UNSUPERVISED MACHINE LEARNING

NLP: TOPIC EXTRACTION AND TEXT CLUSTERING

FINAL PROJECT SUMMARY

By vectorkoz, March 2024

Contents:

I. Introduction and specification of the project goals

In Natural Language Processing, topic extraction and text clustering are common tasks with many practical applications. While usually they are performed separately, in this project they are performed on the same dataset, and it is discussed how one might benefit the other.

The goals of this project are to analyze the labelled topics of the existing text dataset by creating Non-negative Matrix Factorization models that split the text into topics as closely to the labels as possible, as well as using dimensionality reduction and clustering to split the text into clusters as closely to the labels as possible. This project also serves as the final assignment of the online course "IBM Unsupervised Machine Learning".

This project was created with Python in JupyterLab using scikit-learn, pandas, numpy, matplotlib and seaborn packages.

This report contains dataset summary and short descriptions, visualizations and summaries of multiple dimensionality reduction and clustering models, as well as topic extraction results, conclusions and suggestions for future work.

II. Data summary

In this project, CNN and FOX News data from Mingjie Qian's Homepage was used. The dataset contains RSS feeds of CNN and FOX news websites from Jan. 1st, 2014 to Apr. 4th, 2014. The dataset also contains pictures and results of text processing done by the authors, but for this project, only raw text files were used.

The dataset consists of hundreds of text files in 11 different folders: 7 (**'cnn_crime'**, **'cnn_politics'**, **'cnn_living'**, **'cnn_health'**, **'cnn_technology'**, **'cnn_travel'**, **'cnn_entertainment'**) for CNN data, and 4 (**'foxnews_health'**, **'foxnews_Science_technology'**, **'foxnews_sports'**, **'foxnews_travel'**) for FOX News data. These folders correspond to different sections where the articles were posted online.

The names of the folders were treated as labels of the text data. Since this is an unsupervised learning project, the labels were used only to determine how well the clusters aligned with the original topics, as well as which news section was represented by a NMF topic. The labels were not used during training.

The text files contained article texts, as well as titles and other information. During text preprocessing, only the relevant parts of each article text were selected. Then, the resulting texts were encoded via either raw term counts (a.k.a TF, Term Frequency), or TF-IDF (Term Frequency – Inverse Document Frequency). During text encoding, some common stop words were ignored, as well as terms that appeared in <1% or >95% of texts. Only the 1000 most frequent words were used.

A few files in the original dataset were missing any article text, and therefore were deleted.

Different folders had different amounts of text files. To prevent problems with unbalanced classes, the minimum numbers of texts in a folder from each news agency were determined (148 for CNN and 178 for FOX News), and only that many files from folders of each news agency were analyzed.

III. Summary of modeling and visualizations

First, NMF was used on the TF-IDF data to extract some text topics. Since the whole dataset contains a lot of categories, NMF was used on many different data samples containing of 2, 3 or 4 categories. The number of topics always equaled the number of categories in the data sample. Then, the most prevalent topics for each document were calculated. Since the goal of the project was to create topics that closely align with the original data categories, the "goodness of alignment" of NMF topics to the original categories was evaluated from the point of view of supervised learning: most prevalent topics for each document were used as predicted class labels, and the results were evaluated using accuracy for 2-category samples, and multiclass recall for samples with 3 or 4 categories, as well as confusion matrices. Confusion matrices were plotted using heat maps.

For the sake of comparison, NMF was also attempted on TF data.

Secondly, clustering texts with the goal of aligning clusters to the original topics in the dataset was attempted. Since the text data had 1000 dimensions, dimensionality reduction was needed before clustering. In this project, dimensionality reduction of TF data was performed via t-distributed Stochastic Neighbor Embedding (t-SNE) using cosine distance. T-SNE embedded 1000 dimensions of the data onto 2, or, in a couple of cases, 3 dimensions.

