Search Engine Bias

IR PROJECT

Tools:

We used Python and Jupyter Notebook, primarily using the NLTK and Scikit-learn libraries.

Language modeling

We used a 1-gram (unigram) language model to evaluate the impact of each pre-processing action we performed on our data.

- Our base language model :

```
Number of words in our corpus: 444563
Number of unique words in our corpus : 28143

[('the', {'tf': 20038, 'prob': 0.04507347665010358}),
 ('of', {'tf': 13165, 'prob': 0.02961335063871712}),
 ('and', {'tf': 10333, 'prob': 0.02324304991643479}),
 ('to', {'tf': 8078, 'prob': 0.018170652978318033}),
 ('in', {'tf': 6811, 'prob': 0.015320663213087908}),
 ('a', {'tf': 6243, 'prob': 0.014043004028675352}),
 ('search', {'tf': 5566, 'prob': 0.012520160247254044}),
 ('that', {'tf': 4441, 'prob': 0.00998958527812706}),
 ('is', {'tf': 4228, 'prob': 0.00951046308397235}),
 ('for', {'tf': 3807, 'prob': 0.008563465695525718}),
 ('on', {'tf': 3454, 'prob': 0.007769427505212984}),
 ('0', {'tf': 2966, 'prob': 0.006671720318605012}),
 ('we', {'tf': 2351, 'prob': 0.005288393393354822604}),
 ('results', {'tf': 2342, 'prob': 0.005268094735729244}),
 ('’', {'tf': 2322, 'prob': 0.0052231067362780974}),
 ('are', {'tf': 2305, 'prob': 0.005184866936744623 5}),
 ('by', {'tf': 2285, 'prob': 0.005139878937293477}),
 ('The', {'tf': 2210, 'prob': 0.004971173939351678}),
 ('as', {'tf': 2188, 'prob': 0.004921687139955417}),
 (']', {'tf': 2114, 'prob': 0.004755231541986175}),
 ('[', {'tf': 2113, 'prob': 0.004752982142013618}),
 ('1', {'tf': 2103, 'prob': 0.004730488142288045}),
 ('Google', {'tf': 1870, 'prob': 0.004206377948682189}),
 ('engines', {'tf': 1867, 'prob': 0.004199629748764517}),
 ('with', {'tf': 1851, 'prob': 0.0041636393492036}),
 ('from', {'tf': 1746, 'prob': 0.003927452352085081}),
 ('engine', {'tf': 1691, 'prob': 0.003803735353594429}),
 ('information', {'tf': 1643, 'prob': 0.0036957641549116775}),
 ('this', {'tf': 1594, 'prob': 0.0035855435562563687}),
 ('be', {'tf': 1576, 'prob': 0.003545054356750337}),
```

The language model contains many stop words, numbers, and punctuation marks. Additionally, we observed that some of our query terms, such as 'search,' 'engine,' and 'engines,' are included in the list of common words created by the base language model.

