I. Error Analysis of our Grammar Checker

The values of Precision and Recall of our Grammar Checker on the given data are as follows:

Precision: 0.356
Recall: 0.4944

a) Reasons for Low Recall (High FNs)

Recall means of all the sentences that are incorrect (label:1), how many sentences could our parser detect to be incorrect (prediction:1)?

We are getting a recall of about 50% i.e. our parser can reject about half of the ungrammatical sentences in the dataset. **The reason for this average performance is the substantial number of False Negatives** i.e. sentences that were actually incorrect (label:1) but our parser considered them as correct (prediction:0).

1. One reason for a high number of False Negatives is the **lack of intelligence to distinguish between valid and invalid words with the same POS Tag.**

   For e.g. Consider the following sentence:

   “I ’d like to know USA.”

   The POS tags assigned to this sentence are PRP MD VB TO VB NNP .
   The truth class assigned to this sentence is 1 (incorrect grammar)

   You can observe that if the word ‘USA’ was replaced by ‘John’, this sentence would be perfectly alright. ‘John’ and ‘USA’ are both proper nouns (NNP) but if one proper noun is used the sentence will be right whereas if the other proper noun is used, the sentence becomes incorrect.

   Because our parser cannot distinguish between which proper noun is valid according to the context of the sentence, it cannot correctly classify such a sentence as 1(incorrect) and instead predicts the label as 0(correct).

   Another example where we see the same pattern is:

   “It was just an usual day .”

   You can observe that if the determiner ‘an’ was replaced by another determiner ‘a’, the sentence would be perfectly alright.

   Hence we get False Negatives in all such cases.

2. Another reason for the high number of False Negatives is the **lack of intelligence to understand whether the actual word that will replace a POS tag is misspelled.**

   For e.g. Consider the following sentence:

   "Take care of youself ."

   The POS tags assigned to this sentence are VB NN IN PRP .
   The truth class assigned to this sentence is 1 (incorrect grammar)

   You can observe that if the word 'youself' was correctly spelled as 'yourself', this sentence would be perfectly alright. **A human can note this easily but our grammar checker is not intelligent enough to detect spelling mistakes as it is based only on the POS tags and not on the actual words that replace these POS Tags.**

   More examples are :

   "I vish you were here with me ."
   "The entery will be free ."

   Hence we get False Negatives in all such cases.

3. Another reason for getting low recall is the Misclassified sentences in the dataset.

   We came across certain instances of data which happened to be misclassified as incorrect (label 1). Although a newer and more clean version of the dataset was released, we had finished a major portion of the assignment prior to its release.

   Example of sentences that are tagged incorrect (label: 1) but seem grammatically correct to us:
   "In conclusion , I had an unfortunate time ."

   Due to such sentences, we will get False Negatives as our grammar will predict them to be correct (prediction:0).

b) Reasons for Low Precision (High FPs)

Precision means of all the sentences that our parser detected to be incorrect (prediction:1), how many are really incorrect (truth_label:1)?

We are getting a precision of about 34%. **The reason for this below-average performance is the substantial number of False Positives** i.e. sentences that were actually correct (label:0) but our parser considered them as incorrect (prediction:1).

1. One reason for a high number of False Positives is that the **sentences missing a period as the last token are marked correct (label 0) in the dataset but our grammar won't accept these sentences (prediction:1) as we have made a rule that declarative sentences must always end with a period.**

   For e.g. Consider the following sentence:

   "Hope to hear from you soon"

   The POS tags assigned to this sentence are VB TO VB IN PRP RB
   The truth class assigned to this sentence is 0 (correct grammar)
   Because this sentence is missing a '.' at the last, our parser gives it a prediction of 1 (incorrect sentence).

   Similarly, other examples are "I would like to travel", "I bought a group ticket for the show", "It was a very good experience for me", "Shopping is not always enjoyable" etc.

   Hence we get False Positives in all such sentences.

2. Another reason for the high number of False Positives is that **some sentences in the data are ending with comma but still are labeled as correct (label: 0) in the dataset.** Our grammar will instead consider these sentences as ungrammatical (label:1) which seems logical according to the rules of the English language.

   For e.g. Consider the following sentences:

   "Thank you again and take care ,"
   "Technology improves our lives ,"

   Both of these sentences have been provided label 0 and our prediction for them is 1 (ungrammatical) and hence we get False Positives for such sentences.

3. More examples that cause False Positives:

   There are also sentences that end with a colon ':' or have the symbol in between the sentences. Our grammar does not identify these sentences.

   Examples of such sentences:
   "Here is the information you asked for :"
   "Also there is a big problem : the money ."

   Another example of a sentence that is tagged as correct (label:0) but should be incorrect:
   ". I am glad to provide all the information"

   Some examples have noisy POS tags (in the following example, the single quotes are tagged as two separate single quotes at the end of the sentence).

' That 's me ! '	`` DT VBZ PRP . ''

   Hence in all such cases, we will predict the label as 1(incorrect) but the dataset has given these sentences the label:0 (correct) and hence we will get False Positives!

II. With our current design, is it possible to build a perfect grammar checker? If so, what resources or improvements are needed? If not, briefly justify your answer.

While we can certainly improve the performance of our grammar checker by adding more rules that work for **specific sentences** in our training dataset, we will **not be able to reach a perfect grammar checker** with the current design. There are a few reasons for this.

1. There is no way of checking the spelling of words in our current setting. Since we are only considering the POS tags to base our parsing on, we cannot check whether the POS tag will be replaced by a correctly spelled word.

   For example:

   "A free shuttel bus gets you to the stadium."

   This sentence is perfectly valid according to the CFG rules we using because though "shuttel" is not a valid word in English, it is still tagged with the POS tag "NN" and hence this sentence will be parsed by our grammar checker.

2. We will keep having the problem of successfully parsing a sentence that is not valid semantically (as per rules of the English language) but is completely valid according to the POS tag-based CFG rules we are using in the current setting.

   For example:

   “I want to keep eat this sandwich.”

   This sentence is valid according to our grammar because a VP can decompose to Verb NP and then it can further derive the part of the sentence “eat this sandwich”. **We cannot in any way put a check that Verb should derive “eating” but not “eat” because “eat” will not make sense according to the sentence.** Both “eat” and “eating” are valid because both are valid POS tagged words that can derive from a Verb.

   Another example:

   “I would like to travel on July.”

   We cannot in any way put a check that the preposition tag should be replaced by “in” and not “on” in the above sentence.

Thus, we cannot build a perfect grammar checker that takes care of these aspects using the CFG based parsing we are using.

One alternative we can consider is using Context-Sensitive Grammars which are more powerful than CFGs. While replacement in a Context-free grammar happens without considering the left and right context the terminal will be placed at, on the other hand, Context-sensitive grammars derivations involve consideration of the context as well. The rules are of the type $\alpha A\beta \rightarrow \alpha\gamma\beta$ which means that α and β will decide whether A can be replaced by γ or not. This can somewhat improve our derivations for the English language which has valid and invalid sentences according to the context in which words are put together.

Reference:
https://www.wikiwand.com/en/Context-sensitive_grammar