CS 421 – Natural Language Processing

Automated Essay Grading

Sourabh Parime and Kashyap P Desai

Part A

The number of sentences in the essays were counterd using 2 methods:

1. Using sent_tokenize() of nltk
2. Using was based on a research paper – 'Recognizing Sentence Boundaries and Boilerplate' [1].

The first method was straightforward and obvious. Each essay was passed to the sent_tokenize method of nltk and the sentences generated were stored in a list and then the count was taken.

The second method is represented roughly in the following flow-graph:

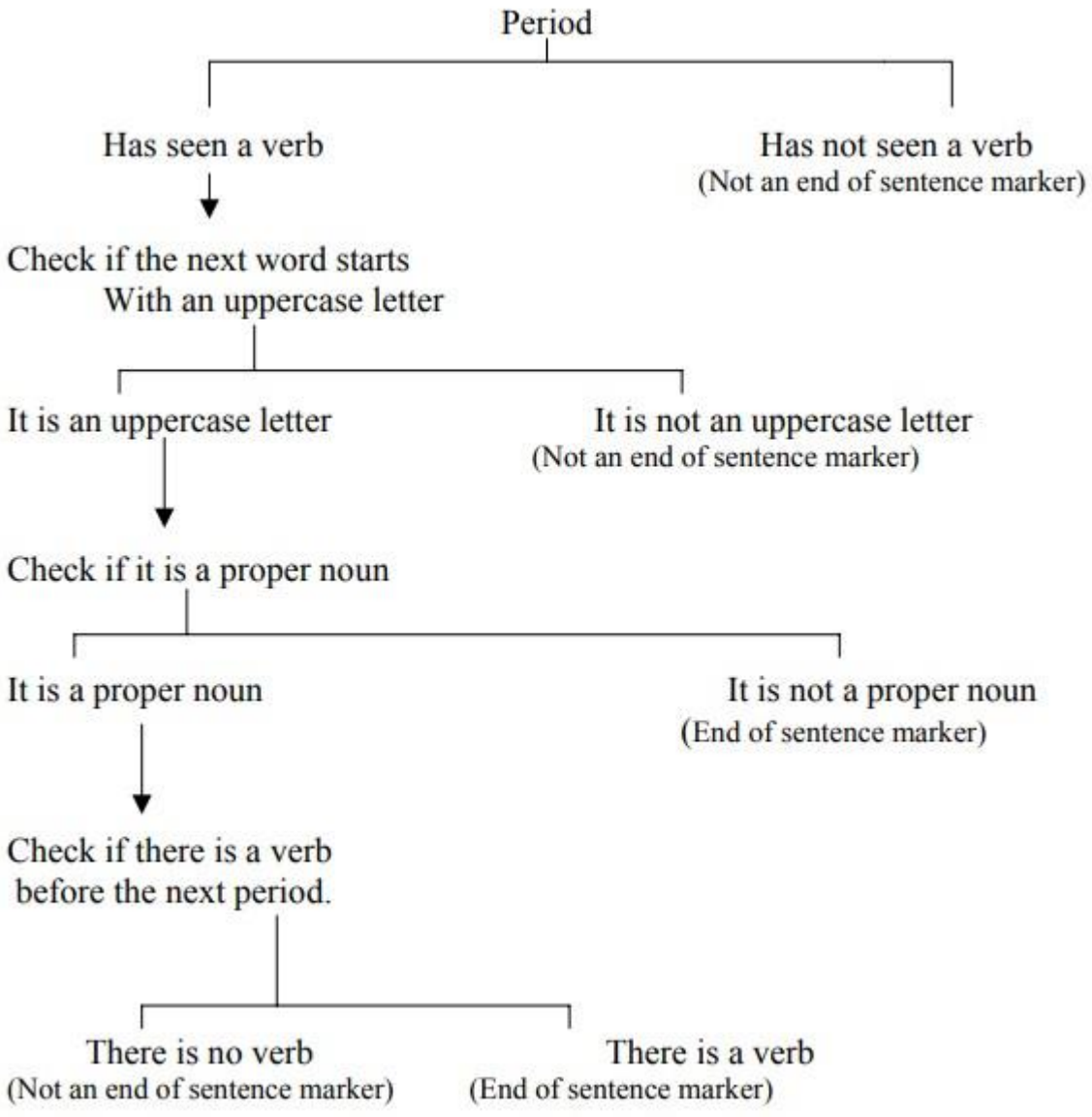

Figure from Aluthgedara, 'Recognizing Sentence Boundaries and Boilerplate'

