Information Retrieval Final Project

Barr Israel, 321620049
Stanislav Tokarev, 337708978

Introduction:

In this project we will complete 3 information retrieval tasks on the datasets given to us:

1. Language modeling:

We will train a language model on Political Bias documents, as is, without stopwords, after casefolding and after stemming and analyze the differences between the different models.

2. Text Classification:

With the added News Fairness documents, we will train 4 classifiers to classify between Political Bias documents and News Fairness documents and analyze and compare the results of each classifier. We will also use 10-Fold cross validation to improve the comparison.

3. Text Clustering:

With the added Social Bias and Bias Mitigation documents, we will cluster all the documents using a K-means algorithm into 4 clusters, analyze the result and explain the different clusters that were created.

We will solve these tasks using different models and methods mostly from the sklearn library, and some manual exploration of the results in Pandas DataFrames and NumPy matrices.

The documents were all preprocessed in each task by loading all the text files into a Pandas DataFrame and adding a label if needed(for tasks 2 and 3) and saving it to a parquet file for use in the actual fitting and analysis.
The parquet file for each task is attached inside the folder of that task but the documents processed to create it are not, only the parquet files are needed for analysis.

Question 1:

1. preprocessed as described in the intro:

```
                                                  text
0    Full Terms & Conditions of access and use can...
1    AMERICANS' VIEWS OF POLITI CAL BIAS IN THE AC...
2    Univ ersity of Chicago Law School Univ ersity...
3    Analyzing Political Bias and Unfairness in Ne...
4    A Question of Balance — 1Running head: A QUES...
5    Automating Political Bias Prediction Felix Bi...
6    Political Behavior, Vol. 24, No. 2, June 2002...
7    ORIGINAL PAPER Biased Judgment of Political B...
8    Sociological Forum, Vol. 14, No. 1, 1999 Mini...
9    © 2006 Sigma Xi, The Scientific Research Soci...
10   See discussions, st ats, and author pr ofiles...
11   Predicting Political Biases Against the Occup...
12   Detecting Political Bias Trolls  in Twitter D...
13   royalsocietypublishing.org/journal/rsif Resea...
14   Univ ersity of Chicago Law School Univ ersity...
15   BulletinofLatinAmericanResearch , Vol. 34, No...
16   See discussions, st ats, and author pr ofiles...
17   Northwestern University School of Law Public ...
18   Bias In Perceptions of Public Opinion Among P...
19   UC Berkeley Working Papers Title How the case...
20   Identifying Political Bias in News Articles K...
21   Is There a Political Bias? A Computational An...
22   See discussions, st ats, and author pr ofiles...
```

Tokenization is done inside CountVectorizer, along with stopwords removal and casefolding, for stemming we created a custom CountVectorizer that applies the nltk Porter stemmer to each word and a custom CountVectorizer that checks the lowercase forms of words for being stopwords because we were instructed to not perform casefolding yet in that step.

For each step: basic(no steps after tokenization), no stopwords, casefolding, stemming, we looked at the dictionary size, token count and the most common words to learn of the effect of each of the steps.

Results:

Here are the dictionary sizes and token counts for each step, along with the cumulative change from the basic step in our dataset and the relative change from the previous step in our dataset and in the Reuters dataset.

	Dictionary Size	Cumu.% Change	Rel, % Change	Rel .% Change in Reuters	Token Count	Cumu. % Change	Rel. % Change	Rel. % Change in Reuters
Basic	37,687	-	-	-	455,108	-	-	-
No Stopwords	37,135	-1.5%	-1.5%	-0.04%	287,365	-36.9%	-36.9%	-47.3%
Casefolding	32,314	-14.3%	-14.6%	-17.4%	287,365	-36.9%	-0%	-0%
Stemming	25,683	-32.9%	-21.5%	-17.8%	287,365	-36.9%	-0%	-0%

Note: Reuters performed the steps in a different order and performed a first initial step of removing numbers, which can affect the cumulative results of a step, but as we can see, the relative change remained relatively similar.

As expected we can see that that:

- Stopwords removal barely affects the dictionary size but masssively reduced the token count.
- Casefolding does not affect the token count at all but does reduce the dictionary size by a significant amount.
- Stemming does not affect the token count at all as well, and also reduced the dictionary size by a significant amount.

All of these behaviors can be seen in both our dataset and the Reuters dataset, the biggest relative difference between the two datasets is that stopwords removal removed a bigger % of words from the dictionary, but because stopwords removal removes a specific amount of words, it makes sense that our dataset was affected more since the Reuters dictionary has significantly more terms.