- Cleaning data from numbers, punctuation marks and non-English words:

```
Number of words in our corpus: 308349
Number of unique words in our corpus : 10315

[('the', {'tf': 20038, 'prob': 0.06498480617741585}),
 ('of', {'tf': 13165, 'prob': 0.042695127923229846}),
 ('and', {'tf': 10333, 'prob': 0.03351072972508424}),
 ('to', {'tf': 8078, 'prob': 0.026197587798241603}),
 ('in', {'tf': 6811, 'prob': 0.022088607389678577}),
 ('a', {'tf': 6243, 'prob': 0.02024653882451378}),
 ('search', {'tf': 5566, 'prob': 0.01805097470723109}),
 ('that', {'tf': 4441, 'prob': 0.014402511439959267}),
 ('is', {'tf': 4228, 'prob': 0.013711735728022468}),
 ('for', {'tf': 3807, 'prob': 0.012346399696447857}),
 ('on', {'tf': 3454, 'prob': 0.011201593000139452}),
 ('we', {'tf': 2351, 'prob': 0.007624477458983165}),
 ('are', {'tf': 2305, 'prob': 0.007475295849832495}),
 ('by', {'tf': 2285, 'prob': 0.007410434280636552}),
 ('The', {'tf': 2210, 'prob': 0.007167203396151763}),
 ('as', {'tf': 2188, 'prob': 0.00709585567003625}),
 ('with', {'tf': 1851, 'prob': 0.006002938229084576}),
 ('from', {'tf': 1746, 'prob': 0.005662414990805872}),
 ('engine', {'tf': 1691, 'prob': 0.005484045675517028}),
 ('information', {'tf': 1643, 'prob': 0.005328377909446763}),
 ('this', {'tf': 1594, 'prob': 0.005169467064916702}),
 ('be', {'tf': 1576, 'prob': 0.005111091652640352}),
 ('not', {'tf': 1523, 'prob': 0.004939208494271102}),
 ('s', {'tf': 1503, 'prob': 0.004874346925075159}),
 ('In', {'tf': 1440, 'prob': 0.004670032982107936}),
 ('or', {'tf': 1406, 'prob': 0.004559768314474832}),
```

After keeping only English words in our model, the total number of words has significantly decreased. This outcome was expected, as we observed in the previous table that the unwanted tokens were quite frequent in our documents.

- Removing stop words:

```
Number of words in our corpus: 118800
Number of unique words in our corpus : 9345

[('search', {'tf': 5566, 'prob': 0.04685185185185185}),
 ('engine', {'tf': 1691, 'prob': 0.014234006734006734}),
 ('Web', {'tf': 1342, 'prob': 0.011296296296296296}),
 ('bias', {'tf': 1342, 'prob': 0.011296296296296296}),
 ('Search', {'tf': 1091, 'prob': 0.009183501683501684}),
 ('query', {'tf': 789, 'prob': 0.006641414141414142}),
 ('content', {'tf': 700, 'prob': 0.00589225589225589}),
 ('result', {'tf': 650, 'prob': 0.005471380471380472}),
 ('user', {'tf': 605, 'prob': 0.005092592592592593}),
 ('al', {'tf': 601, 'prob': 0.005058922558922559}),
 ('data', {'tf': 571, 'prob': 0.004806397306397306}),
 ('study', {'tf': 549, 'prob': 0.004621212121212215}),
 ('model', {'tf': 507, 'prob': 0.004267676767676767}),
 ('number', {'tf': 490, 'prob': 0.004124579124579124}),
 ('web', {'tf': 483, 'prob': 0.004065656565656566}),
 ('Figure', {'tf': 479, 'prob': 0.004031986531986532}),
 ('set', {'tf': 461, 'prob': 0.003880471380471380}),
 ('based', {'tf': 421, 'prob': 0.003543771043771044}),
 ('document', {'tf': 413, 'prob': 0.003476430976430976}),
 ('political', {'tf': 409, 'prob': 0.003442760942760942}),
 ('time', {'tf': 406, 'prob': 0.003417508417508417}),
 ('position', {'tf': 396, 'prob': 0.0033333333333333335}),
```

After removing stop words, we noticed a significant decrease in the total word count, which was expected since stop words are very common in documents. However, the number of unique words only decreased slightly. This is because although stop words appear frequently in documents, there are not many stop words in the English language.

Note: when removing stop words, we took care of capitalized/lowercase stop words, as this step occurred before the case folding step.

