CS 421 – Natural Language Processing – Spring 2015
Term Project: Automatic Grader
(Part 1)

1 Introduction

We will build a system that can automatically score student essays, in the spirit of E-Rater® (`https://www.ets.org/erater/about`). This is a very interesting application of NLP techniques: many individuals are involved in evaluating essays of different sorts, e.g. essays from non native speakers, essays in standardized tests such as TOEFL. In fact, it's part of the business for certain companies, most notably ETS. NLP techniques can help: checking grammar well-formedness is relatively easy, more difficult is evaluating global text properties such as coherence, "flow", etc. This assignment will give you a flavor of what it takes to build such a system. You can read about the research that led to E-Rater in [Burstein and Marcu, 2003; Attali and Burstein, 2006; Shermis and Burstein, 2013; Burstein *et al.*, 2013].

We will set up the project as a competition among the students in the class, divided into teams of two students. You will be given a set of essays written by non native speakers of English, and their grades. This will be your *training set*. You'll be asked to build a system that can give these essays a score, and specifically, as close a grade to the one assigned by the human scorer as possible. You will submit your project in as complete a state as possible (possibly not finished) on 4/27, which is Mon of the last week of classes. We will then evaluate your grader on a *test set* of essays you have not seen and announce the winner(s) the last day of classes. The top three teams will be given small prizes. You will have one more week after 4/27 to finish your project and submit its final version plus a report on the work you did.

Important Notes

- The competition is meant to make the assignment more interesting. For participating in the competition, you will earn 10 points (out of 250 total for project). The grade for your project will not be based on the performance of your system in the competition.
- **You must do the assignment in pairs.** Pick a partner of your choice or talk to the TA to get a random partner assigned. Every group must send an email to the TA stating the names of the two members, **by Wed 4/1**.
- The project should be developed in Java or Python. If your group wants to use any other programming language, you must request approval from your TA.
- The project will be in two parts. The total score for the project is 250 points.

2 The most important information

2.1 All Deadlines

Due dates are as follows (meant as 11:59pm):

	Due	Points
Declare groups	4/1 (Wed)	N/A
Part 1	4/17 (Fr)	100
Competition	4/27 (Mon)	10
Part 2	5/5 (Tue)	140

2.2 Corpus of essays

In the project directory on the class web site (under Assignments), you can find a zip file that contains a corpus of 30 essays written by ESL students — *ESL* stands for *English as a Second Language*. These essays come from a large corpus released by ETS (`https://catalog.ldc.upenn.edu/LDC2014T06`). They are divided in three subdirectory, according to whether they have been scored as *low*, *medium* or *high*. You are provided with both the original essay, and the tokenized version: the tokenized version splits clitics off from the word (i.e., "can't" becomes "ca n't"), and splits the text into "sentences" by inserting end of line after full stops.

These 30 essays were written in response to the prompt *Do you agree or disagree with the following statement? "In twenty years, there will be fewer cars in use than there are today." Use reasons and examples to support your answer.*

2.3 Part 1: Overview of tasks

All that is described here will be discussed in much more detail in the rest of this document, but here's what you will do in Part 1:

1. Set up the general framework for your automatic grader. Your automatic grader will be written as a program that has the following **input** and **output**:

   Input: one essay at a time in input (you must choose whether essays are processed in original form, or tokenized)

   Output: the total score plus all of the component scores for that essay; the total score will also be mapped to one of *low, medium, high*

   You will have to specify in a README file you will supply (see below) how to run the program in order to input several essays one after the other. Your TA will provide further details on the exact submission format.

2. Set up scoring:

   (a) Spell checking

   (b) Exploit POS tagging to evaluate part of the syntactic well-formedness of an essay

   (c) Assess whether the essay is long enough

   (d) Map the assessments from steps 2b and 2c to numeric values

3 Scoring Criteria

Consider the two essay samples below – the given prompt was *The best way to travel is in a group led by a tour guide*. The first one is one of the essays scored *low*, included in its entirety; the second is a paragraph from one of the essays scored *high*.

