Grammatical Error Checking and Polarity Determination of a Simple Sentence

Surajit Malik
Department of Computer Science and Engineering
University of Calcutta

Mrinal Mandal
Department of Computer Science and Engineering
University of Calcutta

Prof. S. K. Bandyopadhyay
Department of Computer Science and Engineering
University of Calcutta

Abstract

Grammar (or syntax) means that the sentence is syntactically correct. The main purpose of a grammar to check whether sentence is free of syntax errors. An entire document or one sentence at a time is checked for grammatical construct. Opinion mining in terms of sentiment analysis to identify the nature of sentence like positive or negative polarity of text. In modern days several social sites indicated text in the form of comment, Part of Speech (POS) etc. These comments are sense as positive text or negative text. This sense will help to find out the accurate result with the help of sentiment analysis. The main objective of the paper is to read a text and separates it into sentences. First phase of the paper is to detect grammatical error in the sentences based on POS. The detection is based on some assumptions. The second phase is to determine the polarity of the grammatically correct sentence.
Keywords- Polarity Analysis, Grammar, and Natural Language Processing

I. INTRODUCTION

An online grammar checker will identify errors as sentences are detected in the text. Grammar checkers can be computationally intensive and often run in the background or must be explicitly invoked. One of the primary requirements for a grammar checker is speed. A practical grammar checker must be fast enough for interactive use in an application like a word processor. The time to analyze a sentence should be sub-second or less following an initial startup time.

The second requirement is accuracy. A grammar checker should find all possible errors and correct sentences as well. There are two types of errors. The first type of error is a false positive or an error that is detected by the grammar checker but which is not an actual error. The second type of error is an actual error that was not detected by the grammar checker (a false negative). In general, the numbers of false positives are minimized to avoid annoying the user of the application.

The third requirement to limit the number of correct sentences that are flagged as errors by the grammar check, is related to the second requirement. At first, it is assumed that simply setting the threshold high enough for an error should be sufficient to satisfy this requirement. However, a grammar checker with a threshold that is too high will miss a large number of legitimate errors. Therefore, the threshold should be such that the numbers of false positives are minimized while simultaneously reducing the number of false negatives as well. The accuracy parameter defined in the Evaluation section combines these two attributes in a single value, making it possible to compare grammar checkers. Since it is difficult to set a universal threshold that is appropriate for all situations, the user can select a level of "strictness" for the grammar corrector. A higher level of strictness corresponds to more rigorous error checking.

Sentence detectors have fairly high precision rates (above 95%) for text that is well- written such as newswire articles, essays, or books. POS taggers also have high accuracy rates (above 90%), but have a dependency on the genre of text used to train the tagger.

Two methods to detect grammatical errors in a sentence have been popular. The first method is to generate a complete parse tree of a sentence to identify errors. A sentence is parsed into a tree like structure that identifies a part of speech for every word. The detector will generate parse trees from sentences that are syntactically correct. An error sentence will Grammatical Error Checking and Polarity Determination of a Simple Sentence either fail during a parse or be parsed into an error tree. One problem with this approach is that the parser must know the entire grammar of the language and be able to analyze all types of text written using the language. Another problem is that some sentences cannot be parsed into a single tree and there are natural ambiguities that cannot be resolved by a parser [1-4].

The second method is to use a rule-based checker that detects sequences of text that do not appear to be normal. Rule-based systems have been successfully used in other NLP problems such as POS tagging. Rule-based systems have some advantages over other methods to detect errors. An initial set of rules can be improved over time to cover a larger number of errors. Rules can be tweaked to find specific errors [5].

Part-of-speech tagging (POS tagging, or just tagging) is the task of assigning each word its POS tag. It is not strictly defined what POS tags exist, but the most common ones are noun, verb, determiner, adjective and adverb. Nouns can be further divided into singular and plural nouns, verbs can be divided into past tense verbs and present tense verbs and so on [6-8].

The more POS tags there are, the more difficult it becomes – especially for an algorithm – to find the right tag for a given occurrence of a word, since many words can have different POS tags, depending on their context. In English, many words can be used both as nouns and as verbs. For example house (a building) and house (to provide shelter). Only about 11.5% percent of all words are ambiguous with regard to their POS tags, but since these are the more often occurring words, 40% percent of the words in a text are usually ambiguous. POS tagging is thus a typical disambiguation problem: all possible tags of a word are known and the appropriate one for a given context needs to be chosen.

