Automatic Grader - CS421 Project Report

Submitted by,
Kevin George Varkey (kvarke2@uic.edu)
Renu Srinivasan (rsrini7@uic.edu)

Overview

This project aimed at building an automated grader for evaluating English essays written by students. The application parses and analyzes the given essays to check if it follows a set of criteria that we think a well-formed essay should follow. After the analysis, the grader assigns a grade for each of the criteria and also an overall final grade.

We built this application in Java and used Stanford Parsing library to aid with language parsing. Below we describe the various approaches, challenges and takeaways that we obtained from implementing this project.

Approach

The essays are graded on 7 different criteria,

1. Syntax/Grammar
    a. Word order
    b. Subject-Verb agreement
    c. Verb correctness
    d. Sentence formation

2. Semantics/Pragmatics
    a. Text coherence
    b. Topic coherence

3. Length of essay

In order to analyze an essay as per these criteria we took the following approaches. Some of the criteria were analyzed at sentence-level and others were analyzed at essay-level.

1. *Syntax/Grammar* - For this criteria we had to look into each essay at a sentence level. In order to extract individual sentences from the essay we made our own version of sentence splitter to extract sentences. While extracting a sentence we took care of the cases where the user has missed out on period or had tried to combine multiple sentences into one.

After extracting the sentences we tagged them using Stanford Parser library. We made use of POS-tagged words, parse trees and dependency structures returned by the parser to verify against a set of grammar rules that we developed. These rules define the word orderings, verb correctness in a bigram format.

a. *Word-Order Bigram Rules*

For the criteria of 1a, 1b, 1c and 1d we maintained separate lists of word-rules that we know are correct in English syntax. We also had lists of wrong word combination rules so as to identify any incorrect orders. These lists pertain to verb-noun agreement, verb-verb agreement, and other word agreements. We also had rules for certain words that cannot occur as the first/last word in a sentence. So to validate against the set of rules we developed, we parse each sentence of an essay and check if 2 adjacent words follow them.

For example, we know that in English, if a TO tagged word (to) has a verb after it, then it must be a VB tagged word (base form of verb like "write"). So the corresponding valid rule is "TO VB". An example of an invalid word combination is, when there are 2 consecutive verbs, a VBG tagged word (verb gerundive like "writing") cannot have a VB or another VBG following it. So rules for these invalid orders are "VBG VB" and "VBG VBG".

b. *Subject-Verb Agreement - Dependency Tree*

For the criteria of 1b, we extracted each subject dependency from parser's dependency tree. These are obtained from "nsubj", "nsubjpass" and other similar dependency tags. The noun and verb are then checked for agreement including number, person etc. There are additional special checks in cases of a missing subject-verb combination, the subject being a pronoun, consecutive noun-verb rules etc.

c. *Specific Verb Combination Rules*

For the criteria of 1c, in addition to having certain verb-verb rules, we also check for certain combinations that are invalid in a proper sentence. These checks include missing verb, is a gerundive verb followed by another verb, pronoun-verb combinations etc.

d. *General Rules for Sentences*

We also checked the sentences for some general rules that were necessary for proper sentence formation. Some of the rules we used are, conjuction should always concatenate objects of same type, if a sentence starts with a ADVP tagged word then it should be followed by a ",".

2. Semantics/Pragmatics - For this criteria we checked for pronoun resolution and whether the sentences were about the given topic.

a. *Pronoun Resolution*

For the criteria of 2a, we checked for pronoun resolutions in cases of 3rd person pronouns. These resolution checks take care of agreements of gender, number etc. between a pronoun and its antecedent noun. If the agreement is wrong then an error is raised. Presence of 2nd person pronouns is also considered as wrong in an autobiography.

b. *Semantics and Syntax*

For the criteria of 2a, in addition to pronoun resolution, we also incorporated certain checks from sentence formations and syntax as well because even though pronouns agree well, a badly written essay is not semantically correct.

c. *Topic Coherence*

For the criteria of 2b, we checked whether a sentence contains certain words that would suggest that the writer was talking about himself. This is because the essays are assumed to be writing an autobiography. We also check for words that refer to family/relatives etc. After calculating count of such words we used a formula to calculate how close the essay is in representing a autobiography.

3. Length of Essay - For this criteria we check whether an essay has at least 6 sentences in it. We assign a grade for this criteria that is scaled to the number of sentences, if it has less than 6. Else it receives full marks.

For each of the above criteria and checks, whenever there is an incorrect combination or agreement, we add it to an error list for the essay. These errors are divided by the criteria that the error was for. At the end of all analysis we check the number of errors that have accumulated and assign a grade for each criteria. These individual grades then add up for a final grade.

Challenges

On of our earlier attempts dealt with generating all possible grammar rules by iterating through all the tagged sentence from Penn treebank. But after parsing roughly 5% of the Penn treebank corpus we had around 21K grammar rules. We soon realized adopting this strategy would make the system very slow and it would not be as efficient as we wanted.

We also tried to generate all possible word bigram combinations from the Penn treebank. This also proved to be tedious as the above case. That is when we moved to the model where we check for erroneous word bigram combinations that was most relevant to simple sentences.

A major limitation of the software we feel is that it can grade essays only of a certain type. Various rules and formula we used for the grader were designed specifically for the given scenario.

Learnings

Being non-native speakers of English language, we learned a lot about the language and rules that are needed to create syntactically correct English sentences. We also learned to use various features provided by the Stanford Parser libraries and to exploit them as best as possible to check whether the sentences are correctly formed or not.

Improvements

In order to improve our current software we can add more set of rules and checks. Also currently we assume that the sentence the software have to grade are very simple sentences. We could add more features that would enable the software to handle complex sentences as well as other topics than autobiography and family.

References

1. Stanford Parser - http://nlp.stanford.edu/index.shtml
2. Speech and Language Processing 2nd ed (2008) - Daniel Jurafsky and James H. Martin
3. Countless websites on english grammar and sentence formation.