- A *low* essay:

  > No, i don't agree with the best way to travel is in a group led. I think in this way they will have many probelme. Firt of all, the group led will be not agree together each one want be the led. Second, when they travel they will be fighting all the time. also, they will not listine to each. n the other hand, when you travel with a group wich has one led, they will be better than onather way for severl reasons. First, all the travels will be nice and specifictly . Next, many people like travel with agroup by one led. finally, i don't agree.

- A paragraph from a *high* essay:

  > I would really prefer to travel on my own with plenty on time, but who wouldn't? Unfortunately that is not always possible. It is always nicer to walk looking around at the same time, steping by little shops and cafes, talking to people, asking for directions, going to the places you choose to go to and discovering everything on your own. I think that is the travel ideal for many of us, but we usually have a hard time on finding the time to do it that way, and instead make plans with too many destinations all at once in a small schedule.

Intuitively, we can recognize that the first is rather poor, but the second appears to be well written. Many factors contribute to this assessment. We do not really know which criteria are included in E-Rater, however some that are mentioned in the papers and / or that ESL teachers use are as follows:

1. Syntax/Grammar
   (a) Spelling mistakes
   (b) Subject-Verb agreement - agreement with respect to person and number (singular/plural)
   (c) Verb tense / missing verb / extra verb - is verb tense used correctly? Is a verb missing, e.g. an auxiliary? For example, in the example of *low* essay above, the sequence *will be not agree* is incorrect. Normally the verb *to be* is not followed by another infinitival verb, but either a participle or a progressive tense.
   (d) Sentence formation - are the sentences formed properly? i.e. beginning and ending properly, is the word order correct, are the constituents formed properly? are there missing words or constituents (prepositions, subject, object etc.)?
2. Semantics (meaning) / Pragmatics (general quality)
   (a) Is the essay coherent? Does it make sense?
   (b) Does the essay address the topic?

3. Length of the essay:

   (a) Is the length appropriate? At least 10 sentences were required. Longer essays are in general considered better.

We will evaluate each criterion on a scale from 1 to 5 (1 is lowest, 5 highest), then we will combine them with a linear combination, as provided by the following formula:

$$Final\ Score = 1a + 1b + 1c + 2 * 1d + 2 * 2a + 3 * 2b + 2 * 3a \quad (1)$$

This indicates that more weight is given to 1d, 2a, 2b and 3a since it is absolutely necessary that students are able to form sentence-like structures, understand the question, and write a long enough essay. Given each criterion can vary between 1 and 5, the minimum cumulative score is 12, and the maximum 60. This numeric score needs to be mapped to the three qualitative scores provided with the essays in the corpus, *low, medium, high*.

Your automatic grader should try to get as close as possible to the actual evaluation. In the second part of the project, we'll use either referential expressions and/or a simplified notion of topic as a crude approximation to coherence.

4 Grading criteria for Part 1: Syntactic Well-Formedness

4.1 Spelling Mistakes (Grading criterion 1a)

For spelling, you can use any of the available spellcheckers, for example, for Java:

`https://lucene.apache.org/core/3_5_0/api/contrib-spellchecker/org/apache/lucene/search/spell/SpellChecker.html`
`https://www.ukp.tu-darmstadt.de/software/dkpro-spelling/?no_cache=1`
`http://jortho.sourceforge.net/`

for Python:

`http://stackoverflow.com/questions/13928155/spell-checker-for-python`

Alternatively, you can input words into Wordnet (a large electronic dictionary of more than 150,000 words); if WordNet doesn't return a match, you can assume the word does not exist

`https://wordnet.princeton.edu/`

Note, you are not asked to correct words spelled wrong. Both the Stanford and OpenNLP parser tolerate misspellings well, and are able to infer POS tags and parse trees for unknown words. For example, an ungrammatical sentence such as *Because I think the sience and tecnology are developping* is tagged as follows by both the Stanford and the OpenNLP parsers:

```
Because/IN I/PRP think/VBP the/DT sience/NN and/CC tecnology/NN are/VBP
developping/VBG ./.
```

The same is returned by NLTK parser demo at `http://text-processing.com/demo/`, however NLTK may require more work to have it run in stand-alone mode.