Then, clustering via K-Means and GMM was performed on the reduced dimensions. Clustering results were visualized with scatter plots and compared to original categories. "Goodness of alignment" of clustering results to the original categories was evaluated using the same principles as with most prevalent NMF topics for each document.

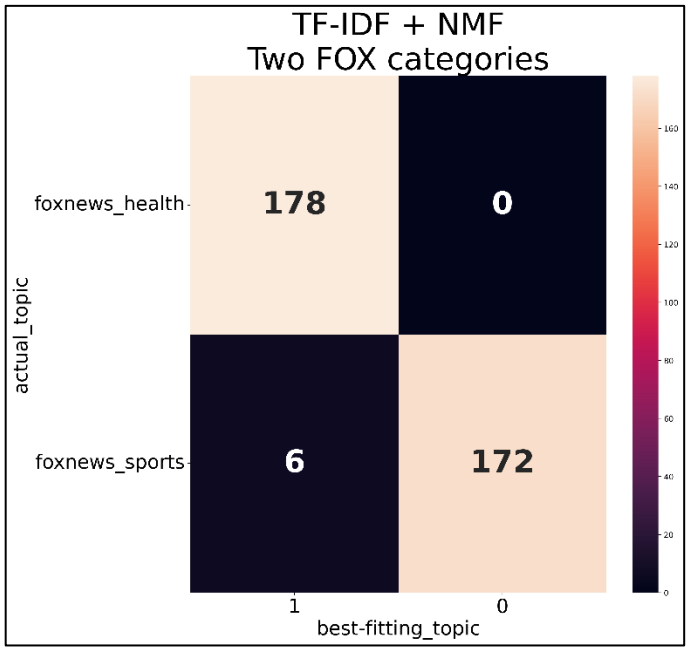
For the sake of comparison, the same thing was attempted on TF-IDF data.

Finally, some NMF models on TF-IDF data from the first section that aligned well with the original classes were analyzed by selecting and top 10 most prevalent words for each category. These results were visualized via bar plots.

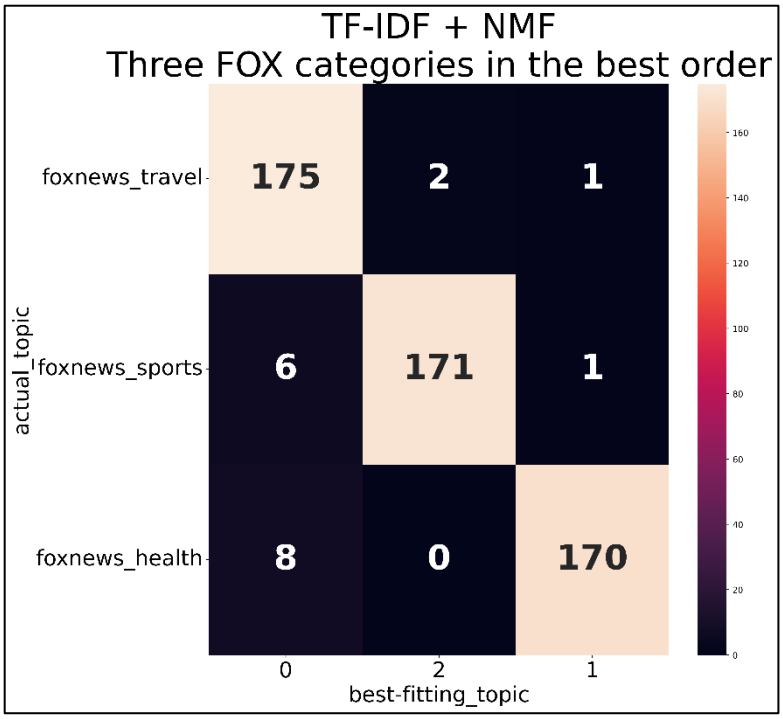
IV. NMF topic extraction

For TF-IDF data, NMF models were applied to all 2-class and 3-class samples of FOX News data, all 4 classes of FOX News data together, and all 2-class, 3-class and 4-class samples of CNN data.