- Case folding tokens:

```
Number of words in our corpus: 118800
Number of unique words in our corpus : 7071

[('search', {'tf': 6762, 'prob': 0.0569191919191192}),
 ('engine', {'tf': 2034, 'prob': 0.01712121212121212}),
 ('web', {'tf': 1834, 'prob': 0.015437710437710438}),
 ('bias', {'tf': 1606, 'prob': 0.013518518518518518}),
 ('query', {'tf': 823, 'prob': 0.006927609427609427}),
 ('content', {'tf': 754, 'prob': 0.006346801346801347}),
 ('data', {'tf': 684, 'prob': 0.005757575757575757}),
 ('result', {'tf': 671, 'prob': 0.005648148148148148}),
 ('user', {'tf': 645, 'prob': 0.005429292929292296}),
 ('al', {'tf': 631, 'prob': 0.005311447811447811}),
 ('study', {'tf': 610, 'prob': 0.005134680134680135}),
 ('model', {'tf': 568, 'prob': 0.004781144781144781}),
 ('number', {'tf': 532, 'prob': 0.004478114478114478}),
 ('figure', {'tf': 530, 'prob': 0.004461279461279461}),
 ('political', {'tf': 523, 'prob': 0.004402356902356902}),
 ('set', {'tf': 465, 'prob': 0.003914141414141414}),
 ('based', {'tf': 454, 'prob': 0.003821548821548217}),
 ('media', {'tf': 453, 'prob': 0.003813131313131313}),
 ('document', {'tf': 435, 'prob': 0.0036616161616161618}),
 ('time', {'tf': 431, 'prob': 0.003627946127946128}),
 ('position', {'tf': 424, 'prob': 0.003569023569023569}),
```

After performing case folding, we observed that the total word count remained unchanged because we did not remove any words from our language model. Rather, we simply modified the capitalization of some words. However, words that were capitalized in some places and lowercase in others, are now mapped to the same term, resulting in a significant decrease in the count of unique words.

- Stemming tokens via Porter Stemmer:

```
Number of words in our corpus: 118800
Number of unique words in our corpus : 5284

[('search', {'tf': 6901, 'prob': 0.058089225589225586}),
 ('engin', {'tf': 2088, 'prob': 0.017575757575757574}),
 ('web', {'tf': 1834, 'prob': 0.015437710437710438}),
 ('bia', {'tf': 1606, 'prob': 0.013518518518518518}),
 ('queri', {'tf': 838, 'prob': 0.007053872053872054}),
 ('content', {'tf': 785, 'prob': 0.0066077441077441075}),
 ('polit', {'tf': 703, 'prob': 0.005917508417508418}),
 ('data', {'tf': 684, 'prob': 0.005757575757575757}),
 ('result', {'tf': 672, 'prob': 0.0056565656565656566}),
 ('studi', {'tf': 660, 'prob': 0.005555555555555556}),
 ('user', {'tf': 645, 'prob': 0.0054292929292929296}),
 ('al', {'tf': 631, 'prob': 0.005311447811447811}),
 ('relev', {'tf': 610, 'prob': 0.005134680134680135}),
 ('rank', {'tf': 609, 'prob': 0.005126262626262626}),
 ('posit', {'tf': 602, 'prob': 0.005067340067340068}),
 ('model', {'tf': 584, 'prob': 0.004915824915824916}),
```

After performing stemming, the total number of words in the model remained unchanged since we did not remove any words. However, the number of unique words decreased because stemming groups together words that appear in different forms but have the same root.

We can see that stemming has resulted in some over-stemming, producing misspelled or meaningless words.

- Lemmatizing tokens (no stemming) :

```
Number of words in our corpus: 118800
Number of unique words in our corpus : 7011

[('search', {'tf': 6762, 'prob': 0.05691919191919192}),
 ('engine', {'tf': 2034, 'prob': 0.01712121212121212}),
 ('web', {'tf': 1834, 'prob': 0.015437710437710438}),
 ('bias', {'tf': 1606, 'prob': 0.013518518518518518}),
 ('query', {'tf': 823, 'prob': 0.006927609427609427}),
 ('content', {'tf': 784, 'prob': 0.006599326599326 6}),
 ('data', {'tf': 684, 'prob': 0.005757575757575757}),
 ('result', {'tf': 671, 'prob': 0.005648148148148148}),
 ('user', {'tf': 645, 'prob': 0.005429292929292296}),
 ('al', {'tf': 631, 'prob': 0.005311447811447811}),
 ('study', {'tf': 610, 'prob': 0.005134680134680135}),
 ('model', {'tf': 568, 'prob': 0.004781144781144781}),
 ('time', {'tf': 563, 'prob': 0.004739057239057239}),
 ('number', {'tf': 532, 'prob': 0.004478114478114478}),
 ('figure', {'tf': 530, 'prob': 0.004461279461279461}),
 ('political', {'tf': 523, 'prob': 0.004402356902356902}),
```

After performing lemmatization, the total number of words in the model remained unchanged since we did not remove any words. However, the number of unique words decreased by a smaller amount than in stemming, since lemmatization modifies the inflectional forms of words, this results in a more precise reduction in unique words that still maintain their meaning.

