NLP4CSS: Homework #1

Due 11:59pm EST on February 12th, 2024

Instructor: Anjalie Field; Lead TA: Samuel Lefcourt; special thanks to Carlos Aguirre

Guidelines. This assignment is to be completed **individually**. Be sure to comply with course policies on the course website.
Starter Code. Starter code is provided.

```
HW1
|- log_odds.py
|- topic_models.py
|- word_embeddings.py
|- data
   |- cr_metadata.csv
```

Submission. This homework has written and coding components. For coding, you will complete the python files and submit everything else to gradescope. For the written part, you will write your answers in a PDF named `README.pdf` and also submit it to gradescope. Course Entry Code: YDPR48. Your final submission should have all the completed python files as well as your `README.pdf`.
Data. In this homework we will use a sample from the *Congressional Record*, which records all speeches given on the floor of the US Congress. This is a classical corpus used in many political science studies. Normally, to use any data for analysis, you would have to preprocess the text data, however, the preprocessing has been done for us already.

1. Corpus originally constructed in plaintext format by (Gentzkow et al., 2018)

2. Prepared for NLP methods (`word2vec` models) by (Rodriguez and Spirling, 2022): remove non-alphabetic characters, lowercase, and remove words of length 2 or less, then filter to Congressional sessions 111-114 (Jan 2009 - Jan 2017) and to speakers with party labels D and R.

3. Stewart and Keith (2022) converted the plaintext R-data files to txt and csv, subsampled the corpus for convenience.

Additionally, throughout the homework problems we utilize a list of curated political keywords that are useful in exploring the performance of word embeddings models according to human raters which was originally collected by (Rodriguez and Spirling, 2022).

```
politics_words = [
                'freedom', 'justice', 'equality', 'democracy',
                'abortion', 'immigration', 'welfare', 'taxes',
                'democrat', 'republican'
               ]
```

Problem 1: Log-Odds Ratio Informative Dirichlet Prior

In class, you learned about methods to measure word statistics in corpora. In this section, you will implement the Log-Odds Ratio Informative Dirichlet Prior method, as well as additional applications that may deliver useful insights on our data. Throughout this section we will use notation as described in Monroe et al.

(2008), and editing the `log_odds.py` file.

Part A

(10 Points) **Complete the log-odds ratio code**. We can define the frequency of words being in a corpus through *odds*, that is, the observed "odds" O of word w in group i's usage are defined as:

$$O_w^{(i)} = \frac{f_w^{(i)}}{1 - f_w^{(i)}}$$

Where $f_w^{(i)} = y_w^{(i)}/n^{(i)}$ is the normalized proportion of word w given word counts y_w and total number of words ($n = \sum_{w=1}^{W} y_w$). However, the lack of symmetry between groups makes odds ratio hard to compare across groups, therefore, we will take the *log*, resulting in

$$L_w^{(i)} = log(\frac{f_w^{(i)}}{1 - f_w^{(i)}}) = log(\frac{y_w^{(i)}/n^{(i)}}{1 - y_w^{(i)}/n^{(i)}}) = log(\frac{y_w^{(i)}}{n^{(i)} - y_w^{(i)}})$$

And when comparing two groups, say in our dataset democrats (D) and republicans (R), the log odds ratio becomes:

$$L_w^{(D-R)} = log(\frac{y_w^{(D)}}{n^{(D)} - y_w^{(D)}}) - log(\frac{y_w^{(R)}}{n^{(R)} - y_w^{(R)}})$$

Part B

(10 Points) **Complete the log-odds ratio with prior code**. While the *log-odds ratio* is a helpful metric, it often is dominated by low frequency words. Addressing this issue, we can first model the usage of words for the full collection of documents, *prior*, and use that as a starting-point for the group-specific analysis. We will implement a prior in our log-odds ratio as following:

$$\Omega_w^{(i)} = \frac{y_w^{(i)} + \alpha_w}{n^{(i)} + \alpha_0 - y_w^{(i)} - \alpha_w},$$

$$\delta_w^{(i-j)} = log(\frac{\Omega_w^{(i)}}{\Omega_w^{(j)}}),$$

where $\alpha_0 = \sum_{w=1}^{W} \alpha_w$. For our assignment, the prior α will be the complete dataset (if we compare D and R, then $\alpha_w = y_w^{(D)} + y_w^{(R)}$), however, we could alternatively impose a more informative prior by using a much bigger background corpus that is independent of our dataset to estimate the complete distribution of word usage. Finally, we use an approximation of the variance:

$$\sigma^2(\delta_w^{(i-j)}) \approx \frac{1}{(y_w^{(i)} + \alpha_w)} + \frac{1}{(y_w^{(j)} + \alpha_w)}$$

, since infrequently spoken words have higher frequency variance in our groups, to obtain a final statistic:

$$\zeta_w^{(i-j)} = \frac{\delta_w^{(i-j)}}{\sqrt{\sigma^2(\delta_w^{(i-j)})}}$$

Part C

(10 Points) **Complete the issue evolution code.** One of the applications of obtaining word statistics is to investigate the dynamics of word usage across time. In a naive implementation, we would separate the data in discrete time periods and calculate the word statistic as before for each group of documents across

time. However, this would result in noisy counts, especially if our time intervals are small. Instead, we apply a smoother to the data: we calculate a b-window moving count, m, of word use, and apply an exponential smoother, with a smoothing factor A,

$$m_{wt}^{(i)} = \sum_{\tau=t-b}^{t} y_{w\tau}^{(i)},$$

$$s_{w(b)}^{(i)} = m_{w(b)}^{(i)},$$

$$s_{wt}^{(i)} = Am_{wt}^{(i)} + (1-A)s_{w(t-1)}^{(i)}$$

The second equation denotes that we start at $t = b$. We can then calculate $\zeta_{wt}^{(i)}$, using $s_{wt}^{(i)}$ instead of $y_{wt}^{(i)}$. [Note: exponential smoothing is more commonly applied to moving average computation, but in this case we are smoothing count variables for a fixed window size, so there is no need to normalize by window size]