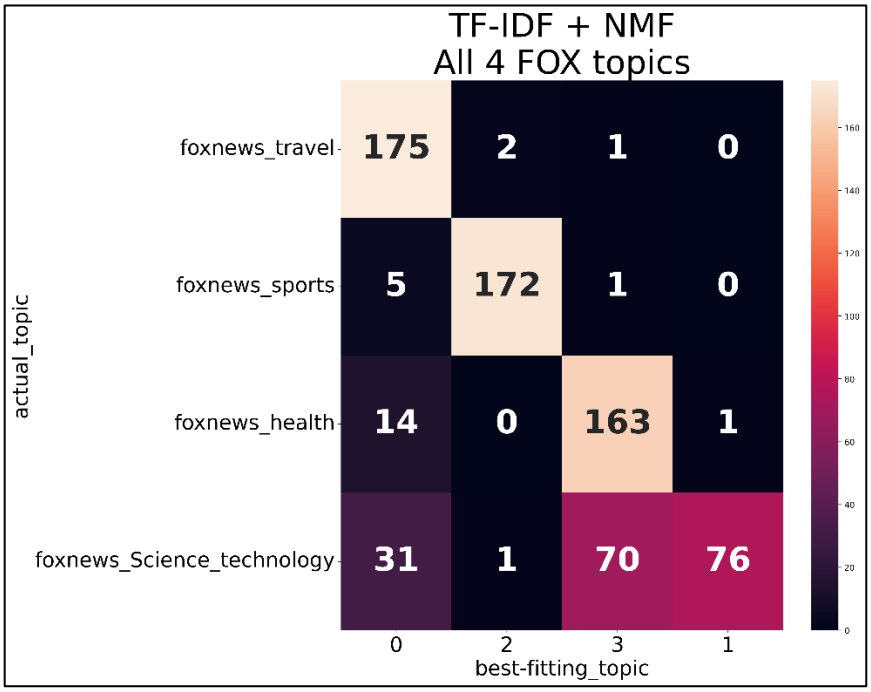
For all 2-class samples of FOX News data, NMF was able to achieve good splits without changing hyperparameters, with the lowest accuracy score of 0.949. For the sample of **'foxnews_health'** and **'foxnews_sports'**, the accuracy reached 0.983. Below is the confusion matrix for that model.



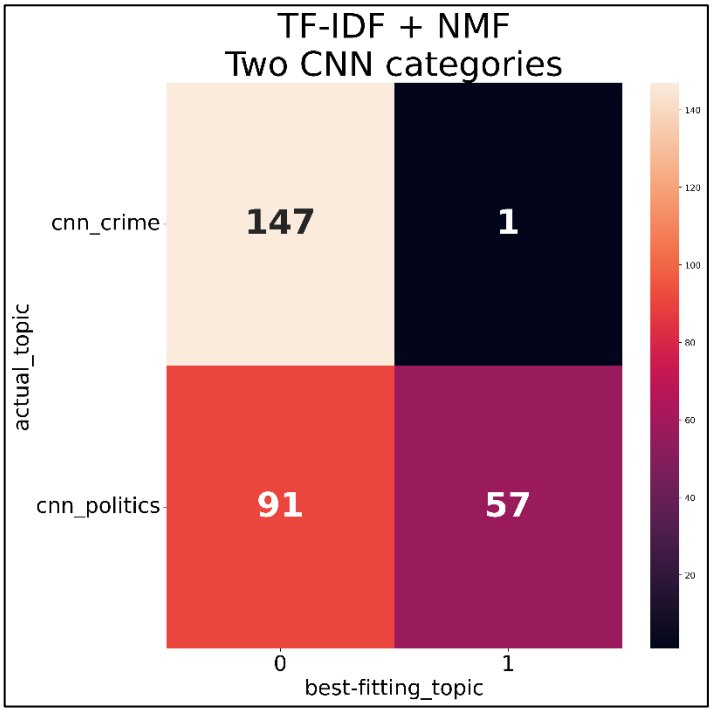
For all but one of the 3-class samples of FOX News data, NMF was able to achieve good splits without changing hyperparameters, with the lowest recall score of 0.915. However, for the other 3-class sample, the recall score was low due to the model grouping most of the 2 out of 3 classes into a single topic. Changing either the random state of initialization of NMF, or the order of rows in the input data led to big improvements, with recall score reaching over 0.96. Below is the confusion matrix for the NMF model with the same hyperparameters, but a different order of data rows.