Next we will look at the 20 most common words in each step to observe the changes ourselves:
Basic: We can see that almost all of the words in the top 20 list are useless for language modeling(the only apparent exceptions are “political” and “bias”)

Top 20 words:

	Term	Count
0	the	22331
1	of	15784
2	and	11459
3	in	9046
4	to	8648
5	that	4738
6	is	4327
7	for	3963
8	bias	3155
9	on	3136
10	are	2891
11	The	2839
12	as	2661
13	political	2484
14	with	2483
15	by	2258
16	or	2001
17	from	1917
18	be	1778
19	this	1747

No Stopwords: We can see a very significant improvement over the previous step, most of the words in the list could be helpful for langauge modeling, we can also now see that removal of short words(<2 letters) and numbers might help if neccesary.

Top 20 words:

	Term	Count
0	bias	3155
1	political	2484
2	results	888
3	Table	842
4	media	792
5	data	790
6	10	739
7	search	683
8	news	669
9	Political	669
10	social	622
11	model	597
12	al	561
13	conservative	511
14	state	499
15	users	497
16	content	490
17	Journal	485
18	et	480
19	information	474

Case Folding: The top 20 list remains largely unchanged, mainly a few words changing positions, this makes sense as there are not a lot of capitilized words in most texts.

Top 20 words:

	Term	Count
0	bias	3761
1	political	3229
2	media	1080
3	news	1003
4	results	987
5	social	973
6	table	929
7	data	884
8	model	823
9	party	807
10	10	739
11	search	738
12	conservative	693
13	state	668
14	information	605
15	al	603
16	professors	591
17	public	586
18	high	576
19	content	570

Stemming: with the final step there are some changes that have pushed a few remaining useless terms out and added a few more useful terms instead.

```
Top 20 words:
        Term  Coun
0      polit  3869
1        bia  3767
2     result  1224
3      model  1153
4     differ  1150
5      media  1080
6        use  1036
7       news  1003
8     social   999
9       tabl   987
10     parti   984
11     state   935
12     studi   897
13      data   884
14   conserv   868
15   student   863
16    search   805
17      user   774
18      bias   749
19    inform   745
```

Question 2:

1,2. preprocessed like we explained in the intro, labeled political bias docs as “poli” and news fairness docs as “news” and saved the dataframe to a parquet file for the actual analysis.

```
   label                                               text
0   poli   Full Terms & Conditions of access and use can...
1   poli   AMERICANS’ VIEWS OF POLITI CAL BIAS IN THE AC...
2   poli   Univ ersity of Chicago Law School Univ ersity...
3   poli   Analyzing Political Bias and Unfairness in Ne...
4   poli   A Question of Balance — 1Running head: A QUES...
..   ...                                                ...
93  news  doi:10.1111/j.1662-6370.2011.02015.x\n\nThe Fa...
94  news  University of Pennsylvania\n\nScholarlyCommons...
95  news  8_AMMORI_COMPLETE\n\n12/3/2008 2:47 PM\n\nTHE ...
96  news  Toward Fairness in Misinformation Detection Al...
97  news  Two-Sided Fairness in Non-Personalised Recomme...

[98 rows x 2 columns]
```

3. All the tokenization and stopwords removal is done inside TfidfVectorizer

4. a static seed was set in numpy to maintain consistent results.

Our sklearn pipelines consisted of:

1. TfidfVectorizer set to the “english” stopwords removal of sklearn
2. One of the classifiers: MultinomialNB, SGD, Kneighbors, RandomForest

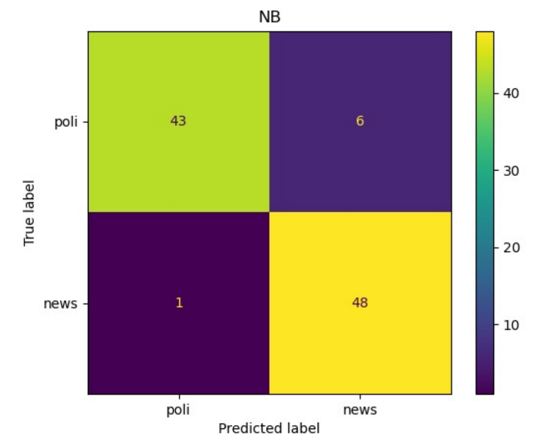
We performed 10-fold validation(using sklearn.model_selection.Kfold with shuffle) on each of the classifiers