Part D

Now that we have our analysis tools, answer the following questions by using the code from **Part B** & **C**:

1. (5 Points) What are the top words used by women Democrats compared to men Republicans?
2. (5 Points) Rodriguez and Spirling (2022) curated a list of 10 political words to explore the analysis of models according to human raters. `[‘freedom’, ‘justice’, ‘equality’, ‘democracy’, ‘abortion’, ‘immigration’, ‘welfare’, ‘taxes’, ‘democrat’, ‘republican’]`. What are top 2 words that had the changed the most in usage between Democrats and Republicans across congressional sessions?

Problem 2: Topic Models

In class you learned about topic models, in this section we will not ask you to implement Latent Dirichlet Allocation (LDA) (lucky you), rather we will use the `gensim` implementation to use an already trained topic model (which was trained with subset of the data) on our congressional speech datasets. For this problem, you will be editing the `topic_models.py`
In the previous section, we investigated how the keywords usage changes between political parties across time. This direct metric has some drawbacks as it may not be desirable to compare word usage in varied contexts. Instead, we will now narrow the context of the analysis by measuring the change across time within a specific topic. This is the preferred method in practice (Monroe et al., 2008):

$$\delta_{kw}^{(i)} = log(\frac{y_{kw}^{(i)} + \alpha_{kw}}{n_k^{(i)} + \alpha_{k0} - y_{kw}^{(i)} - \alpha_{kw}}),$$

Where we take topic-specific counts for words and prior, e.g. $y_{kw}^{(i)}$ is the word-count of word w in documents of topic k within the group (i). Thankfully, the only thing we need to update from the previous section is how we calculate the counts.

Part A

1. (10 Points) Complete the code to assign documents to topics. Use the `LdaMulticore.get_document_topics()` function to obtain the topic distribution for each document in our dataset, and assign the topic with the highest score to each document. Answer the following:
2. (5 Points) Create a table that lists the change in the following word usage over the congressional sessions: `['abortion', 'justice', 'freedom']` within documents related to **healthcare** (topic 5). How did the political party assigned to each word change? Compare your findings to the output of `part 1D` which calculated the change in words across all documents. How does using a more specific context (congressional speeches related to healthcare) change the results?

Problem 3: Word Embeddings

In this section, we will use the `word2vec` gensim implementations to learn vector representations of words and use them to analyze language variation and change. You will be editing the `word_embeddings.py` file.

Part A

1. (10 Points) **Train `word2vec` models.** Train models to learn word embedding matrices for speeches of each party using the gensim library.
2. (5 Points) Using the code and the models you trained, we can attempt to answer the question: How does the usage of the word "`taxes`" change between democrats and republicans? **Hint**: examine the top 10 nearest neighbors of `taxes` in the democrat and republican models.

Part B

(10 Points) **Complete the code for word embedding space alignment.** The comparison we made in the previous section was good enough to suggest similarities and differences but not enough to conduct a comparative analysis. For this it would be ideal to compare vectors across models, however this results in nonsensical conclusions as each embedding space was created independently and the vector spaces are not directly comparable. To compare them, we first have to align the embeddings.

One way to align these embedding matrices is called Procrustes alignment, which uses singular value decomposition. Defining $\mathbf{W}^{(g)} \in R^{|V|\times d}$ as the embedding matrix for group g, we align across groups g_i, g_j while preserving cosine similarities by optimizing:

$$\arg\min_{Q^TQ=I} ||W^{g_i}Q - W^{g_j}||_F$$

The expression is minimized for $Q = UV^T$ where $\text{SVD}((W^{g_i})^T W^{g_j}) = U\Sigma V^T$.

Part C

Answer the following questions:

1. (5 points) Using the models you trained in part A, rank by similarity the political keywords from (Rodriguez and Spirling, 2022) as used by Republicans and Democrats.
2. (10 points) Train Republican and Democrat `word2vec` models across each congressional session and calculate the average cosine distance over the 10 keywords we have been using during the homework to answer this question. Have congressional speeches around specific issues become increasingly polarized over the years?

3. (5 points) We will use the cosine distance metric from the previous answer as a proxy for polarization: distance between Republican and Democrat embeddings for each words means they have been used in distant contexts. What are some possible limitations of this approach?

References

Matthew Gentzkow, Jesse M Shapiro, and Matt Taddy. 2018. Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts. In *URL: https://data. stanford. edu/congress text*.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Pedro L Rodriguez and Arthur Spirling. 2022. Word embeddings: What works, what doesn't, and how to tell the difference for applied research. *The Journal of Politics*, 84(1):101–115.

Ian Stewart and Katherine Keith. 2022. Democratizing machine learning for interdisciplinary scholars: Report on organizing the nlp+ css online tutorial series. *arXiv preprint arXiv:2211.15971*.