II. REVIEW WORKS

A grammar checker has two main functions:

- It notifies the user of possibly incorrect sentences (or fragments).
- It proposes corrections, possibly with a "linguistic" explanation of the error.

In practice, additional properties also are desirable:

- It should be fast: checking should be perform as the user types.
- It should minimize noise: too many false alarms would make it unusable.
- It should minimize silence: only few errors should remain unnoticed.

Some distinguishes 4 steps in the grammar correction process: (1) identification of possibly ungrammatical segments, (2) identification of the possibly infringed constraints, (3) identification of the possible source of the error, and (4) construction and Grammatical Error Checking and Polarity Determination of a Simple Sentence ordering of the correct alternatives. Before text is checked for grammatical errors, part-of-speech tagging and parsing need to be performed. The grammar checker used then starts by finding each sentence in the text, attempts to find any grammar errors in it and often suggests possible corrections. Most grammar checkers are rule-based, statistical or a hybrid of the two. Before grammar checking can be performed on a text it needs to be run through a part-of- speech (POS) tagger and parser. This enables the grammar checker to recognize types of phrases within each sentence, syntactic roles and features of each word [9].

The text is first run through a POS tagger which generates a tag for each word in a sentence. The tag indicates the word's class and morphological features (such as case, number, person and gender). Next, the text (with tags) is run through a parser which performs syntactic analysis on it, adding tags to parts of the sentence, marking phrases within it and syntactic roles.

In the statistical approach the system is trained on a corpus to learn what is 'correct'. As an example, this can be done using trigram frequency information. A POS-tagged corpus is then used to construct a list of tag sequences (each sequence being the tags of three consecutive words in the sentence). The system would then be trained to assume that a sentence is grammatically incorrect if it contains a tag trigram that was not seen in training (and possibly if it occurs very rarely in the corpus).

This method has a few disadvantages. One of these is that it can be difficult to understand the error given by the system as there is not a specific error message. This also makes it more difficult to realize when a false positive is given [2]. The largest disadvantage is that the largest hand-checked POS tagged corpus is relatively small (around 600,000 tokens). This could result in a lot of false positives in the error detection as there would be a rather high ratio of unseen tag trigrams. However, when one has all the resources needed, the statistical approach to grammar checking can be a good choice as it does not require as much manual work. The system could therefore possibly cover a larger error set than with the rule-based approach, with the same amount of work [10].

Using the rule-based approach to grammar checking involves manually constructing error detection rules for the language. These rules are then used to find errors in text that has already been analyzed, i.e. tagged with a part-of-speech tagger and parsed, so that the grammar checker recognizes types of phrases within each sentence, syntactic roles and features of each word, such as case, number, person and gender.

Word processor not only points out the errors it has detected but also suggests one or more corrections to the error. The user can then choose whether to ignore the error found (e.g. if it is a false positive or deliberate uncommon usage) or choose one of the correction suggestions from the grammar checker.

A good example of a rule-based grammar checking system is the Rule-based Style and Grammar Checker developed by [2]. The system checks text for certain grammatical errors and new rules can easily be added. For each rule in the system a description of the corresponding error and example sentences are provided for the user so as to make it easier to understand the problem and correct it.

III. PROPOSED METHOD

Initially it is required to use notation for each part of speech are these are as follows:

Noun → N	*Pronoun → P*	*Verb → V*	*Adjective → J*
Adverb → R	*Determiner → D*	*Preposition → I*	*Cardinal Digit → G*
Modals → L	*Coordinating Conjunction →C*		*Existential There → E*

Subject → M	*Predicate → O*		*Sentence → S*

Production rules for some sentences are placed below to demonstrate the basic idea of testing the validity of these sentences.