The constraint of checking if an uppercase letter was removed since many essays in the corpus didn't follow capitalization after period. Also, it was not only the period being checked it was exclamation mark (!), question mark (?), semi-colon (;) and double question marks (??) too.
Sentences were counted based on these sentence boundaries recognizing criterion and were stored in a list.

The final count of number of sentences was calculated by taking an average of the counts obtain by the above 2 methods.

Part B:

Spelling mistakes were counted using the following method:

1. Each essay is tokenized into sentences and then into words.
2. If the word is not available in wordnet and is not a stop word the it is counted as a mistake.
3. For the next phase of the project an additional step to check in the English dictionaries would be used for example the words color and colour are both correct according to different dictionaries like US and UK

Part C (i)

Firstly, parse trees were built out of the sentences tokenized. Each parse tree was then traversed to check if the subject verb agreement was satisfied.

The tree was traversed to find a node which has the label 'S' and has 'NP' and 'VP' among its various children. Consequently, NP and VP were identified for that node following which the head noun, head verb, head pronoun were identified.

The subject-verb agreement was checked between these head nouns, pronouns and verbs in terms of number and person. To identify the agreement in person the following constraints were checked:

1. if head noun was "NNP" and the head verb was "VBP" or "VBN" then a penalty of 1 was added.
2. if head pronoun was a third-person pronoun (like 'he', 'she' or 'it') and head verb was "VBP" or "VBN" then a penalty of 1 was added.
3. if head pronoun was not a third-person pronoun and head verb was "VBZ" or "VBN" then a penalty of 1 was added.

To identify the agreement in number the following constraints were checked:

1. if head noun was a plural noun and head verb was a singular verb then a penalty score of 1 was added
2. if head pronoun was a plural pronoun and head verb was a singular verb then a penalty score of 1 was added
3. if the determiner was singular (like 'a', 'an') and head noun was plural then a penalty score of 1 was added
4. if the determiner was plural (like 'these', 'those') and head noun was singular then a penalty score of 1 was added

Later, all the penalty scores of each sentence were added to give a cumulative penalty score for each essay. A list containing all penalty scores for every essay was generated.

Part C2:

To account for the verb tense consistency and missing verb problem the following was done:

1. A Context Free Grammar File is created with all the rules that are necessary but don't over generate. For example, consider "I eating a sandwhich" while this is wrong it is still tagged by the standard parser because of

over generation. In the CFG file created there are rules that flag such sentences and at the same time do not over generate.

2. The determiners were extracted from the corpus of essays we use to make sure to avoid the token not found error, also it helps not to explode the lexicon.
3. In the last step a Bottom Up chart parse is used which is available in the nltk package to parse the sentence. If there is no valid parse returned the mistake count increments by one.

Graduate Students Extra Task:

After collecting all the variables i.e. A, B, C1 and C2, we use a logistic model to fit the coefficients. They are as follows:

```
                          Results: Logit
================================================================
Model:              Logit            No. Iterations:   7.0000
Dependent Variable: Grade            Pseudo R-squared: 0.302
Date:               2018-04-18 21:21 AIC:              76.7304
No. Observations:   69               BIC:              87.9009
Df Model:           4                Log-Likelihood:   -33.365
Df Residuals:       64               LL-Null:          -47.820
Converged:          1.0000           Scale:            1.0000
----------------------------------------------------------------
             Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
----------------------------------------------------------------
intercept   3.3743    1.3249   2.5468  0.0109   0.7775   5.9711
A          -0.1237    0.0569  -2.1741  0.0297  -0.2352  -0.0122
B          -0.2492    0.0708  -3.5206  0.0004  -0.3880  -0.1105
C1          0.0913    0.6063   0.1505  0.8804  -1.0971   1.2796
C2          0.2292    0.0992   2.3105  0.0209   0.0348   0.4237
================================================================
```

Part C3:

According to this webpage there are 2 types of fragments: that have a missing verb or a missing subject.