Summary :

	Total words count	Reduction percentage	Unique words count	Reduction percentage
No changes	444563	-	28143	-
Keeping English words only	308349	30%	10315	63%
Removing stop words	118800	61%	9345	9%
Case folding	118800	-	7071	24%
Stemming words	118800	-	5284	25%
Lemmatization (no stemming)	118800	-	7011	0.008%

- At the end we decided to keep working on tokens that were lemmatized.

Text Classification:

Preparing data:

- We performed tokenization on each document, and checked if our tokens were present in the tokenized file list, if a token was found, we counted the number of times it appeared in the tokenized file. We performed the same pre-processing steps on both the original tokens and the tokenized files. Additionally, we included a label column, 'doc_relevance', to indicate whether each document was relevant or not.

filename	ber	lucky	striven	examine	assertion	saddle	habitual	supercharged	advancement	cashier
A User Browsing Model to Predict Search Engine Click Data from Past Observations.	0	0	0	11	0	0	0	0	0	0
Algorithmic_Auditing__the_Holocaust__and_Search_Engine_Bias.	0	0	0	0	0	0	0	0	0	0
An Experimental Comparison of Click Position-Bias Models.	0	0	0	1	0	0	0	0	1	0
Auditing the Personalization and Composition of Politically-Related Search Engine Results Pages.	0	0	0	0	0	0	0	0	0	0
Auditing Web Search Results Related to the 2020 US Presidential Primary Elections Across Six Search Engines.	0	0	0	2	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
Understanding the Demographics of Twitter Users.	0	0	0	10	0	0	0	0	1	0
Unmasking-Contextual-Stereotypes_-Measuring-and-Mitigating-BERTs-Gender-Bias.	1	0	0	0	0	0	0	0	0	0
User attitudes towards news content personalization.	0	0	0	6	0	0	0	0	0	0
User Interface Personalization in News Apps.	0	0	0	0	0	0	0	0	0	0
When Personalization Is Not an Option An In-The-Wild Study on Persuasive News Recommendation.	0	0	0	1	0	0	0	0	0	0

Snippet of the data

- We have too many features, so we needed to handle some outliers and drop rare terms, to get good accuracy in classifying documents. To do so, we removed rare words in the following way: we dropped terms that appeared in less than 50% of the data. We tried setting the threshold at 75%, but we achieved better classification results with the 50% threshold, so we kept it.

filename	test	general	language	fact	popular	approach	simply	negat
A User Browsing Model to Predict Search Engine Click Data from Past Observations.	22	3	1	4	0	0	1	
Algorithmic_Auditing__the_Holocaust__and_Search_Engine_Bias.	0	0	0	0	0	0	0	
An Experimental Comparison of Click Position-Bias Models.	5	4	0	2	0	2	3	
Auditing the Personalization and Composition of Politically-Related Search Engine Results Pages.	3	10	0	0	0	1	0	
Auditing Web Search Results Related to the 2020 US Presidential Primary Elections Across Six Search Engines.	0	5	0	4	3	2	0	
...	...	...	...	...	...	...	...	
Understanding the Demographics of Twitter Users.	0	2	0	2	4	2	2	
Unmasking-Contextual-Stereotypes_-Measuring-and-Mitigating-BERTs-Gender-Bias.	9	3	54	6	0	2	0	
User attitudes towards news content personalization.	2	81	0	1	0	3	0	
User Interface Personalization in News Apps.	0	0	0	0	3	8	1	
When Personalization Is Not an Option An In-The-Wild Study on Persuasive News Recommendation.	6	6	1	21	14	6	0	

