CSC470-01 NLP
Final Project
Due: May 17, 2020 by 11:59PM

Final Project Title: Text Classification using the COVID-19 Open Research Dataset (CORD-19)

1 Project Description

In this project you will gain hands-on experience working in a team to implement text classification systems from scratch. You will also gain additional hands-on practice applying and continuing to increase your foundational corpus processing and language modeling skills. You will gain experience working with another dataset that you haven't worked with yet this semester, dealing with the timely topic of COVID-19. You will implement Naive Bayes text classifiers using bag of words representation models. You will also gain additional practice and experience in the production of professional well documented software deliverables and reports.

2 Dataset Description

On March 16, 2020 the White House released the statement "Call to Action to the Tech Community on New Machine Readable COVID-19 Dataset". The statement is accessible from the whitehouse.gov website here: `https://www.whitehouse.gov/briefings-statements/call-action-tech-community-new-machine-readable-covid-19-dataset/`.

For the purpose of self-containing this project description, here is a paste of a brief summary from the kaggle.com website (`https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge`): "In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 59,000 scholarly articles, including over 47,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new insights in support of the ongoing fight against this infectious disease. There is a growing urgency for these approaches because of the rapid acceleration in new coronavirus literature, making it difficult for the medical research community to keep up."

A preprint describing the dataset [1] is located here: `https://arxiv.org/pdf/2004.10706.pdf`.

3 Teams

Team Assignments will be emailed to students on May 9, 2020.

4 Tasks

4.1 T1: Familiarize yourself with the CORD-19 dataset and with JSON (JavaScript Object Notation)

Download the CORD-19 dataset, which is available by clicking on the approprate link at this site: `https://www.semanticscholar.org/cord19`. For the purposes of this project, you can use the biorxiv/medrxiv subset of CORD-19. Also download the metadata.csv file (separate download) from the same site as you will need the information in it. Observe that there is a lot of information on the site as well as the kaggle.com site referenced in section 2 above in case you want to read more about the dataset. Once you have the dataset unpacked you will see that it has the contents of scientific articles stored in a JSON format in the directory pdf_json. Observe that you can find the metadata about each of the articles by inspecting the metadata.csv file. For example, for the file 2fe18190f7b54e589dccf1d3bbfd73e87bfd2ab9.json, by inspecting the JSON file you can see that it is for the paper with title "Animal virus ecology and evolution are shaped by the virus host-body" authored by Jan Slingenbergh. By inspecting the metadata.csv file, you can find that the publish_time column has a value of 12/10/18. In my download the entry for this paper in the metadata.csv file is on line 38561.

Another example is paper ffbf8ea9948d73572fd052a74afa01b19e6758a3. This paper has title "Planning horizon affects prophylactic decision-making and epidemic dynamics" authored by Luis G Nardin from University of Idaho in Moscow, ID, United States along with several other authors as you can see from the JSON file. The first sentence of the abstract is: "Human behavior can change the spread of infectious disease." From the metadata.csv file you can see that the publish_time for this paper is 8/12/16. In my download the entry for this paper in the metadata.csv file is on line 35697.

If you're not familiar with JSON, then please search it online and learn about it. It's a very commonly used data storage format that's easy to understand. It's frequently used as a data storage format in NLP as well as more generally in computer science. You will need to process the JSON files to extract the parts of the files that are of interest to you for completing your tasks. Familiarize yourself with JSON.

Familiarize yourself with the dataset and the metadata.csv file and see how they are related. Make sure you understand how to obtain the publish_time value for an article.

4.2 T2: Process the data in order to prepare it for the experiments

Write software to transform the dataset into the format that we will need for subsequent tasks. You are allowed to use preexisting programming language libraries for processing JSON in order to extract the contents of the files if you wish. For your text classification tasks, you will want to only extract the text content of the articles that are represented in the JSON format. You might also want to keep track of the paper id so your software can use the id to look up the publish_time value from the metadata.csv file.