I used the following rules to find these kinds of fragments that indicates incorrect sentence formation.

I have mainly used 3 rules:

1. Checking if the very first label is NP or a VP or a PP (which is usually ‘S’). Fragments with no verb, usually begin with ‘NP’ and fragments with no subject usually begin with ‘VP’ or ‘PP’.
2. Checking if there was an 'S' visited before SBAR and FRAG are encountered. If not, then it is flagged as mistake.
3. Checks for sentences with a missing verb by doing a DFS over the parse tree.

Part D1:

The following rules were implemented to check if there were any co-referencing errors for third person pronouns:

1. For 3rd person plural pronouns like 'they/them', if there is no re-occurrence of pronoun or if there are no plural nouns present in previous 2 sentences and the current sentence (upto current word) then it's a mistake.
2. Presence of word 'and' Is also checked to denote multiple singular/plural nouns being introduced and is therefore considered while checking for co-referencing of plural 3rd person pronouns.
3. For 3rd person male pronouns like 'he/him', if there is no re-occurrence of male pronoun or nouns like 'student'/'person' (discovered experimentally) or if there are no male names (proper nouns) present in previous 2 sentences and the current sentence (up to current word) then it's a mistake.
4. For 3rd person female pronouns like 'her/she', if there is no re-occurrence of female pronoun or nouns like 'student'/'person' (discovered experimentally) or if there are no female names (proper nouns) present in previous 2 sentences and the current sentence (up to current word) then it's a mistake.

Part D2 (Graduate Student Extra task):

The final score for this part is composed of three separate scores and is calculated as follows:

- Related word counts: This score is calculated by capturing the related set of words in the question (mainly nouns and action words) by using the wordnet sysnets. Higher the match better the score. The motivation behind doing so is to check if the essay contains the words that are related to the question being asked. It makes no sense if the essay is coherent but talks about a totally different topic. So, this score is included.
- Subjectivity analysis: A subjectivity score is calculated between 0 and 1 to account for the fact that good essays are often subjective and expressive of the author's opinions and thoughts. However, objective essays are not penalized in this experiment.
- Topic analysis: Finally, LDA topic analysis model is employed to give out the most probable topics that the essay talks about. Those words are matched against the extended set of words captured from the question in the previous step to account for the over all theme of the essay. Better the match higher the score. The theme has been limited to 1 as we are matching against the question and not trying to extract all the possible topics.

All these scores are scaled and averaged to give the final score between 1 and 5.

Overall Accuracy and a classification report is printed on screen after running the Main_comp.py file with Stanford path argument.

Project Experience:

Sourabh Parime: I executed the same experiment using LSTM technique last summer and got good accuracy but then while implementing this project I learned a lot about the nuances of language processing. I also used many rules taught in the class to tune the code. I faced difficulties while solving the part 1 of the project especially the C2 part and while solving it I learnt to create my own grammar and load it using nltk's functionality which was new.

Kashyap Desai: I learn a lot about implementing different grammatical rules and constraints used to check whether the text written is grammatically correct or not. In part 1 of the project I learned how to use parse trees to check agreement and learnt about the structure of an English sentence while I was tokenizing and counting the number of sentences in an essay. In Part 2, I learnt about co-referencing and how to implement it. All-in-all I got to implement actual/core NLP techniques using libraries like NLTK and Stanford CoreNLP which I wanted to since a long time.

References

[1] Nilani Aluthgedara, ‘Recognizing Sentence Boundaries and Boilerplate’