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 (**?**)
2. Prepared for NLP methods (`word2vec` models) by (**?**): 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. **?** 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 (**?**).

```
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 **?**, 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

Problem 1 continued on next page. . .

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,

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

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

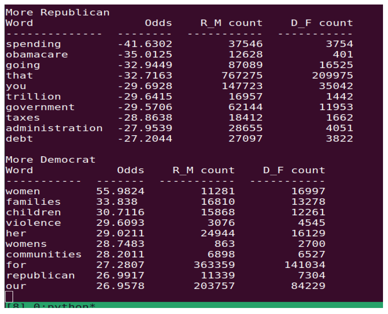
$$s_{wt}^{(i)} = Am_{wt}^{(i)} + (1-A)s_{w(t-1)}^{(i)}$$

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?



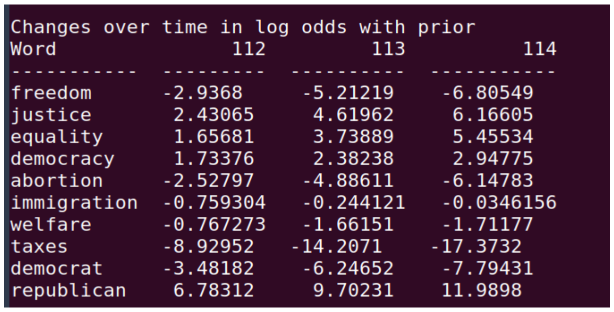
```
More Republican
Word                 Odds    R_M count    D_F count
--------------  --------  -----------  -----------
spending         -41.6302        37546         3754
obamacare        -35.0125        12628          401
going            -32.9449        87089        16525
that             -32.7163       767275       209975
you              -29.6928       147723        35042
trillion         -29.6415        16957         1442
government       -29.5706        62144        11953
taxes            -28.8638        18412         1662
administration   -27.9539        28655         4051
debt             -27.2044        27097         3822

More Democrat
Word              Odds    R_M count    D_F count
-----------  -------  -----------  -----------
women        55.9824        11281        16997
families     33.838         16810        13278
children     30.7116        15868        12261
violence     29.6093         3076         4545
her          29.0211        24944        16129
womens       28.7483          863         2700
communities  28.2011         6898         6527
for          27.2807       363359       141034
republican   26.9917        11339         7304
our          26.9578       203757        84229

[8] 0:python*
```

The top words by Women Democrats are women, families, and children. The top words by men Republicans are spending, obamacare, and going.

2. (5 Points) **?** 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?



```
Changes over time in log odds with prior
Word               112          113          114
-----------  ---------  -----------  -----------
freedom      -2.9368     -5.21219     -6.80549
justice       2.43065     4.61962      6.16605
equality      1.65681     3.73889      5.45534
democracy     1.73376     2.38238      2.94775
abortion     -2.52797    -4.88611     -6.14783
immigration  -0.759304   -0.244121    -0.0346156
welfare      -0.767273   -1.66151     -1.71177
taxes        -8.92952   -14.2071     -17.3732
democrat     -3.48182    -6.24652     -7.79431
republican    6.78312     9.70231     11.9898
```

Looking at absolute differences, the words taxes and republican changed quite drastically. Taxes changed by -8.44368 points, and republican changed by 5.20668 points.

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 (**?**):

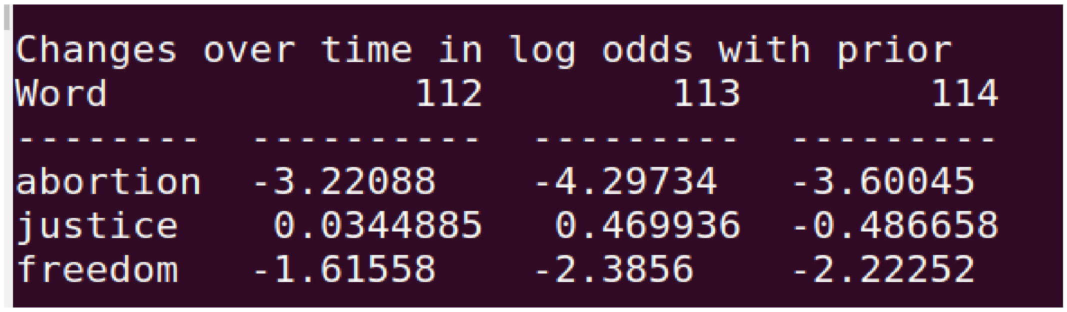
$$\delta_{kw}^{(i)} = log(\frac{y_{kw}^{(i)} + \alpha_{kw}}{n_k^{(i)} + \alpha_{k0} - y_{kw}^{(i)} - \alpha_{kw}}),$$

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?



