COMP 6751 Natural Language Analysis

Project Report 3

Student name: Md Sakib Ullah Sourav

ID: 40264066

Expectations of originality:

I, Md Sakib Ullah Sourav (student id 40264066), certify that this submission is my original work and meets the Faculty's Expectations of Originality.

Date: November 15, 2023

Contents

Q1. Why did you make the grammar do what it does? Why did you allow it to not do what it doesn't?

As instructed in the project description, I enforced number agreement in the Noun phrases and number and person agreements between verb and subject. The lexicon has been built in light of these agreement features (Please refer to the grammar I submitted).

For this project, the grammar looks at the feelings expressed in adjectives, which then move up the tree through Noun Phrases, Verb Phrases, Adjective Phrases, and so on. Also the grammar uses three types of features: pos for positive, neg for negative, and neut for neutral sentiments. These features are assigned at the lexicon level. As the Feature Earley Parser finishes lexicons based on the rules, the sentiment feature moves up toward the top of the tree. The sentiment at the top of the tree is what the whole sentence is considered.

In the grammar, if a sentence has two phrase parts, it cannot distinguish the sentiment wisely. If the sentiments of the two phrases are different, the whole sentence is marked as neutral. There are also specific cases, like NP[pos] and PP[neg], where phrases with opposing sentiments are considered, but these are rare and handled individually.

After applying all these rules, the parser might create one or more trees for each sentence. Each tree can label the sentence as positive, negative, or neutral. To combine all the results, there's a voting module that counts the number of trees for each sentiment category, and the sentence's sentiment is based on the category with the most trees.

To keep the grammar less complicated, I did not apply the extensive rules of "And" and "But" so that the grammar could detect larger sentences well. However, the current version is pretty well performed under the training sentences that also validates its wider usability.

Q2. What does your grammar not do that you think important?

 When looking at sentiment analysis result in my grammar, many sentences marked as False were considered "Neutral" instead of "Positive" or "Negative". The problem often comes from longer, more confusing sentences that end up being seen as neutral due to errors in the analysis. This results in more "Neutral" sentences than "Positive" or "Negative" ones during voting.

To fix this, we can make the grammar rules more precise and add more features; for example,

i)   Introducing passive sentence forms,

ii)   Inclusion of Auxiliary verbs

iii)   Subcategorization of different verbs

iv)   Inversion (INV) of sentence structure

v)   Include Modals, negation and similar kinds in the grammar.

But all these might make the rules stricter and time consuming at the same time.

Another issue with the grammar is that it doesn't pay enough attention to commas. Even though commas carry a lot of meaning in a sentence, it's easier for programming to ignore them. However, keeping them in could add emphasis and change the positivity or negativity of a phrase, which is something the current system overlooks.

During the next project, I would try to implement the things I just mentioned above.

Q3. What kind of semantics you can do with your grammar (and how much additional work that would require)

In my views, making the grammar better can be done by adding more grammar features. This will make it easier to classify sentences correctly.

As I mentioned above, there might be more meanings in English that we can find by working more on this grammar.

Also, if we create a sentiment analyzer for a specific area, we can adjust the feelings assigned to words to match that area more accurately than we can in a general situation.