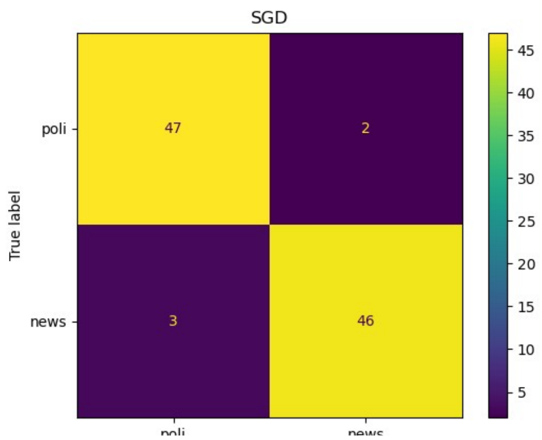
5.
Results:

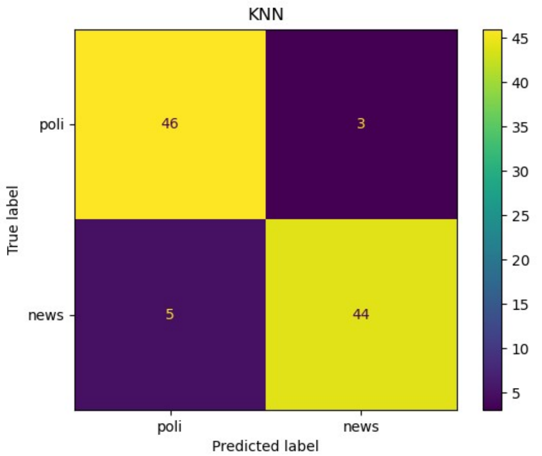
the average correct classification rate for each classifier based on all the test:
MultinomialNB: 92.89%, SGD: 94.67%, Kneighbours: 91.89%, RandomForest: 88%.(the values before each % are the individual results from each of the 10 folds per classifier)

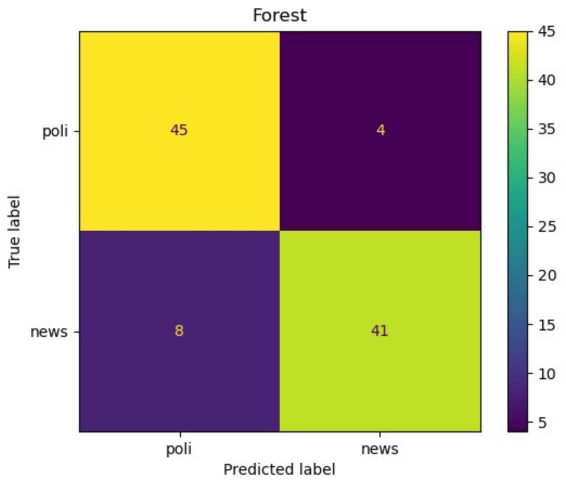
```
nb:
 [1.0, 0.7, 1.0, 0.8, 0.9, 1.0, 1.0, 1.0, 0.8888888888888888, 1.0] 92.89%
sgd:
 [1.0, 1.0, 1.0, 0.9, 1.0, 1.0, 1.0, 0.9, 0.7777777777777778, 0.8888888888888888] 94.67%
knn:
 [0.9, 1.0, 0.9, 0.9, 0.9, 1.0, 1.0, 0.7, 0.8888888888888888, 1.0] 91.89%
forest:
 [0.9, 0.7, 0.8, 0.9, 0.7, 0.9, 1.0, 0.9, 1.0, 1.0] 88.00%
```

We generated a confusion matrix for each classifier based on all the tests for that classifier:









And we calculated the scoring statistics for each classifier and label:

	Political Bias			News Fairness		
	Precision	Recall	F-Score	Precision	Recall	F-Score
MultinomialNB	0.977	0.878	0.925	0.889	0.98	0.932
SGD	0.94	0.959	0.949	0.958	0.939	0.948
KNN	0.902	0.939	0.92	0.936	0.898	0.917
Random Forest:	0.849	0.918	0.882	0.911	0.837	0.872

Based on these results, SGD seems to be the best classifier, following by MultinomialNB, KNN and finally Random Forest.