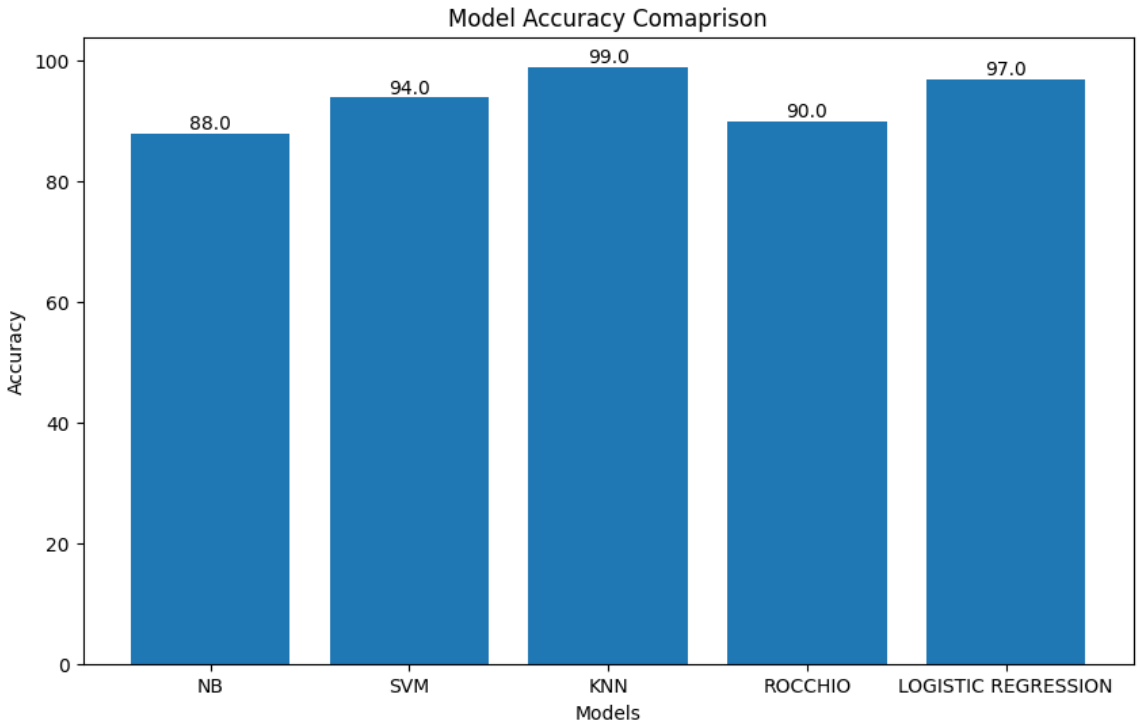
100 rows × 203 columns

Snippet of the data

Classification models:

We chose the following algorithms : Naïve Bayes, SVM, KNN, Rocchio and Logistic Regression.

The average accuracy of 10-fold cross validation among classifiers was as follows:



- Overall, our data cleaning and outlier handling contributed to achieving good classification accuracy results, with KNN being the best performing classifier.

Exploring Mislabeled Documents:

We will take a closer look at two documents, one non-relevant document that was classified as relevant, and the other is a relevant document that was classified as non-relevant.

- A false positive case:
  One false positive case occurred when 'Query-biased Learning to Rank for Real-time Twitter' was incorrectly classified as relevant. When we reviewed the document, we discovered that some of our query terms such as 'bias' and 'search' were mentioned frequently, leading to the misclassification.

- A false negative case:

  The document ‘A User Browsing Model to Predict Search Engine Click Data from Past Observations’, was incorrectly classified as non-relevant, when we reviewed the document, we observed that the query terms did not appear a lot in the document, which is why it was classified as non-relevant.

