A Rule-based Grammar and Spell Checking

Yash Gondaliya¹, Poojan Kalariya¹, Brijeshkumar Y. Panchal¹*, Amit Nayak²

¹Department of Computer Science and Engineering, Devang Patel Institute of Advance Technology and Research (DEPSTAR), Faculty of Technology and Engineering (FTE), Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India
²Department of Information Technology, Devang Patel Institute of Advance Technology and Research (DEPSTAR), Faculty of Technology and Engineering (FTE), Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat, India

ABSTRACT

Rule-Based Grammar and spell checking is the task of checking the content of any specific language using a particular set of rules for their grammar and a set of spellings for spell checking. In this work, libraries and functions of Python useful for Natural Language Processing are used to perform grammar and Spell checking. Python Libraries NLTK (for Sentence Boundary Detection and POS tagging), pyEnchant (for Spell checking by comparing with a standard dictionary), and Inflect (for indefinite article). The article includes information about these libraries, techniques, and steps used for Grammar and Spell Check.

Keywords: Grammar and Spell Checking, POS tagging, Sentence Boundary Detection, Spell Check.

INTRODUCTION

There are many conditions where the spellings of some words that researchers are frequently using in a speech are somewhat different from what researchers are expecting, and many have a habit of forgetting spellings frequently. Using incorrect spelling in any way gives the wrong impression of someone’s proficiency; hence, it is necessary to use the correct spelling and grammar while texting or proposing anything. Grammar checkers and spell checkers are frequently used nowadays.

There are many techniques used for grammar and spelling checking among them. A rule-based approach is the one in which the model is given the pre-defined set of rules compared with the input to detect the errors in the input, and other sets of corrections are given that are used to correct the errors. The whole task is performed into different parts where the given input text is divided into specific sentences. The POS taggers tag those sentences, and then spell check is applied to the words which do not match the given library. Suggestions are given to choose the correct spelling, then comes the Indefinite Article checking, which checks the specific exceptions where the article needs to be corrected.

Related Work

In 1992, Vosse had developed a rule-based grammar checking system for the Dutch language, which was targeted to detect a morpho-syntactic error. It is concerned with three different types of errors i.e. Typographical error (error due to Typing Mistake), Orthographical error and morpho-syntactic error. This system contains errors like homophonic words, which are the words that spell different but have the same pronunciation, homophonic words that differ only in inflection, repeated words, agreement errors, and idiomatic expressions errors which frequently occur in the Dutch language and is considered important as they are seen as insufficient language competence rather than any mistake. This system is formed of two main levels, which are “word level” and “sentence-level”. Before moving to the sentence level it is necessary to check the spelling module in a language like Dutch because compound nouns, verbs, and adjectives are written as a single word, they cannot always be looked up in a dictionary hence they need to be analyzed.¹¹

In 1998 a rule-based grammar checker was developed in the Swedish language by Hein. The non-structural and structural problems were recognized by using Local Error Rules. A parser and a chart scanner were the two main components of the system. Input text was fed to the parser, and the parser generated the chart, and that chart was further...
A Rule-Based Grammar and Spell Checking

Disadvantage

In terms of morpho-syntactic errors, the system in English text used the Python rules. 54 grammar rules, 81 pairs of a false friend, and four built-in checking models. This system used 5 rules defining style, the system as its POS tagset. POS tagged and phase chunked dependent on the training corpus. BNC C5 tagset was used by this feature helped to overcome the problem of inaccurate results of the probabilistic tagger, which were completely dependent on the training corpus. BNC CS tagset was used by the system as its POS tagset. POS tagged and phase chunked text was finally passed to the manually constructed grammar checking models. This system used 5 rules defining style, 54 grammar rules, 81 pairs of a false friend, and four built-in Python rules.

In 2005, Rider came up with a rule-based system for Dutch texts. Morpho-syntactic Errors in Detecting and Correcting

Method/Algorithm/
Techniques
Detecting and Correcting Morpho-syntactic Errors in Dutch texts
Advantage:


Advantage: There are only basic spell checkers that use simple word lookup. The result is completely different when this method is applied to the sample text using the same Dictionary as in the full system.

Disadvantage: The word “Yash” is a spelling mistake (Figures 1 and 2), but it’s the name of a

Comparative Analysis

Table 1: Comparative analysis of related work.