Misclassifications:

Looking at the misclassified documents, we can observe multiple things:
1. for the applicable classifiers(all except SGD), looking at the probabilities generated, the misclassified docs are almost always within 10% of 50%, which means the model is not very certain about them, and usually it is more certain about the correct classifications
example for probabilities from forest to demonstrate:

```
correct: [[0.3  0.7 ]
 [0.29 0.71]
 [0.4  0.6 ]
 [0.57 0.43]
 [0.69 0.31]
 [0.72 0.28]
 [0.62 0.38]
 [0.6  0.4 ]
 [0.69 0.31]]
misclassified: [[0.45 0.55]]
correct: [[0.43 0.57]
 [0.28 0.72]
 [0.46 0.54]
 [0.6  0.4 ]
 [0.6  0.4 ]
 [0.79 0.21]
 [0.63 0.37]]
misclassified: [[0.51 0.49]
 [0.48 0.52]
 [0.49 0.51]]
correct: [[0.3  0.7 ]
 [0.26 0.74]
 [0.22 0.78]
 [0.81 0.19]
 [0.75 0.25]
 [0.58 0.42]
 [0.65 0.35]
 [0.58 0.42]]
misclassified: [[0.46 0.54]
 [0.44 0.56]]
correct: [[0.4  0.6 ]
 [0.28 0.72]
 [0.36 0.64]
 [0.32 0.68]
 [0.33 0.67]
 [0.37 0.63]
 [0.44 0.56]
 [0.81 0.19]
 [0.7  0.3 ]]
misclassified: [[0.54 0.46]]
correct: [[0.4  0.6 ]
 [0.37 0.63]
 [0.35 0.65]
 [0.22 0.78]
 [0.71 0.29]
 [0.72 0.28]
 [0.71 0.29]]
misclassified: [[0.49 0.51]]
```

2. Looking at the actual text inside the misclassified docs, we can see that political bias docs often also talk about news and not just political bias or the other way around, news fairness docs that also talk about politics, so it is hard to put them in a single class(and in general news and politics tend to be relatively similar subjects), this is a flaw of using a one-of classifier.
And looking at the highest scoring words in the docs(TF-IDF), we can see “news”, “fairness” and related words a lot more in News Fairness related docs than in Political Bias docs(example, in one of the Naive Bayes classifiers, “news” and “fairness” are the 1^{st} and 2^{nd} most common terms among the correctly classified News Fairness docs, and in correctly classified Political Bias docs, these terms are not among the top 10 terms, but “political” is 6^{th} and “bias” is 11^{th} (which do not appear among the most common terms in News Fairness docs), behind a few social media related terms and a few non political/news terms.
On the other hand, in misclassified docs we can see the opposite behavior, documents that barely contain words common to their label and/or containing words common to the opposite label:
We can almost always see “news” among the top words in misclassified political bias docs, or “news” or other news related terms barely appear in news fairness docs that were misclassified.
Examples:
quotes from political bias docs that were misclassified:
”A Question of Balance — 1Running head: A QUESTION OF BALANCE A Question of Balance:Are Google News search results politically biased...This study examines search results from the popular online news portal Google News inan effort to determine whether they are politically biased”, and looking at the histogram for the entire doc, “news” is the most common word.
“Political Bias and Factualness in News Sharing across more than 100,000 Online Communitie”, and looking at the histogram, “news” is the 5^{th} most common word, behind 4 non-political words(reddit, links, communities, content)
“Many people view news on social media, yet the production of news items online has come under fire because of the common spreading of misinforma- tion”

quotes from news fairness docs that were misclassified:
“MEDIA BIAS, POLITICAL POLARIZATION,\nAND THE MERITS OF FAIRNES”

“This article asks how the press communicates political issues to citizens during referendum campaign”
the doc for the article titled “Fairness to Rightness: Jurisdiction, Legality, and the Legitimacy of International Criminal Law” hardly contains the word “news” and other news related terms, the term “news” is the 40,413rd most common term.

3. one of the misclassified docs is invalid(essentially an empty doc with some noise):
a misclassified political bias post(in its entirety):
“A c c e l e r a t i n g t h e w o r l d ' s r e s e a r c h . Press Bias and Politics: How the Media Frame Controversial Issues J i m A . K u y p e r s C i t e t h i s p a p e r G e t t h e c i t a t i o n i n M L A , A P A , o r C h i c a g o s t y l e sD o w n l o a d e d f r o m \xa0 A c a d e m i a . e d u \xa0 \uf08e R e l a t e d p a p e r s D o w n l o a d a P D F P a c k o f t h e b e s t r e l a t e d p a p e r s \xa0 \uf08”