```
Changes over time in log odds with prior
Word              112          113         114
--------  ----------  ---------  ---------
abortion  -3.22088    -4.29734   -3.60045
justice    0.0344885   0.469936  -0.486658
freedom   -1.61558    -2.3856    -2.22252
```

There's quite a big difference. When I specified the context to just be congressional speeches related to healthcare, you notice that there is a noticeable dip in general change (it changes a lot less) than before when it was open/general to all documents.

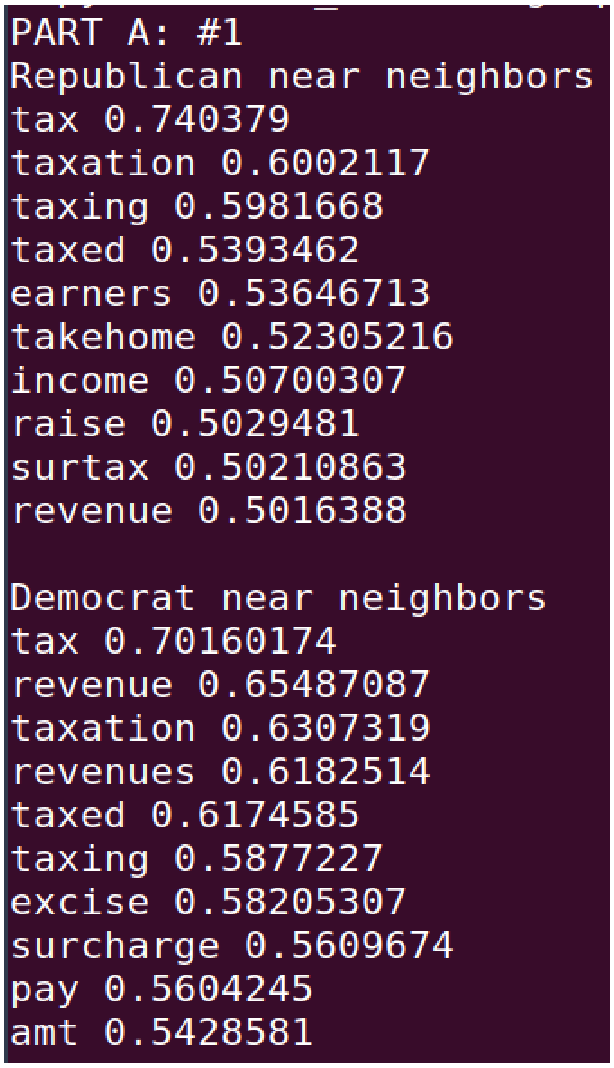
Problem 3: Word Embeddings

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 A: #1
Republican near neighbors
tax 0.740379
taxation 0.6002117
taxing 0.5981668
taxed 0.5393462
earners 0.53646713
takehome 0.52305216
income 0.50700307
raise 0.5029481
surtax 0.50210863
revenue 0.5016388

Democrat near neighbors
tax 0.70160174
revenue 0.65487087
taxation 0.6307319
revenues 0.6182514
taxed 0.6174585
taxing 0.5877227
excise 0.58205307
surcharge 0.5609674
pay 0.5604245
amt 0.5428581
```

Republicans have taxes related to words like "takehome" and "earners" which probably has an emphasis on how you as the individual are earners who make money and should take home more, trying to lower taxes. Democrats on the other hand have more unique words like "pay" and "amt" where they may be looking at it in a more general manner, taxes to fund programs probably and really focusing on who should pay the taxes, and what they are paying for. This is also really apparent with the emphasis on revenue being at the bottom for republicans, since they care less about the amount versus democrats who have it 2nd since the total amount matters to them (wishing to spend more, possibly to increase the tax revenue).

