pps) = 0.213047910296
P (at | pps vbd) = 0.166456494325
P (nn | vbd at) = 0.483086680761
P (in | at nn) = 0.362738953306
P (at | nn in) = 0.350597938144
P (nn | in at) = 0.44004695623
P (. | at nn) = 0.0847696646819
Probability of the tag sequence = 5.16478478489e-06

Result of our grammar checker is: This sentence is probabilistically correct.

Now if we try a sentence in our model with mismatch in agreement "He have the book I want.", calculations of our grammar checker will be,

He/pps have/hv the/at book/nn I/ppss want/vb ./.
P (pps | None None) = 0.0635486169593
P (hv | None pps) = 0.0
P (at | pps hv) = 0
P (nn | hv at) = 0.491712707182
P (ppss | at nn) = 0.00493575681605
P (vb | nn ppss) = 0.293785310734
P (. | ppss vb) = 0.0361445783133
Probability of the tag sequence = 0.0
Result of our grammar checker is: This sentence is either incorrect or impossible to detect.

4. Grammar Checker for Bangla

We employed the same calculations as English for Bangla grammar checker. In the calculations we need to assign POS tags for Bangla words. Research effort on POS tagger lacks for Bangla. To implement a rudiment POS tagger, stochastic tagger is always preferable, because creating POS tagging rules is an onerous task, on the other hand, stochastic taggers performs better with little efforts. In a stochastic POS tagger, for better generation of POS tags, we need a large tagged corpus, which at present is not available for Bangla.

For our POS tagging, we used the implementation of Brill's tagger [7], which is a transformation-based tagger that generates rules from the training corpus. So the performance of our tagger increases with the increase of the size of training corpus. The present tagger with training corpus of 5000 words from Bangladeshi newspaper Prothom-Alo [8], gives an accuracy of 50%+.

For our grammar checker, we trained the Language Model (trigram) in the same 5000 words Prothom-Alo corpus.

If we try a Bangla sentence "মার্কিন নাগরিকদের প্রতি হুমকির খবর পাওয়া গেছে ।" for grammar checking, calculations of our grammar checker will be,

মার্কিন/ADJ নাগরিকদের/NC প্রতি/POSTP হুমকির/NC খবর/NC পাওয়া/NV গেছে/VF ।/PUNSF
P (ADJ | None None)=0.0523560209424
P (NC | None ADJ) = 0.8
P (POSTP | ADJ NC)=0.0613496932515
P (NC | NC POSTP) = 0.36
P (NC | POSTP NC) = 0.314285714286
P (NV | NC NC) = 0.0807453416149
P (VF | NC NV) = 0.0851063829787
P (PUNSF | NV VF) = 0.363636363636
Probability of the tag sequence = 7.26512469566e-07

Result of our grammar checker is: This sentence is probabilistically correct.

If we reorder some words of the above sentence as follows: "মার্কিন নাগরিকদের হুমকির প্রতি খবর গেছে পাওয়া ।"
Then the calculations of our grammar checker will be,

মার্কিন/ADJ নাগরিকদের/NC হুমকির/NC প্রতি/POSTP খবর/NC গেছে/VF পাওয়া/NV ।/PUNSF
P (ADJ | None None) = 0.0523560209424
P (NC | None ADJ) = 0.8
P (NC | ADJ NC) = 0.349693251534
P (POSTP | NC NC) = 0.0496894409938
P (NC | NC POSTP) = 0.36
P (VF | POSTP NC) = 0.114285714286
P (NV | NC VF) = 0.0
P (PUNSF | VF NV) = 0
Probability of the tag sequence = 0.0

Result of our grammar checker is: This sentence is either incorrect or impossible to detect.

Take another example sentence "বাংলাদেশের কৃষি আলোচনায় অগ্রগতি নেই ।" in our Bangla grammar checker, calculations will be,

বাংলাদেশের/NP কৃষি/NC আলোচনায়/NC অগ্রগতি/NC নেই/PRTN ।/PUNSF

P (NP | None None) = 0.157068062827
P (NC | None NP) = 0.233333333333
P (NC | NP NC) = 0.37037037037
P (NC | NC NC) = 0.260869565217
P (PRTN | NC NC) = 0.00621118012422
P (PUNSF | NC PRTN) = 0.25
Probability of the tag sequence= 5.4984269000e-06
Result of our grammar checker is: This sentence is probabilistically correct.

5. Performance

We have tested our grammar checker for both English and Bangla. Since the performance of grammar checker significantly depends on POS tagging output, we checked the performance of grammar checker by manual tagged sentences and also using automated taggers.

For English, using manual tagging the grammar checker's performance is 63% (detected 545 sentences as correct, out of 866 correct sentences). Using manual tagging for 378 correct sentences in Bangla, we have found that the grammar checker's performance is 53.7%. That is the grammar checker detected 203 sentences out of 378 sentences as correct.