You will need to write software to do the following. Place all the publish_time dates of the articles in your download into a sorted order by publish_time from the earliest published to the most recently published. Let the median date in this ordered list be referred to as medianDate. So half the articles should be published before or on the medianDate and the other half should be published on or after the medianDate. Let those that are published before or on the medianDate be called the $beforeSet$ and let those that are published after or on the medianDate be called the $afterSet$. The total set of articles in your download should be equal to $beforeSet \cup afterSet$.

Write software to randomly select 90% of the $beforeSet$ and 90% of the $afterSet$. Let those sets be called $trainBefore$ and $trainAfter$, respectively. Let the remaining 10% of the $beforeSet$ and the remaining 10% of the $afterSet$ be called $testBefore$ and $testAfter$, respectively. Your training data for your first experiment will be $trainBefore \cup trainAfter$. Your test data for your first experiment will be $testBefore \cup testAfter$.

4.3 T3: Implement naive bayes text classification, run it on the data, and analyze the results

Write software to implement Naive Bayes Text Classification as we discussed it in class. You'll need to write software to convert the text of each article in your dataset to a bag of words representation. In this classification task, your two categories are "published before your medianDate" and "published after your medianDate". You may use Add-1 smoothing and you can let your Vocabulary be all the unigrams in the entire dataset. Train your model on your training data created in Task T2. Test your model on your testing data created in Task T2 and compute the accuracy of your model, that is, the percentage of the test cases for which your model gave the correct classification. For example, your model gives the correct classification if it predicts the test case is in the class "published before the medianDate" and if that was actually true or if it predicts the test case is in the class "published after the medianDate" and if that was actually true. Otherwise your model made an incorrect classification. Analyze the results that you obtain and prepare a detailed report with the statistics of your model's accuracy and your analysis of the situation.

4.4 T4: Cross entropy experiments

Train a unigram language model on the $beforeTrain$ data and another one on the $afterTrain$ data. Compute four cross entropies: $beforeTrain$ model on $beforeTest$ data, $beforeTrain$ model on $afterTest$ data, $afterTrain$ model on $beforeTest$ data, and the $afterTrain$ model on the $afterTest$ data. To avoid underflow and to increase efficiency of your program, using a similar derivation as I taught you for working in log space to compute perplexity, do the same for computing the cross entropies by making sure you take sum of logs instead of the product of all the probabilities and then the log of the product. For example, here's how you can compute the cross entropy for the model learned on $beforeTrain$ tested on $afterTest$. Let the learned unigram model be called $P_{beforeTrain}$, and let the text contents of $afterTest$ concatenated together be the sequence of n words W_1^n. Then the cross entropy can be computed as $H_{beforeTrain-afterTest}(W_1^n) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{beforeTrain}(w_i)$

to avoid underflow and increase efficiency. The other three cross entropies, $H_{beforeTrain-beforeTest}$, $H_{afterTrain-afterTest}$, and $H_{afterTrain-beforeTest}$, can all be computed analogously.

Prepare a detailed report of your four cross entropies and your analysis of the situation.

5 Deliverables

5.1 D1

All source code neatly documented with a README.txt file saying what is what and where everything is and giving clear usage instructions. Your source code can be written in the computer programming language of choice, as long as it can run with the standard STEM 112 Mac lab configuration from terminal command-line prompt (same as for Projects 1, 2, 3, and 4). Your code must not be wastefully inefficient and slow to run. The README.txt file should be a guide to where to find the contents of your project and provide clear and detailed usage instructions, but it should not contain all of the deliverables of your project. They should be put in separate files.

5.2 D2

Your train and test sets that were created in T2.

5.3 D3

Your results and writeup of analyses from T3 and T4.

5.4 D4

Responses to the following questions:

- What was easy about this assignment?
- What was challenging about this assignment?
- What did you like about this assignment?
- What did you dislike about this assignment?

- What did you learn from this assignment?

6 Submission Format

Please submit your entire package as a tar.gz file via file upload to Canvas.

References

[1] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Michael Kinney, Ziyang Liu, William. Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset. *ArXiv*, abs/2004.10706, 2020.