When using NMF on all 4 classes of FOX News data together, a decent recall score of 0.823 was achieved. As can be seen on the confusion matrix below, most texts from **'foxnews_Science_technology'** were "misclassified", but more importantly, for each NMF topic, the majority of texts clearly correspond to one label, and all 4 labels are represented by a topic. Therefore, this model can be used for NMF topic analysis.



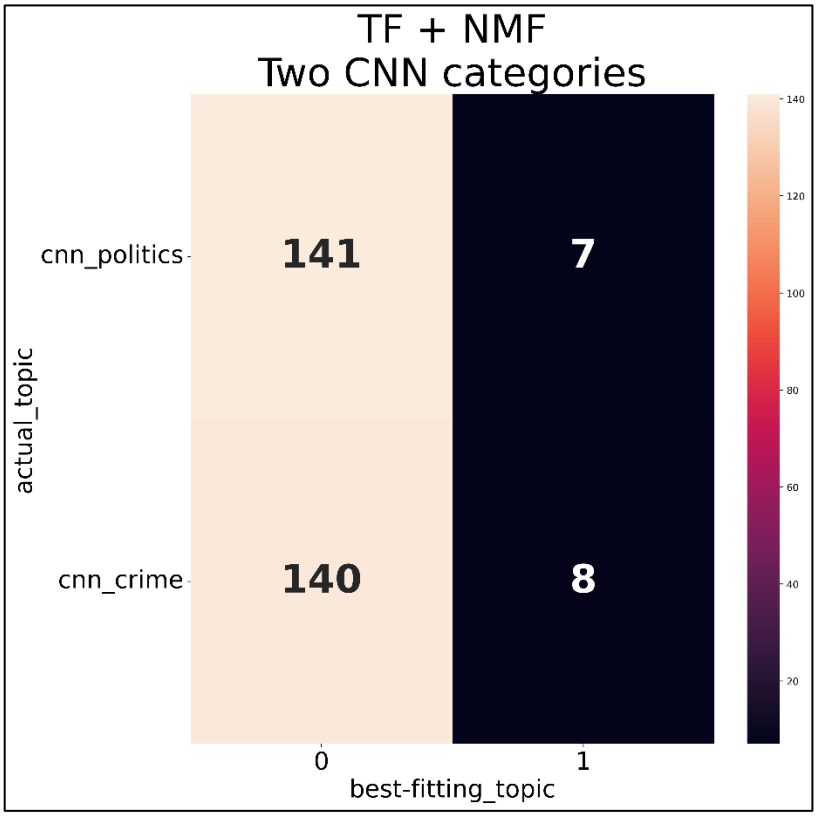
When applying NMF to 2-class samples of CNN data, many good results were achieved, including multiple with the accuracy score of over 0.94. However, for other samples, no good results were achieved. For the combination of **'cnn_crime'** and **'cnn_politics'** data, even after changing the random state of initialization and data row order, the recall score did not go over 0.689. As seen below, NMF separated a minority of the **'cnn_politics'** data in one topic, heaping the rest of the data together in the other topic.



When applying NMF to 3-class samples of CNN data, many good results were achieved, including multiple with the recall score of over 0.8, and the median recall score of around 0.7. However, for many samples, the score was low.

When applying NMF to 4-class samples of CNN data, only a few good results were achieved, including just 4 with the recall score of over 0.7, with the maximum recall score of 0.805.