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:

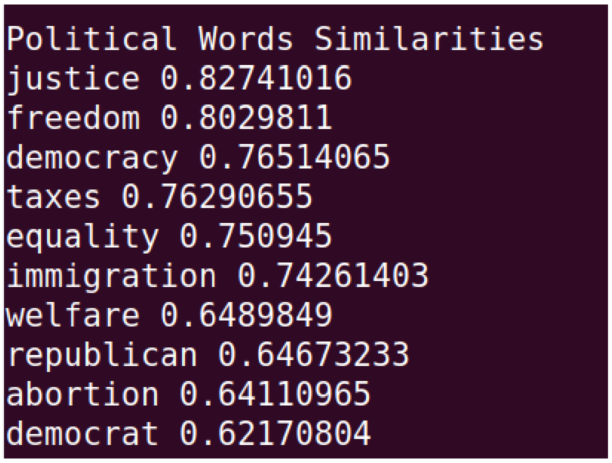
$$\arg\min_{Q^TQ=I} ||W^{g_i}Q - W^{g_j}||_F$$

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

Part C

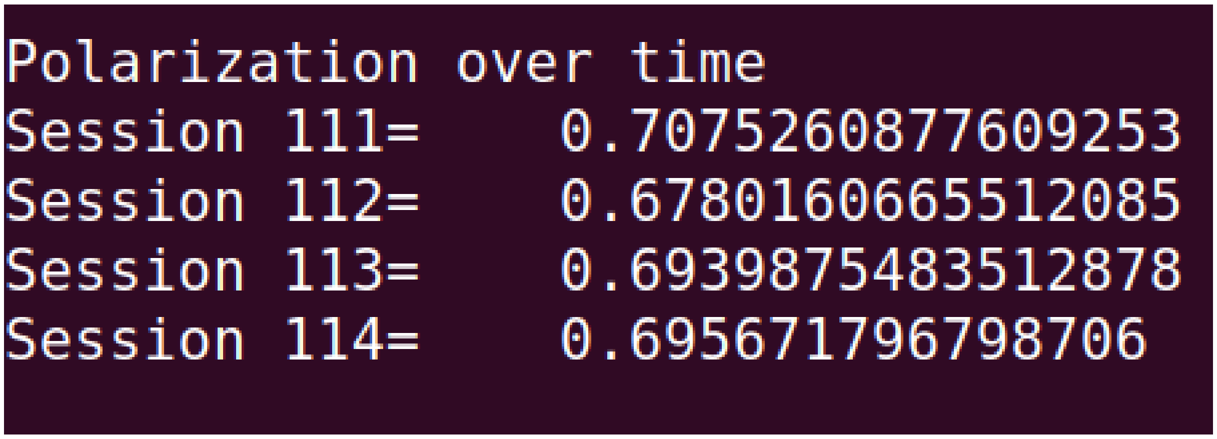
Answer the following questions:

1. (5 points) Using the models you trained in part A, rank by similarity the political keywords from (**?**) as used by Republicans and Democrats.



```
Political Words Similarities
justice 0.82741016
freedom 0.8029811
democracy 0.76514065
taxes 0.76290655
equality 0.750945
immigration 0.74261403
welfare 0.6489849
republican 0.64673233
abortion 0.64110965
democrat 0.62170804
```

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?



```
Polarization over time
Session 111=    0.7075260877609253
Session 112=    0.6780160665512085
Session 113=    0.6939875483512878
Session 114=    0.695671796798706
```

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?

   One issue is that word embeddings cannot distinguish the finer use cases of the same word. For example, something like "legalize" being referred to in an immigration or drug/marijuana policy standpoint could be an issue. Additionally, even if a word is used in a very similar context among both parties, so small cosine distance, they could be on a different side/opinion with high polarization that isn't reflected in a metric like word embedding. Overall, the biggest issue is the lack of context from word embeddings (mushing it all into one word), both on same word diff cases or words being used similar context so seemingly similar but actually very polarizing views.