1) I am happy. || <P><V><J> || Production Rule: M ← P O ← VJ
2) Pritam is happy. || <N><V><J> || Production Rule: M ← N
3) He plays cricket. || <P><V><N> || Production Rule: O ← VN
4) They work hard. || <P><V><R> || Production Rule: O ← VR
5) The flower is very beautiful. || <D><N><V><R><J> || Production Rule: O ← VRJ
6) The boy is a student. || <D><N><V><D><N> || Production Rule: M ← DN and O ← VDN
7) The boy is a good person. || <D><N><V><D><J><N> || Production Rule: O ← VDJN
8) Wisdom comes with age. ||<N><V><I><N> || Production Rule: O ← VIN
9) Soldiers are pride of our nation. || <N><V><N><I><P><N> || Production Rule: O ← VNIPN
10) India became free in 1947. || <N><V><J><I><G> || Production Rule: O ← VJIG
11) You and I work together. || <P><C><P><V><R> || Production Rule: M ← PCP
12) The distance between India and Japan is 8000 kms. || <D><N><I><N><C><N><V><G><N> ||

Production Rule: M ← DNINCN O ← VGN

13) The cackling of geese saved Rome. || <D><N><I><J><V><N> || Production Rule: M ← DNIJ
14) I can read. || <P><L><V> || Production Rule: O ← LV
15) I cannot read. || <P><L><R><V> || Production Rule: O ← LRV
16) The early bird catches the worm. || <D><J><N><V><D><N> || Production Rule: M ← DJN
17) They arrived soon after. || <P><V><R><I> || Production Rule: O ← VRI

The final production rules in CFG are as follows:

M ← N | P
M ← DN | PD | JN | PN
M ← DJN | PCP | DNIG | DNIJ | GIDN | DNINCN | DINIDN
O ← V
O ← VJ | VN | VR | LV
O ← VRJ | VDN | VIN | VGN | LRV | RVR | VRI
O ← VDJN | VJIG | VNIPN
S ← MO

It is now required to minimize the above production rules. The rules are minimized and the final production rules are as follows:
M ← N | P | DM | MD | JM | MM | MCM | MIG | MIJ | MIM | DIM | GIM O ← V | OJ | OM | OR | LO | OI | RO | OO | OG | OGM | ODM S ← MO

IV. ALGORITHM

Input: A simple sentence
Output: Message whether the inputted sentence is grammatically correct or not. It also underlines the error.
Start
Step1: Tokenizing the sentence according to words
Step2: Determine the parts of speech (POS) of different word tokens
Step3: Indexing parts of speech and make a string pattern of POS of the words of the sentence.
Step4: Examine the string pattern whether there is any sub-string pattern that matches with right hand side of the stored production rule.
If matching found then
Replace the left hand side index with matching pattern in POS string pattern and continue from step 5.
Else
Go to step 6
Step 5: Check whether final string pattern matches with "S".
If matching match then
The sentence is grammatically correct else it is incorrect.
Stop

Others' opinions can be crucial when it's time to make a decision or choose among multiple options. When those choices involve valuable resources (for example, spending time and money to buy products or services) people often rely on their peers' past experiences. Until recently, the main sources of information were friends and specialized magazine or websites. Now, the "social web" provides new tools to efficiently create and share ideas with everyone connected to the World Wide Web. Forums, blogs, social networks, and content-sharing services help people share useful information.

Capturing public opinion about social events, political movements, company strategies, marketing campaigns, and product preferences is garnering increasing interest from the scientific community (for the exciting open challenges), and from the business world (for the remarkable marketing fallouts and for possible financial market prediction). The resulting emerging fields are

opinion mining and sentiment analysis. Although commonly used interchangeably to denote the same field of study, opinion mining and sentiment analysis actually focus on polarity detection and emotion recognition, respectively. The identification of sentiment is often exploited for detecting polarity. The next task is to find out the polarity of the sentence.
The algorithm is aimed at determining the polarity (i.e. positivity, negativity and mixed/neutrality) of a review by searching for certain words and phrases that particularly refer to positive and negative emotions in an individual.

The term polarity has a number of different uses, but in this dissertation it is used primarily to refer to the positive or negative sentiment being expressed by a word. However, there is an important distinction between the prior polarity of a word and its contextual polarity [4]. The prior polarity of a word refers to whether a word typically evokes something positive or something negative when taken out of context. For example, the word beautiful has a positive prior polarity, and the word horrid has a negative prior polarity. The contextual polarity of a word is the polarity of the expression in which the word appears, considering the context of the sentence and the discourse. Although words often do have the same prior and contextual polarity, many times the word's prior and contextual polarities differ. Words with a positive prior polarity may have a negative contextual polarity, or vice versa For example, in sentence 1 the word "denounced" has negative and approved has a positive polarity while in sentence 2 don't hate has positive and hate has negative polarity according to algorithm.