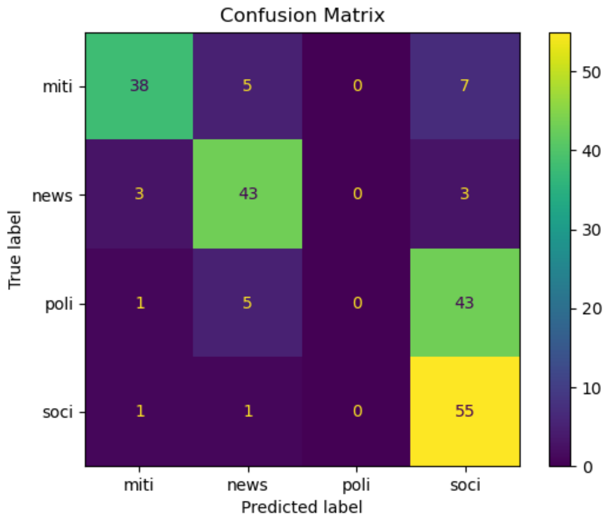
Question 3:

The additional docs were preprocessed in the same way as in question 2 with the exception that to prevent reading errors the files were read with ISO-8859-1 encoding and the added categories were labeled “soci” and “miti” for Social Bias and Bias Mitigation respectively.
And like in question 2, tokenization and stopwords removal was done inside TfidfVectorizer.
We have attempted to use PCA to reduce the very high dimension count of the input vectors but did not see a significant improvent.
Since we were not instructed to use K-Fold or a test set, we fitted K-means on the entire docs set.
The K-means algorithm was seeded to keep the starting centroids consistent.
The sklearn pipeline consisted of

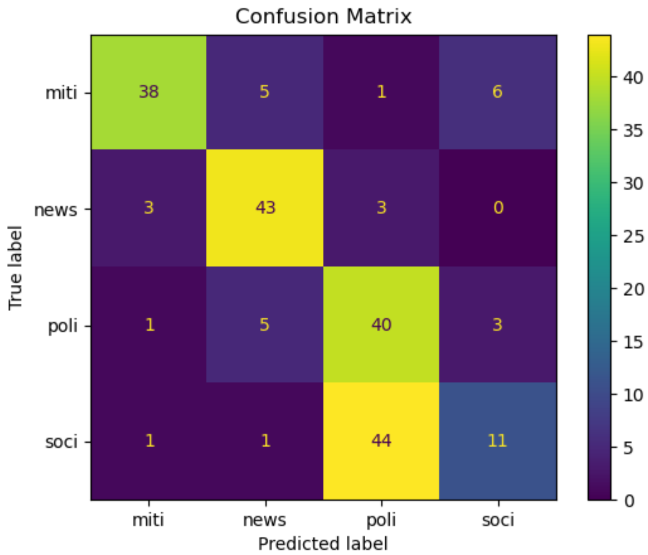
1. TfidfVectorizer set to the “english” stopwords removal of sklearn
2. Kmeans classifier

Results:

The **rand score** of the set is 0.777, meaning ~77.7% of the pairs were clustered together/apart correctly.
Based on the most common label in each cluster, we will choose that label for that cluster.
We ended up with 2 clusters with the same label, so we will show 2 options for confusion matrices:
Strictly based on most common label:



And by choosing the 2nd most common(by a small margin, a bit like ranked-choice voting) for cluster 3:



Neither options look like good results and in the next section we will explore and explain why it happend

Misclassifications:
we will look at each cluster and specifically at the docs with a different label than the clsuter and try to explain why they were put in that cluster.
We will also look at the most common words in each cluster and in specific labels or docs within the cluster.
We wrote a function that shows the 20 highest scoring word of a given set of docs on average.
Top 20 words in each cluster:

```
>>> common_in_docs(clusters.get_group(0))
['fairness' 'news' 'doctrine' 'user' 'ranking' 'media' 'bias' 'users'
 'fcc' 'search' 'public' 'recommendation' 'broadcasters' 'broadcast'
 'fair' 'information' 'journalism' 'political' '2018' 'items']
>>> common_in_docs(clusters.get_group(1))
['bias' 'gender' 'et' 'al' 'word' 'language' '2019' 'sentence' '2018'
 'models' '2020' 'linguistics' 'model' 'embeddings' 'biases' 'dataset'
 'computational' 'political' 'nlp' 'words']
>>> common_in_docs(clusters.get_group(2))
['data' 'ai' 'bias' 'fairness' 'learning' 'classiï' 'model' 'dataset'
 'machine' 'training' '2018' 'models' 'mitigation' 'gender' 'cation' 'ml'
 'accuracy' 'protected' 'algorithms' '2019']
>>> common_in_docs(clusters.get_group(3))
['bias' 'political' 'social' 'desirability' 'research' 'media'
 'participants' 'respondents' 'percent' 'self' 'sdb' 'party' 'professors'
 'study' 'journal' 'data' 'children' 'students' 'al' 'et']
```