4.2 Agreement and verbs (Grading criteria 1b and 1c)

We will exploit POS tagging and parsing to evaluate the syntactic well-formedness of the essay. In this first part of the project, we will concentrate on POS tagging and the information it can give us. You can use the POS taggers associated with any of the three parsers we experimented with in homework 2: Stanford, OpenNLP, NLTK.

1. Stanford (Java): `http://nlp.stanford.edu:8080/parser/`
2. OpenNLP (in Java): `http://opennlp.apache.org`
3. NLTK (in Python): `http://www.nltk.org/`

Grading criterion 1b: agreement. Consider the sentence *Jessica have 8 years old*: *Jessica* is singular, but *have* is not in the 3rd person singular form. The Stanford tagger returns the following tags for the sentence (OpenNLP and NLTK return similar results):

```
Jessica/NNP have/VBP 8/CD years/NNS old/JJ ./.
```

According to the Penn TreeBank tagset:
NNP = Proper singular noun; VBP = Verb, non 3rd ps. sing. present

Since we know that a proper noun (NNP) is a 3rd person noun, we can identify a violation of agreement here between the subject and the verb.

Grading criterion 1c: verbs. You can also use POS tags to identify many verb tense mistakes made in the essays, and to check whether sentences contain a main verb. The example sentence above does contain a main verb, since *have* is tagged as *VBP*. Of course, this requires that you have identified what sentences there are: these two problems are actually intertwined. Some of the same ideas apply to grading criterion 3a (length) below.

Patterns of errors. You will need to write patterns of errors in terms of sequences of POS tags, and relate those patterns to the specific grading criterion, 1b and 1c. You will then need to transform the number of errors in a score from 1 to 5.

4.3 Number of sentences and length (Grading criterion 3a)

Grading criterion 3a assesses whether the essay is long enough – at least 10 sentences are required; in general, longer essays are preferred. To count the number of sentences, you cannot just count the number of full stops, or end-of-line characters in the tokenized version of the essay. Rather, you should exploit as many cues as you can think of. For example, you can exploit (appropriate) capitalization; and / or you can count the number of finite verbs via their POS tags, but you should take the context of the verb into account, because sentences containing coordinate or subordinate clauses will have more than one main verb. Analyze the essays in the corpus and see what patterns seem to arise.

To understand which lengths of essays are expected, you can compute the average number of sentences in each of the three classes of essays (*low, medium, high*), and then assign a numeric score according to where the number of sentences of the current essays falls with respect to the average of those classes.

5 Notes and Assumptions

- The POS taggers may not necessarily return the correct POS tags at times. This is expected and is a well known phenomenon when building robust applications. You have to make the most out of the taggers.
- You are NOT EXPECTED to do the following, but if you read the papers suggested earlier in this document, you may ADDITIONALLY use some of the techniques outlined in any of those research papers. However, you need to use the criteria outlined earlier, and combine them according to Formula 1. Also BE SURE to cite the source.
- We don't expect your system to be able to score each essay in perfect agreement with the human scorer. On the other hand, scoring an essay as *low* as opposed to its true *high* score is obviously more of a discrepancy than scoring the same essay as *medium*.

6 What and how to hand it in

Only one person in the group will upload through blackboard. You will need to turn in the following:

1. The source code for your entire system that uses the POS tagger, and outputs the scores for criteria 1a, 1b, 1c and 3a.
2. Instructions on how to install/run your system (README). A template for the README file will be included in the project folder after the break.

References

[Attali and Burstein, 2006] Yigal Attali and Jill Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

[Burstein and Marcu, 2003] Jill Burstein and Daniel Marcu. A machine learning approach for identification thesis and conclusion statements in student essays. *Computers and the Humanities*, 37(4):455–467, 2003.

[Burstein *et al.*, 2013] Jill Burstein, Joel Tetreault, and Nitin Madnani. The e-rater automated essay scoring system. *Handbook of automated essay evaluation: Current applications and new directions*, pages 55–67, 2013.

[Shermis and Burstein, 2013] Mark D Shermis and Jill Burstein. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge, 2013.