For comparison, NMF models were also applied to the TF data. On that data, NMF achieved generally worse results than on TF-IDF data. For example, on most 2-class samples of FOX News data it achieved only slightly worse scores. But for the combination of **'foxnews_health'**, **'foxnews_Science_technology'**, the NMF model couldn't produce any useful topics regardless of hyperparameters or data row order. In comparison, NMF achieved the accuracy score of 0.966 on the same texts when using TF-IDF data.



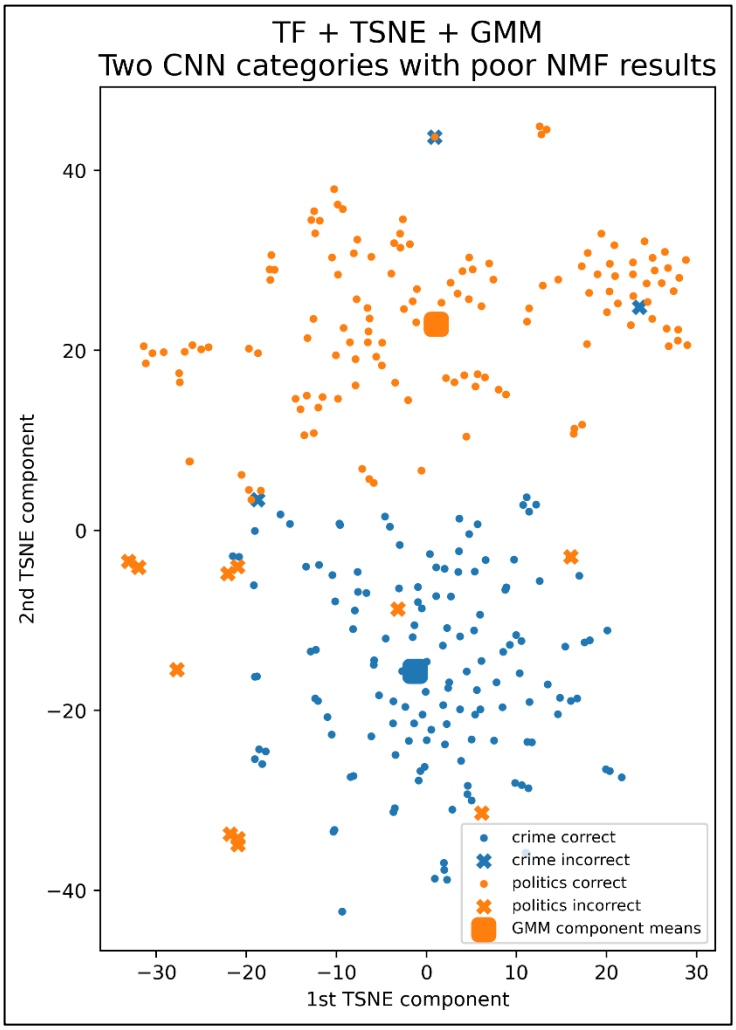
V. Dimensionality reduction with t-SNE and clustering with K-Means and GMM

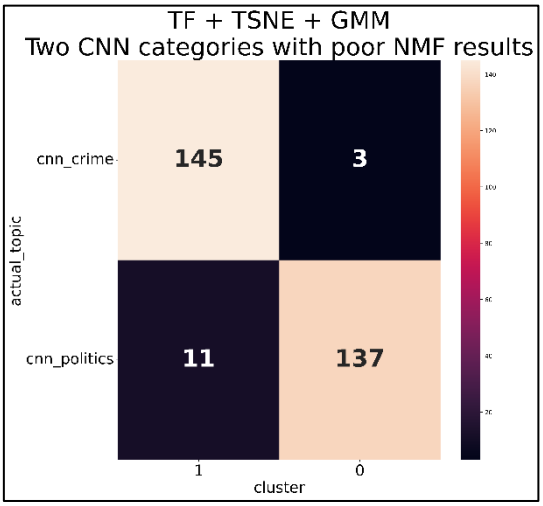
For TF data, dimensionality reduction with t-SNE was applied to multiple samples, for which NMF on TF-IDF data was unable to achieve good splits. During t-SNE application, the perplexity hyperparameter was altered for each sample to produce well-separable data. While in most cases most values of perplexity in the recommended range of [5, 50] led to the data points of different groups being clearly separable in 2 dimensions, some values of perplexity were better than others.