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>Year and Author</th>
<th>Method/Algorithm/Techniques</th>
<th>Accuracy</th>
<th>Disadvantage/Advantage/Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Theo Vosse 1992 [1]</td>
<td>Detecting and Correcting Morpho-syntactic Errors in Dutch texts</td>
<td>46.67</td>
<td>Disadvantage: There are only basic spell checkers that use simple word lookup. The result is completely different when this method is applied to the sample text using the same Dictionary as in the full system. Advantage: In terms of morpho-syntactic errors, the combination of a word-level spell checker and a syntactic parser performs from almost perfect to satisfactory depending on the complexity of the sentences.</td>
</tr>
</tbody>
</table>
Here the spelling of “good” is detected with error (Figure 3), and model had suggested the appropriate suggestions for that error and selecting the option “3,” which researchers need to use for our sentence.

### Steps Implemented

#### Sentence Boundary Detection

This task is just to divide the whole input text entered by the users into single sentences so that it can be further processed.
A Rule-Based Grammar and Spell Checking

```python
print("Corrected sentence: ", spellchecked)
print("---------------------------------")
print("UNCHECKING INDEFINITE ARTICLE AND FOLLOWING WORD AGREEMENT...
"
)+
tokens1 = nltk.word_tokenize(spellchecked)
a_v_a_an_checked = t.a_v_a_an(tokens1)
if a_v_a_an_checked is None:
    print("There is no indefinite article in the sentence."
 else:
    print("Corrected sentence: ", a_v_a_an_checked)
```
for grammar and spell-check. Researchers have done this task using the Natural Language Toolkits’ “sent tokenize,” which separates the whole text into different sentences.

**Part-of-Speech (POS) Tagging**

Part-of-Speech tagging is the task of giving each of the word of the sentence its part and role in the sentence, such as nouns, verbs, and many other components that make the complete sentence. By this, every component of the sentence is tagged and ready for further checking and other operations. researchers have used NLTK’s "pos_tag" for POS tagging in this model.
A Rule-Based Grammar and Spell Checking

Once each word in the sentence has given its part of speech tag, the error in each word can be detected according to their part of speech and their role in the sentence. Now researchers have used the penchant library of python to check the spelling of the words tagged, which are compared to the given Dictionary, and one can also add special words to their set so that the checker does not check those words and consider them as always correct.

Indefinite Article and Following Word Agreement

The articles are frequently used in sentences where it is necessary and important to use the proper article before the proper word. Hence, to do this task, researchers have used the "inflect" to consider some expectations while allocating specific articles before the specific word.

Demonstration Screenshots

Accuracy

Here, researchers have used the dataset of nearly 100 sentences to test the Spell Check ability of the model and ten sentences for the Indefinite Articles ("a vs. an"), including some sentences with exceptions. In most cases, the model was able to detect the error and give the appropriate suggestion for the error. In some cases, the model was able to detect the error correctly but failed to give the appropriate suggestions. In some cases, the model was totally unable to detect the error.

Accuracy = \frac{\text{No. of Incorrect sentences corrected by model}}{\text{Total No. of Incorrect sentence entered}}

The accuracy achieved by proposed mode in Spell Check as per the above equation is "0.94," i.e., 94%
The accuracy achieved by proposed model in Indefinite Articles ("a vs. an") as per the above equation is "1," i.e., 100%
Overall accuracy for the model is 0.94, which is 94%.

Previous System

As there are many different types of names in this world, any of the systems cannot tag each and every name from the world, so the previous system was showing some names as Spell Error and was giving suggestions related to the words in the Dictionary which can be considered as the failure of the system while checking the accuracy, so researchers introduced the feature where user can the original word into the sentence to overcome this problem. As the previous system had the module where one can enter the list of words that does not Spell check, it is difficult to collect the dataset which considers all the Names and some of the short forms. Also, hence researchers feature will be useful to the user and increase the model's accuracy.

Conclusion

A rule-based system always needs an accurate and complete set of rules so that the system can be made more accurate. As the rules are provided manually, this type of system is more predictable, and personalized results can be obtained through it. This type of system is always useful for
comparing and using the rules to make other systems more accurate.

Grammar and spellings check of the given texts flows in this way like, firstly the sentences are separated then every part of speech of sentence is tagged and then each part is checked in accordance to their tag and suggestions for detected errors can be given.

**Future Work**

In this model, researchers are able to detect and solve the errors of spelling mistakes and Indefinite Articles ("a" vs "an"), but the styling and syntax of the proper English sentence can be improved by using a proper set of grammar and styling rules for parsing of sentences.

**References**


