CSCI 544: Applied Natural Language Processing

Course Project

Jonathan May and Nanyun Peng

Out: **12. September 2018**
Proposals/Teaming Due: **26. September 2018**
Re-submission Due: **24. October 2018**
Completion Date: **30. November 2018**

The goal of the project will be to define a new natural language classification task, to collect and annotate data for this task, to build a classifier that attempts to label held-out data, and to analyze your results. For this project **only** you may form a team of up to **four**. Teaming decisions must be made by the Proposal/Teaming date (see above) and may not be changed; the entire team will receive the same score regardless of the actual work distribution.

The major emphasis of this project is your collection and annotation data. A secondary, but still important focus, is your analysis of your classifier's behavior and a comparison to baseline approaches. Far less important to us are the particulars of your classifier code.

1 Task Specification and Approach

Determine a natural language classification task that, as far as you know, there is no data available for. By *natural language classification task*, we mean assigning a categorical label to some natural language text. Here are some examples about natural language text (you need not use any of these):

- English news articles about cricket
- Korean reviews of yachts
- 18th century Spanish poems
- Sentences from Tagalog research articles about cancer

And here are some examples about labels you could collect:

- Gender of the writer
- Sentiment (positive or negative)
- Role the text plays in a larger text (introduction, persuasion, argumentation, counterpoint)
- Voice of a sentence (active/passive)
- How funny is the text on a scale of 1–5

However you need not (indeed, you should not) be constrained to just these! You will have more fun doing the project if you choose data and labels that are interesting to you. Make sure to investigate by searching the web to be reasonably sure your collection is novel.

1.1 How to collect the labels

Collecting labels of data can be time consuming and expensive, but it need not be. Here are some suggestions:

- Be clear about the labels you are trying to create. If you are going to be asking another human to provide annotation you should come up with clear guidelines that capture what you are trying to label and lead to high *inter-annotator agreement*: if several people label the same data independently, ideally they will provide the same label.
- Collaborate with several other teams – agree to label their data if they agree to label yours.
- Use Mechanical Turk, Figure Eight, or some other human annotation service. Note that this costs money; if you choose this route you do so at your own expense. You must also craft your instructions and monitoring carefully to prevent annotators from providing nonsense responses.
- Find an existing dataset that you can use to create the annotation you want. For example, reviews with user profiles can be converted into gender of the writer by looking up the common male and female names from Social Security office. Note that these *automatically generated labels* can be noisy. You should propose methods to clean up the annotations.

What to turn in:

- **By 26 September: [270 points; up to 2 late days may be used for this part (if available)]**

  1. The name and description of the task, uploaded to Crowdmark. The description should:

     (a) explain why this is an interesting data set and what would be learned from building a good classifier of this data

     (b) describe how the data will be collected and labeled

     (c) include an estimate of how many annotated examples you will be able to collect (note: this number should be at least 500. If you think you have a good idea but one that will result in fewer than 500 entries, write this up as well, but be warned that it may be rejected).

  2. Teaming information, uploaded to crowdmark and in your dataset on Vocareum (see below).

  3. A small sample of the dataset (minimum of 10 annotated instances) using the json format (and, optionally, the helper script) provided. For each of at least two label types, at least two examples with that label type must be provided. In other words, your sample data should not be 10 instances, each with a separate label, nor should it be 10 instances, each with the same label. The json file has the following requirements:

     (a) named `proposal.json` and located in the main user directory of Vocareum

     (b) contains the following metadata entries:

        i. `description` describes the data in a few sentences. This could match the description you submit to crowdmark.

        ii. `authors` contains the entries `author1`, `author2`, ... Each of these is a key that points to the names of a person on your team. `author1` should be the name of the person whose crowdmark/vocareum account will contain your submission.

        iii. `emails` contains the entries `email1`, `email2`, ... Each of these is a usc email address that members of your team are registered for the class under. `email1` should be the email address of `author1` and so on.

iv. `corpus` contains multiple entries each with a `data` key pointing to the input data item and a `label` value pointing to the annotation.

See the provided `sample.json` file which was formed from the provided tab-separated `sample.data` file using `group_project.py`. You are welcome to use this script to create your file but it is not necessary as long as your file is compliant.

- **By 24 October [no late days may be used; see below]:**

  1. If you are not notified that there was a problem with your 9/26 submission, nothing is due on this date.
  2. If your task idea is not acceptable (e.g. it's too similar to an existing widely-available resource, it's not a human language task, it's trivial to classify, it's impossible to label consistently), a re-submission is due. The re-submission will not be scored, but failure to submit by the deadline will incur 10% penalty on the total grade per day late. Grace periods may not be used for this submission.

- **By 30 November: [630 points; up to 2 late days may be used for this part (if available)]**

  1. The final report, including:
     (a) A description of how the data was collected [100 points]
     (b) A description of how the data was labeled [100 points]
     (c) A description of the classifier approach [100 points]
     (d) An analysis of the results (preferably with charts and figures) showing comparison to at least one reasonable baseline approach[1]. [100 points]
     (e) The final annotated dataset, uploaded to Vocareum, with at least 500 non-duplicate annotated items [100 points], however...
        i. +15 points for a dataset with at least 1,000 non-duplicate items
        ii. +7 points for a dataset with at least 5,000 non-duplicate items

[1] random is not a reasonable baseline

iii. +4 points for a dataset with at least 10,000 non-duplicate items

iv. +4 points for a dataset with 10% or more non-duplicate items than any other project in the class

The dataset should follow the following format (basically, the same as for the preliminary report but with a different file name and larger):

i. named `final.json` and located in the main user directory of Vocareum

ii. contains the following (meta)data entries:

A. `description` describes the data in a few sentences. This could match the description you submit to crowdmark.

B. `authors` contains the entries `author1`, `author2`, ... Each of these is a key that points to the names of a person on your team. `author1` should be the name of the person whose crowdmark/vocareum account will contain your submission.

C. `emails` contains the entries `email1`, `email2`, ... Each of these is a usc email address that members of your team are registered for the class under. `email1` should be the email address of `author1` and so on.

D. `corpus` contains multiple entries each with a `label` field and a `data` field. This is the main content of the file.

See the provided `sample.json` file which was formed from the provided tab-separated `sample.data` file using `group_project.py`. You are welcome to use this script to create your file but it is not necessary as long as your file is compliant.

(f) The code for running the classifier(s) you used and reproducing the results in your report, uploaded to Vocareum, with instructions on how to reproduce your results. [100 points]