Then, for each sample, clustering with k-means and GMM was applied on the reduced dimensions. K-means clustering required no changes in hyperparameter values and produced good results most of the time. GMM required changes to the covariance_type hyperparameter. Usually, GMM with a wrong value of covariance_type produced worse clustering results than k-means, with a correct value of covariance_type – slightly better results.

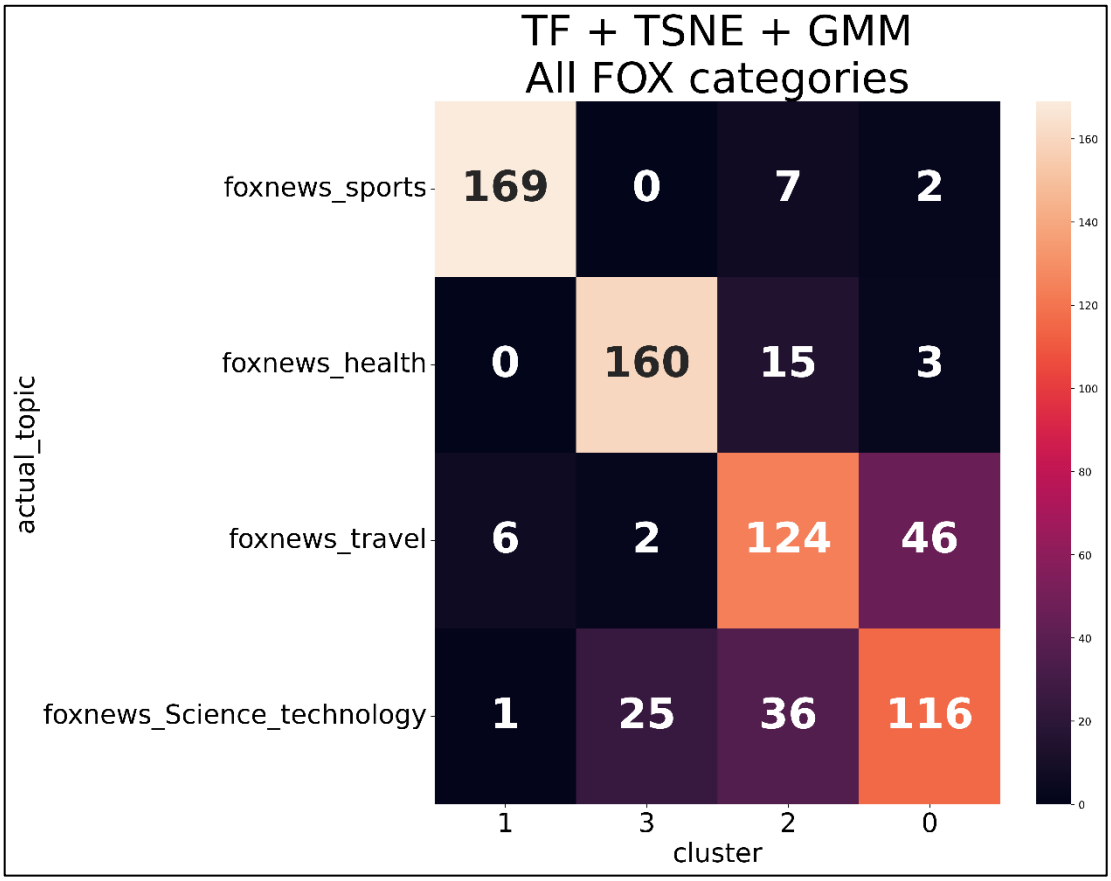
Generally, when applying t-SNE + clustering with correct hyperparameters, the resulting clusters did much better than NMF topics in terms of closeness to the original groups.