Cluster 0:
The most common label is **News Fairness**(5 Bias Mitigation, 43 News Fairness, 5 Political Bias, 1 Social Bias), which matches the most common terms being “fairness” and “news”
The misclassified docs tend to also talk about news:
quotes from misclassified Political Bias docs:
“This study examines search results from the popular online news portal Google News inan effort to determine whether they are politically biased”
“Many people view news on social media, yet the production of news items online has come under fire because of the common spreading of misinforma- tion”
“Political Bias and Factualness in News Sharing across more than 100,000 Online Communities” and more
We have already expanded on the common words in these articles as they were also misclassified in question 2.
misclassified Social Bias doc: only one Social Bias doc appears in this cluster and it’s 2nd most common term is “fairness”.
misclassified Bias Mitigation docs: while based on the texts it is less visible than the other misclassified docs why they were misclassified, we can see they are further away from the centeroid and are closer to being in another cluster than News Fairness docs in the cluster
As an example, here are the distances of 3 of the News Fairness docs to each centroid, followed by the distances of the 5 Bias Mitigation docs(the left most distance is to this cluster):

```
(Pdb) (classifier.transform(clusters.get_group(0)["text"].iloc[-9:-6]))
array([[0.89872422, 1.08198251, 1.04498023, 1.00257898],
       [0.90099788, 0.95733098, 0.93843433, 0.95454453],
       [0.92051662, 1.0170795 , 0.99984209, 0.98802525]])
(Pdb) (classifier.transform(clusters.get_group(0)["text"].iloc[-5:]))
array([[0.95608185, 0.99208468, 0.98749433, 0.97302707],
       [0.99170405, 1.03471012, 1.01225135, 0.99926655],
       [0.96942909, 1.02201176, 0.99581793, 0.9745787 ],
       [0.96623284, 1.03716483, 0.9879572 , 0.99820008],
       [0.98206994, 1.04048718, 1.00774953, 1.01141751]])
```

Cluster 1:
The most common label is **Social Bias** (6 Bias Mitigation, 3 Political Bias, 11 Social Bias, this appears to be a small cluster relative to the others)
there are no misclassified News Fairness docs in this cluster, and some misclassfieid Political Bias and Bias Mitigation docs in the cluster, this makes some sense as all 3 of the labels in the cluster are about bias in some way and News Fairness is not(or at least less so).
Unlike in cluster 0, the misclassified docs are generally not further away from the cluster than the Social Bias docs:

```
(Pdb) clusters.get_group(1)
    label                                            text
3    poli   Analyzing Political Bias and Unfairness in Ne...
41   poli   Studying Political Bias via Word Embeddings J...
48   poli   We Can Detect Your Bias: Predicting the Polit...
103  soci  Social Bias in Elicited Natural Language Infer...
106  soci  1911.00461v1 [cs.CL] 1 Nov 2019\n\narXiv\n\n \...
109  soci  1903.10561v1 [cs.CL] 25 Mar 2019\n\narXiv\n\nO...
110  soci  2005.00813v1 [cs.CL] 2 May 2020\n\narXiv\n\nSo...
112  soci   \n\nTowards Understanding and Mitigating Soci...
122  soci  2210.04337v1 [cs.CL] 9 Oct 2022\n\narXiv\n\n \...
127  soci  The Thirty-Sixth AAAI Conference on Artificial...
131  soci  Exploring Social Bias in Chatbots using Stereo...
135  soci  1911.03891v3 [cs.CL] 23 Apr 2020\n\narXiv\n\nS...
147  soci                                              ...
150  soci  Counterfactually Measuring and Eliminating Soc...
165  miti  Bias Mitigation for Toxicity Detection via Seq...
170  miti  Mitigating Political Bias in Language Models T...
182  miti  Anatomizing Bias in Facial Analysis\nRicha Sin...
194  miti  The Thirty-Fourth AAAI Conference on Artiﬁci...
196  miti  Mitigating Gender Bias in Natural Language Pro...
202  miti  REVISE: A Tool for Measuring and Mitigating Bi...
(Pdb) (classifier.transform(clusters.get_group(1)["text"]))
array([[0.95149213, 0.85938738, 0.95373816, 0.94370383],
       [0.98826407, 0.87117047, 0.98462223, 0.95393143],
       [0.93628313, 0.87914466, 0.95341813, 0.95320314],
       [1.01832878, 0.92162739, 1.01693757, 0.98548636],
       [1.00292777, 0.90233954, 0.98325658, 0.98898816],
       [1.01433977, 0.88214211, 1.00406485, 0.98417631],
       [1.00465376, 0.89948476, 1.00048047, 0.98677316],
       [0.98821925, 0.83053994, 0.96223642, 0.97434072],
       [1.00693363, 0.88238925, 0.99261671, 0.98801269],
       [1.02423512, 0.90024881, 1.02078802, 0.99750407],
       [1.01734773, 0.92772362, 1.00499431, 0.98322929],
       [0.99789499, 0.90612211, 0.9910382 , 0.96794438],
       [1.00691095, 0.84042623, 0.97928002, 0.9791349 ],
       [1.01185864, 0.90988244, 0.98353571, 0.98308803],
       [0.99807296, 0.91289625, 0.97155655, 0.97871522],
       [0.98593757, 0.85220979, 0.94301592, 0.96724198],
       [0.97216312, 0.87103701, 0.89497918, 0.97030906],
       [1.02033581, 0.89318951, 1.0089752 , 0.99422144],
       [0.98721944, 0.78634991, 0.90240012, 0.96593415],
       [0.98937768, 0.91475323, 0.95049283, 0.98367405]])
```