Text Clustering:

Preparing data:

We performed tokenization on the 200-document set to define our corpus, then performed tokenization on each file individually and checked if tokens were present in the corpus set, if a token was found, we counted the number of times it appeared in the tokenized file. We performed the same pre-processing steps on both the corpus set and the tokenized files. Additionally, we included a label column, 'group_name' to indicate to which directory each file belongs.

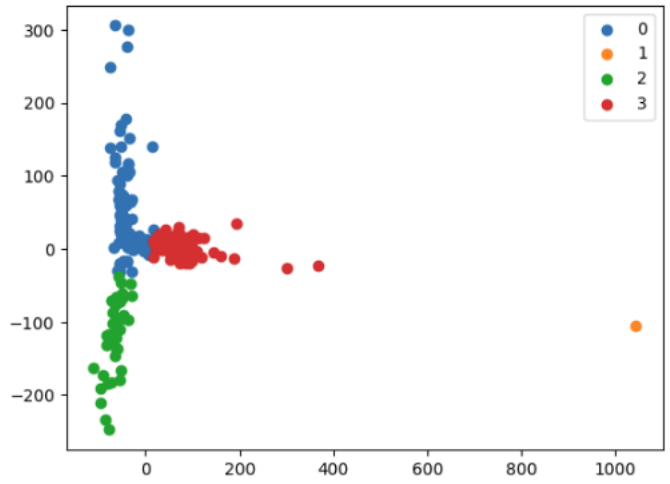
Clustering with K-means:

We used K-means clustering to cluster the data points into four clusters based on our prior knowledge of the dataset.

Evaluating Clustering:

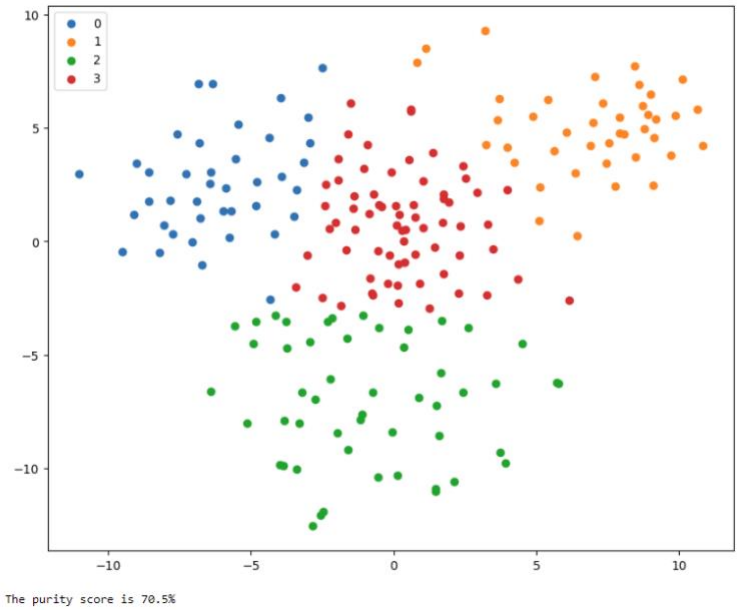
We evaluated clustering via "purity score", which is calculated by first determining the most common class (i.e., topic or theme) among the documents in each cluster, and then calculating the fraction of documents in each cluster that belong to that class. The purity score is the average of these fractions over all clusters.)

Clustering Data:



The purity score is 64.0%

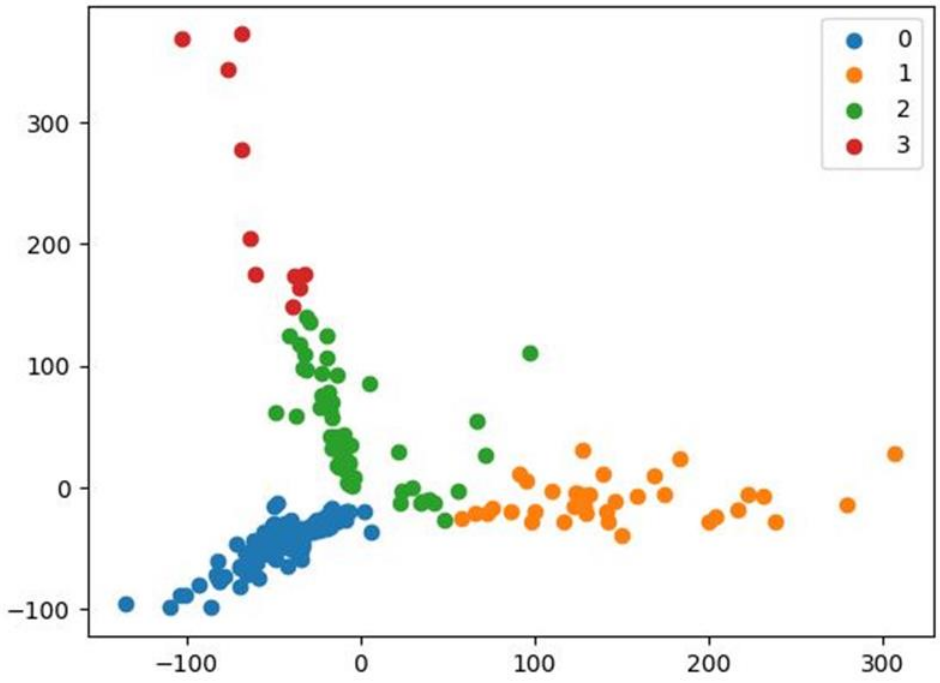
Kmeans with PCA – purity score 64%



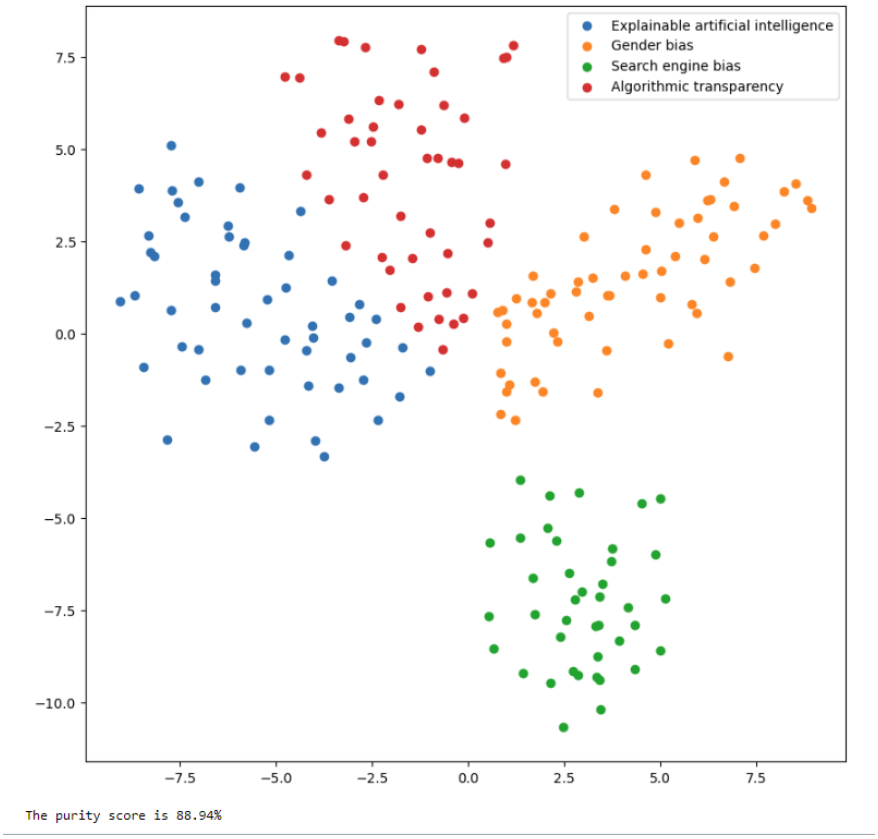
Kmeans with t-sne - purity score is 70.5%

The data was not well separated, so we decided to clean the data and handle outliers.

