CS 421 – Natural Language Processing – Spring 2024
Term Project (Part 1)

Deadline: 4/11/24 (Th)
Points: 150

As discussed in the introduction to the project, your task is to grade an English essay written by a non-native speaker as *high* or *low*. **Important clarifications:**

1. This is **not** a text classification project, namely, we are not just interested in the final score, but in the reasons why an essay may be scored high or low (but graduate students will also develop classification models).
2. There's **no autograder** for this project. This will allow you more creativity in assessing the various criteria that comprise the final score.
3. You are **not allowed** to use existing **grammar checkers**. You can use existing NLP tools as detailed below, but you will need to supplement them, or their results.

1 Python and packages

The project needs to be completed using Python (version 3.10+). You will need the following packages:

- `pandas`
- `numpy`
- `nltk` or `SpaCy`
- `scikit-learn` (graduate students only)

Notes

- `NLTK http://www.nltk.org/` and `SpaCy https://spacy.io/` are well-established Python packages to perform NLP tasks, such as sentence tokenization, POS tagging and parsing. You can explore both and decide which one to use.
- **Graduate students:** you will use `scikit-learn https://scikit-learn.org/` to train classifiers for this task.

2 Tasks for the first part of the project

More details are in the remainder of this handout, but as an overview, this is what we expect you to accomplish in the first part of the project:

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

Output: the total score plus all of the component scores for that essay. The total score will also be mapped to one of *low* or *high* (for Part 1, you will report partial results)

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

2. Set up scoring as concerns:

1. Length
2. Spell checking
3. Exploit POS tagging to evaluate part of the syntactic well-formedness of an essay

3. Graduate Students only: you will also set up your framework to train classifiers (details in a companion file).

3 Scoring Criteria

Consider the two essay samples below, both responding to the prompt: *The best way to travel is in a group led by a tour guide.* The first 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: in **bold** the ones you will focus on in part 1:

(a) **Length of the essay**: Is the essay long enough? At least 10 sentences are required. Longer essays are in general considered better.

(b) **Spelling mistakes**

(c) Syntax/Grammar

  (i) **Subject-Verb agreement** -agreement with respect to person and number (singular/plural) (see example in introduction file)

  (ii) **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. (Please refer to the introduction to the project for **another example of verbal grammatical incorrectness** captured by POS tag patterns.)

  (iii) Sentence formation: we can ask whether the sentences are formed properly. We can look at this from two different points of view: constituent well-formedness and/or dependency relations well-formedness.

  - *Constituent well-formedness* may include, among others: a) are sentences beginning and ending properly? b) are the constituents formed properly? c) are there missing words or constituents (prepositions, subject, object etc.)?
  - *Dependency well-formedness* may include, among others: a) does the sentence have at least an `nsubj` or `expl` dependency (`expl` is used in *there is / there are* type of clauses, instead of `nsubj`); b) does the essay include many types of dependencies? if the sentences are very simple, there will be many `nsubj` and `dobj` dependencies, but not many other dependencies.

(d) Semantics (meaning) / Pragmatics (quality at the paragraph/document level):

  (i) Does the essay answer the question / address the topic? we can use word embeddings.

  (ii) Is the essay coherent? We will use a simple algorithm for reference resolution

We will evaluate each criterion other than spelling mistakes 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:

$$Final\ Score = 2 * a - b + c.i + c.ii + 2 * c.iii + 3 * d.i\ + 2 * d.ii \tag{1}$$

Spelling mistakes will be scored from 0 to 4, since if there are no spelling mistakes, no deductions should be taken.

Formula 1 indicates that more weight is given to *a* (length), *c.iii* (sentence formation), *d.i* and *d.iii* since it is absolutely necessary that students are able to write a long enough essay, form sentence-like structures, and understand the question. The numeric score obtained through Formula 1 needs to be mapped to the two qualitative scores provided with the essays in the corpus, low and high.

4 Scoring criteria to be covered in Part 1

4.1 (a) Number of sentences and length

Scoring criterion *a* 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. You can use the sentence tokenizers in one of the tools you will use, however again, just counting the number of "sentences" they return may be insufficient. For example, when given a string like *I want to do well I am sad*, sentence tokenizers may return one single sentence, while most human graders would say, there are two.

To get a more accurate count of sentences, you should exploit as many cues as you can think of. For example, you can use (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 for the two different classes, besides the 10 sentence requirement (and essays may fall short of this requirement), you can compute the average number of sentences in each of the two classes of essays (low, and high); then, you can assign the numeric score from 1 to 5 according to where the number of sentences of the current essay falls, with respect to the average of those classes.

4.2 (b) Spelling Mistakes

To recognize spelling mistakes, you can use any of the available spellcheckers. Of course, you will have not just to recognize spelling mistakes, but to count them and to map the raw number to the range [0,..,4].

NLTK does not have a spell checking module per se, however many other libraries exist for spellchecking in Python, e.g. the `pyspellchecker` `https://pyspellchecker.readthedocs.io/en/latest/`

SpaCy does include a spellchecker.

Note, you are not asked to correct misspelled words. NLTK and SpaCy 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 NLTK:

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

4.3 (c) Syntax and grammar

4.3.1 (c.i and c.ii) Agreement and verbs

In Part 1, we will exploit POS tagging as a means to evaluate the syntactic well-formedness of the essay. You can use the POS taggers associated with either NLTK or SpaCy.

c.i Agreement. Consider the sentence *Jessica have 8 years old*: *Jessica* is singular, but *have* is not in the 3rd person singular form. The NLTK tagger returns the following tags for the sentence:

```
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 - a correct verb would be tagged as VBZ (3rd person).

c.ii Verbs. You can also use POS tags to identify many verb mistakes made in the essays, and to check whether sentences contain a main verb - please see examples above, and in the introduction file. 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 *a* (length) above.

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, c.i and c.ii. You will need to transform the number of errors into a score from 1 (worse) to 5 (best); note that the more the mistakes, the lower the score: namely number of mistakes is inversely proportional to the score.

5 Additional requirements for graduate student groups

Please see graduate student addendum.

6 Notes and Assumptions

1. 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.

2. You are NOT EXPECTED to do the following, but if you read the papers suggested earlier in the introductory 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.

3. 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 most of the *low* essays as *high*, or viceversa, would be a problem.

7 What and how to hand it in

Only one person in the group will upload through gradescope. More specific directions will be given after spring break, but in general, you will need to turn in the following:

1. The source code for your entire system, which will take essays in input, and for each of them, output the scores for subscores *a*, *b*, *c.i* and *c.ii*.
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.
3. A short report on how you addressed the scoring components discussed above, for example how you counted sentences, which POS patterns you identified, etc.
4. For graduate groups: additional information in the graduate student addendum.