For Bangla, we have tested 34 correct sentences, which were tagged by automated Bangla POS tagger to analyze the performance of the grammar checker. From the analysis we have found that the grammar checker produces about 38% correct result.

6. Discussion on performance

There are few reasons behind the low performance of our grammar checker. These reasons are described below.

Training data that is used to train the language model should have wide coverage of common grammatical and syntactical rules.

We have seen that current model works well for simple sentences but doesn't work the same way for compound sentences. Low performance for English test set is due to the large compound sentences in the Brown corpus.

Significant amount of performance of grammar checking depends on the result of POS tagging. We have seen this difference between manual tagging and automated tagging for Bangla.

We need a tag set for POS tagging. Since most of the grammatical mistakes are due to agreement (number, person etc.) mismatch, so we need a tag set with agreement features. The tag set we are using for Bangla do not have enough agreement features. As a result the grammar checker considers some of the wrong sentences with agreement mismatch as correct. For example, the Brown Corpus tag the word 'I' and 'You' with same tag. As a result a conflict arose as for the following cases,

Sentence 1: "**I are** playing"
I/ppss are/ber playing/vbg
P (ppss | None None) = 0.039321111615
P (ber | None ppss) = 0.0560131795717
P (vbg | ppss ber) = 0.236514522822
Probability of the tag sequence= 0.000520923351424
*Result of our grammar checker is: This sentence is *probabilistically correct!*

Sentence 2: "**You am** playing"
You/ppss am/bem playing/vbg
P (ppss | None None) = 0.039321111615
P (bem | None ppss) = 0.0280065897858
P (vbg | ppss bem) = 0.153846153846
Probability of the tag sequence= 0.000169423114296
*Result of our grammar checker is: This sentence is *probabilistically correct!*

Again, if we interchange two adjacent words with same tag then our grammar checker cannot detect the incorrect sentences.

For example, calculations for "বাংলাদেশের আলোচনায় কৃষি অগ্রগতি নেই ।" will be,

বাংলাদেশের/NP আলোচনায়/NC কৃষি/NC অগ্রগতি/NC নেই/PRTN ।/PUNSF
P (NP | None None) = 0.157068062827
P (NC | None NP) = 0.233333333333
P (NC | NP NC) = 0.37037037037
P (NC | NC NC) = 0.260869565217
P (PRTN | NC NC) = 0.00621118012422
P (PUNSF | NC PRTN) = 0.25
Probability of the tag sequence= 5.4984269000e-06
* Result of our grammar checker is: This sentence is *probabilistically correct!*

We have seen that interchanging two words produced wrong result. To resolve this problem, word level n-gram can be used. Using word level n-gram we can determine which word is more likely after given word(s).

Performance of our grammar checker also depends on which Language Model is used. Because

bigram consider coherence between two words (here between two tags), trigram consider among three, quadrigram four and so on. So which gram to use for a language depends on the average length of the sentences in the language.

7. Future work

Other than statistical grammar checker, rule based grammar checker can be introduced for Bangla. Final grammar checker can be a hybrid system combining both statistical and rule based approach.

8. Conclusion

Grammar checker is one of the most widely used applications in word processors, which itself is a very important tool for local language computation. We are proposing a statistical grammar checker for Bangla, which has a reasonably good performance as a rudiment grammar checker. We also discussed the limitation of our model with the suggestions to overcome these limitations.

9. Acknowledgment

This work has been supported in part by the PAN Localization Project (www.panl10n.net) grant from the International Development Research Center, Ottawa, Canada, administrated through Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan.

10. References

[1] K. Jensen, G.E. Heidorn, S.D. Richardson (Eds.), *Natural Language Processing, the PLNLP approach,* 1993.
[2] W.K. Chen, *Linear Networks and Systems*, Belmont, CA, Wadsworth, 1993, pp. 123–135.

[3] E. Atwell and S. Elliott, "Dealing with Ill-formed English Text", In: R. Garside, G. Leech and G. Sampson (eds), *The Computational Analysis of English: A Corpus-based Approach*, London, Longman, 1987.

[4] D. Naber, *A Rule-Based Style and Grammar Checker*, Diploma Thesis, Computer Science - Applied, University of Bielefeld, 2003.

[5] S. Krishnamurthy, *A Demonstration of the Futility of Using Microsoft Word's Spelling and Grammar Check*, available online at: http://faculty.washington.edu/sandeep/check/

[6] The Summer Institute for Linguistics (SIL) Ethnologue Survey, 1999.

[7] Brown Tagset, available online at: http://www.scs.leeds.ac.uk/amalgam/tagsets/brown.html

[8] E. Brill, "Some advances in rule based part of speech tagging", *In Proceedings of The Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Washington, 1994.

[9] Bangladeshi Newspaper, Prothom-Alo. Online version available online at: http://www.prothom-alo.net/

[10] Natural Language Toolkit, available online at http://nltk.sourceforge.net/index.html