Additionally, looking at the most common terms in each label in this cluster, “bias” is the most common term among them, but it is also very common in clusters 2 and 3 so it doesn’t fully explain the clustering, but the terms “word” and “gender” are common in all 3 labels and are not common in the other clusters, which does explain it.
Top 20 words in each label:

```
>>> words[soci_vec.argsort()[::-1][:20]]
array(['bias', 'language', 'sentence', 'et', 'word', 'al', '2019',
       'linguistics', 'gender', 'tokens', 'models', 'vlp',
       'computational', 'person', 'social', 'biases', '2018',
       'stereotype', 'templates', 'association'], dtype=object)
>>> words[poli_vec.argsort()[::-1][:20]]
array(['bias', 'political', 'media', 'articles', 'word', 'news',
       'classiﬁ', 'et', 'al', 'article', 'words', 'level', 'republican',
       'ideology', 'corpus', 'tweets', 'granularity', 'axis', 'baly',
       'gender'], dtype=object)
>>> words[miti_vec.argsort()[::-1][:20]]
array(['bias', 'gender', 'et', 'al', 'embeddings', 'word', 'toxicity',
       'debias', 'glov', '2019', 'object', 'elmo', '2018', 'debiasing',
       '2020', 'images', 'dataset', 'recognition', 'data', 'model'],
      dtype=object)
```

Cluster 2:
The most common label **Bias Mitigation**(38 Bias Mitigation, 3 News Fairness, 1 Political Bias, 1 Social Bias), but in reality, it appears that the cluster is actually about **AI and machine learning**. Looking at the distances, we can see somewhat more decisive distances for Mitigation Bias docs than other labels(1st is Political Bias, then 3 News Fairness docs and 1 Social Bias doc, followed by a few Bias Mitigation Docs, this cluster is the 3rd column from the left):

```
(Pdb) (classifier.transform(clusters.get_group(2)["text"]))
array([[0.99440033, 0.96853767, 0.92946386, 0.96985339],
       [0.95665813, 0.92484955, 0.91055396, 0.97945087],
       [0.93413032, 0.91897119, 0.87635378, 0.97076617],
       [0.91127197, 0.89954352, 0.81983435, 0.96310996],
       [1.00583657, 1.01526633, 0.91643949, 0.97796257],
       [0.99305061, 1.02442042, 0.9012208 , 0.98428018],
       [1.00008413, 1.01637107, 0.94043798, 0.98520496],
       [0.97509222, 0.9922331 , 0.896265  , 0.99012818],
       [0.99436338, 0.94675903, 0.92002651, 0.98479608],
       [0.98292073, 0.93005868, 0.86533161, 0.96199751],
       [1.00202472, 1.0025916 , 0.90095245, 0.98166728],
```

But looking at the common words in the cluster(seen earlier) we can see that the docs in this cluster tend to be more related to AI and machine learning than other clusters, and this holds for each label specifically
Top 20 words in each label:

```
>>> common_in_docs(clusters.get_group(2)[clusters.get_group(2)["label"]=="miti"])
['ai' 'data' 'bias' 'fairness' 'classiï' 'learning' 'model' 'dataset'
 'machine' 'training' '2018' 'mitigation' 'cation' 'gender' 'protected'
 'models' 'algorithms' 'ml' 'systems' 'accuracy']
>>> common_in_docs(clusters.get_group(2)[clusters.get_group(2)["label"]=="news"])
['dbias' 'fairness' 'bias' 'detection' 'module' 'news' 'pipeline'
 'sentiment' 'biased' 'ml' 'data' 'model' 'biases' 'accuracy'
 'recognition' 'models' 'bert' 'distilbert' 'tfidf' 'words']
>>> common_in_docs(clusters.get_group(2)[clusters.get_group(2)["label"]=="poli"])
['ibc' 'f1' 'data' 'oti' 'auc' 'model' 'lstm' 'dropout' 'network'
 'political' 'set' 'directional' 'hidden' 'score' 'rnn' 'text' 'recursive'
 'layer' 'url' 'neural']
>>> common_in_docs(clusters.get_group(2)[clusters.get_group(2)["label"]=="soci"])
['ai' 'patients' 'bias' 'clinician' 'data' 'clinicians' 'health'
 'framingham' 'risk' 'pennsylvania' 'jama' 'parikh' 'perelman'
 'philadelphia' 'care' 'algorithm' 'clinical' 'medicine' 'complacency'
 'predictions']
```