A different method for determining sentiment is the use of a scaling system whereby words commonly associated with having a negative, neutral or positive sentiment with them are given an associated number on a −4 to +4 scale (most negative up to most positive). This makes it possible to adjust the sentiment of a given term relative to its environment (usually on the level of the sentence). When a piece of unstructured text is analyzed using natural language processing, each concept in the specified environment is given a score based on the way sentiment words relate to the concept and its associated score. This allows movement to a more sophisticated understanding of sentiment, because it is now possible to adjust the sentiment value of a concept relative to modifications that may surround it. Words, for example, that intensify, relax or negate the sentiment expressed by the concept can affect its score. Alternatively, texts can be given a positive and negative sentiment strength score if the goal is to determine the sentiment in a text rather than the overall polarity and strength of the text. The algorithm is proposed to find out the polarity of sentence.

V. ALGORITHM

First of all we have to create a file (as dictionary.txt) which contains sentiment words. Here scaling system is used where sentiment words commonly associated with number or index on a -4 to +4 scale (most negative to most positive). Actually in this file semantic indexing is done to assign polarities to individual sentiment words.
Input: The grammatically correct sentence output from above algorithm
Output: Show the polarity of each sentiment word that contain in the inputted sentence and determine finally determine the polarity of the whole sentence.
Step 1: Input the sentence from and polarity Count ←0 and i← 0
Step 2: Tokenizing the sentence according to words and store them in a list namely wordToken.
Step 3: Take wordToken[i] and matches with sentiment word one by one stored in dictionary.txt until the wordToken list is empty.
If matching found then
polarityCount ← polarityCount + polarityIndex(actually polarityIndex of the relevant matching sentiment word)
Go to Step3
Else then
Go to step3
Step 4: Print matched word along with its polarity and polarity Count of the sentence.

VI. RESULTS

Enter the Sentence: You miss 100 percent of the shots you never take.
The process
The POS pattern: PVGNIDNPRV.
PVGNIDNPRV.
PVGMIDMPRV.
MVGMIDMMRV.
MVGMIMMRV.
MVGMIMRV.
MVGMRV.
MOGMRO.
MOMRO.
MORO.
MOO.
MO.

S.
The Sentence is Grammatically Correct.
Polarity of the following Text:
'You miss 100 percent of the shots you never take.'
It is: -2
Since its value is negative so the polarity of the text is negative.

VII. CONCLUSIONS

Two tasks are necessary in all grammar checkers which are sentence detection and part of speech (POS) tagging. Therefore, this dependency detects the accuracy of any grammar checker to the combined accuracy of the sentence detector and POS tagger. The first objective of the paper for checking syntax of the given text. The paper is also considered the relative clauses. Finally the polarity of sentences are also detected.

REFERENCES

[1] Sågvall Hein, A." A chart-based framework for grammar checking – initial studies", 11th Nordic Conference in Computational Linguistic, 1998.
[2] Dey, L., Haque, S.M. "Opinion mining from noisy text data", Int. J. Document Anal. Recognition 12, 205–226, 2009.
[3] Neviarouskaya, A., Prendinger, H., Ishizuka, M." Semantically distinct verb classes involved in sentiment analysis", Proceedings of the International Conference on Applied Computing (AC 2009), Japan, 2009.
[4] Puneet Singh, Ashutosh Kapoor, Vishal Kaushik, and Hima Bindu Maringanti. "Architecture for auto- mated tagging and clustering of song less according to mood", International Journal of Computer Science Issues, 2010.
[5] Jisha P. Jayan, Deepu S. Nair, Elizabeth Sherly "A subjective feature extraction for sentiment analysis in Malayalam language", IEEE 2015.
[6] Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi and Tao Li, "dual sentiment analysis: considering two sides of one reviews", IEEE 2015.
[7] Duyu Tang, Bing Qin, Furu Wei, Li Dong, Ting Liu and Ming Zhou , "A joint segmentation and classification frame work for sentence level sentiment classification " , IEEE 2015.
[8] Yejin Choi and Claire Cardie "Learning with compositional semantics as structural inference for sub sentential sentiment analysis", IEEE 2008.
[9] V.S.Jagtap and Karishma Pawar "Analysis of different approaches to sentence level sentiment classification", IEEE 2013.
[10] Xiaoqian Zhang, Shoushan Li and Hongxia Zhao, "Polarity shifting: Corpus construction and analysis", IEEE 2006.