We removed rare terms and some outliers from the data, and got the following results:



Kmeans with PCA - purity score 67.84%



Kmeans with t-SNE - purity score 88.94%

Result analysis:

Purity score increased after doing more pre-processing steps, and in both cases using t-sne as dimensionality reduction led to better clustering.

As we can see from the plot the green cluster represents Search Engine Bias, the orange cluster represents Gender Bias, the red cluster represents Algorithmic Transparency, and the blue cluster represents Explainable Artificial Intelligence

The plot shows that the four query results are clearly separated into distinct clusters. Some documents from Explainable Artificial Intelligence and Algorithmic Transparency clusters are close to each other, but still clearly separated. Search Engine Bias and Gender Bias clusters almost did not overlap with any cluster.

Overall, the plot provides a clear visualization of the relationships between the queries.

The plot shows the results of a clustering algorithm applied to a set of documents from four different folders: "Search Engine Bias", "Gender Bias", "Algorithmic Transparency",

and "Explainable Artificial Intelligence". The goal of the clustering algorithm is to group similar documents together based on their content.

The clustering results appear to be reasonable and reflect the expected patterns based on the contents of the four folders. However, it is important to note that the quality of the clustering results depends on various factors, including the choice of clustering algorithm (Kmeans cluster result is known to be affected by the initial set of seeds), the number of clusters (in our case we knew beforehand that we had 4 different categories), and the choice of features used to represent the documents (we tried to clean the data to get good results).

Explaining errors:

There are possible errors in general that lead to bad clustering in K-means :

- **Choosing an inappropriate number of clusters**: The number of clusters chosen can have a significant impact on the quality of the clustering results. Choosing too few clusters can result in poor separation of the data points and obscure patterns, while choosing too many clusters can lead to overfitting and poor generalization. In our case, we have documents from four different directories, and data will cluster according to its original directory. Therefore, choosing four clusters is reasonable as it aligns with the underlying structure of the data. However, in other cases where there is no prior knowledge about the underlying structure, we can use various techniques such as elbow method.
- **Sensitivity to initialization**: K-means is sensitive to the initial placement of the centroids. Different random initializations can lead to different clustering results and starting with a bad set of seeds results in a bad clustering. To mitigate this issue, we can use multiple initializations with different seeds and choose the clustering result that has the best clustering (in our case, we used purity score to get best cluster output).
- **Noise and outliers:** Text data can contain noise and outliers, leading to poor clustering results. To handle this, we can use outlier detection techniques to identify and remove the outliers before clustering, in our case we showed the cluster results before and after handling outliers and noise, and how clustering improved.
- **Preprocessing techniques**: Preprocessing techniques such as text representation and feature selection can have a significant impact on the quality of the clustering results. To improve the results, we can use different text representations such as bag-of-words, TF-IDF, or word embeddings, or use different feature selection methods such as mutual information, chi-squared, or feature importance scores.
- **Term frequency-inverse document frequency (tf-idf) sparsity** - This refers to the problem of having many features (i.e., words or terms) in a dataset, where

many of these features occur only a few times across the entire set of documents (i.e., high term sparsity), leading to a high-dimensional and sparse feature space., we also handled this as we dropped rare terms when we handled outliers.

Further inspection of cluster:

We took a closer look at the clusters results and given labels; for example, we took a closer look at the document 'A "Scientific Diversity" Intervention to Reduce Gender Bias in a Sample of Life Scientists' from Gender Bias directory, that was mislabeled as "Search Engine Bias", which is our query, and since we know the language model of our query, it was easy to check the mislabeled document, and when we examined it, we saw that our query terms occurred a lot in the document, especially the terms 'bias' and 'search', these two directories had some overlapping (not a lot), which is reasonable since they have a common query term 'bias', in common, but in general they were separated.

Another observation was that there was a lot of overlapping between the directories "Algorithmic transparency" and "Explainable artificial intelligence", and after we looked at some of the mislabeled files from both directories, we saw that in most cases, they had an observable presence of same terms with high TF values.