And we can also see that by looking at quotes from the docs:
Political Bias doc: “ An algorithmic approach towards detection of such bias is both intellectu- ally challenging and useful in areas like election prediction”
News Fairness doc: “'Balancing Fairness and Accuracy in Sentiment\nDetection using Multiple Black Box Models”
Social Bias doc: “Addressing Bias in Artificial Intelligence in Health Care”
Bias Mitigation doc: “Mitigating Bias in Deep Nets with Knowledge Bases : the Case of Natural\nLanguage Understanding for Robots**”**

Cluster 3: the most common label was **Social Bias**, but not by much, with **Political Bias** a little behind(1 Mitigation Bias, 3 News Fairness, 40 Political bias, 44 Social Bias), but in reality we will see that this label too is not accurate to the contents of the cluster.
Looking at the common words, we can see that the commonality between the the Political Bias and Social Bias docs is “bias”, which is expected, but there is also little mention of AI and machine learning, and since Bias Mitigation docs seem to mostly talk about AI and machine learning, this cluster appears to be a “Bias docs that are not AI and machine learning docs” cluster
Top 20 words in each label:

```
>>> common_in_docs(clusters.get_group(3)[clusters.get_group(3)["label"]=="miti"])
['workers' 'crowdwork' 'worker' 'workerâ' 'tasks' 'perspectives' 'phase'
 'interactions' 'task' 'images' 'biases' 'experiment' 'classiï' 'style'
 'different' 'difï' 'political' 'cation' 'culty' 'styles']
>>> common_in_docs(clusters.get_group(3)[clusters.get_group(3)["label"]=="news"])
['iiasa' 'law' 'crimes' 'icl' 'swiss' 'g3' 'g1' 'interim' 'wiley'
 'onlinelibrary' 'criminal' 'coverage' 'political' 'tribunals' 'nuremberg'
 'referendum' 'evolutionary' 'media' 'legality' 'proposer']
>>> common_in_docs(clusters.get_group(3)[clusters.get_group(3)["label"]=="poli"])
['political' 'bias' 'media' 'professors' 'party' 'percent' 'students'
 'news' 'liberal' 'conservative' 'ideological' 'politicians' 'politics'
 'judicial' 'newspapers' 'social' 'press' 'war' 'nominees' 'right']
>>> common_in_docs(clusters.get_group(3)[clusters.get_group(3)["label"]=="soci"])
['social' 'bias' 'desirability' 'research' 'sdb' 'self' 'children'
 'respondents' 'participants' 'study' 'et' 'group' 'peer' 'al' 'journal'
 'behavior' 'socially' 'effects' 'review' 'studies']
```

The single Bias Mitigation doc talks about bias in crowdsourcing, unlike most other Bias Mitigation docs, which tend to talk about AI and machine learning, which explains why it is not in the cluster with most other Bias Mitigation docs(which ended up being the “AI” cluster).

Summary:
In this project we have performed 3 different Information Retrieval tasks and analyzed their results. We have seen that real-life models do not always perform idealy for various different reasons.
1. In the language modeling task we saw the massive improvement a few simple cleaning steps can do to a language model, stopwords removal turned out to be by far the most significant improvement with stemming behind it, with casefolding making a relatively small improvement.
2. In the text classification task, we have seen that some documents might appear to fit in the other class because they use less words common to its class and more words common to the other class, and some documents can potentially fit in both classes, which is problematic in a one-of classification like we have done here.
3. In the text clustering task, we have seen that despite the clusters we intended to create, it is possible there are a different set of “labels” that better separate the given documents, in our case, the K-means algorithm appears to have clustered the document to “News Fairness”,”Social”,”Bias in AI”,”Bias unrelated to AI” rather than the “News Fairness”, “Social Bias”, “Political Bias”, “Bias Mitigation” labels we have used and expected the clusters to represent.