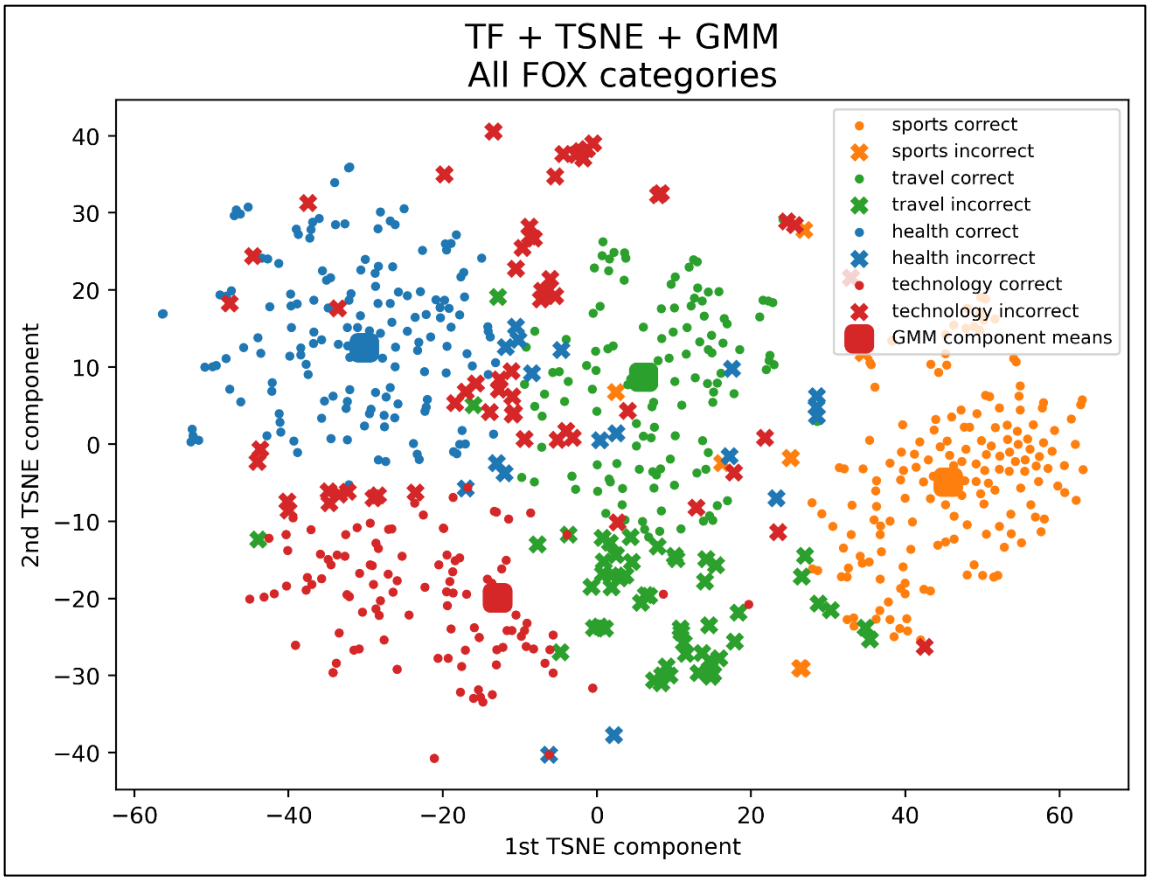
For example, for the combination of **'cnn_crime'** and **'cnn_politics'** data, NMF achieved accuracy of only 0.689. Clustering with k-means was able to achieve accuracy of 0.942, and with GMM – 0.952. The confusion matrix and the clustering plot are shown below.



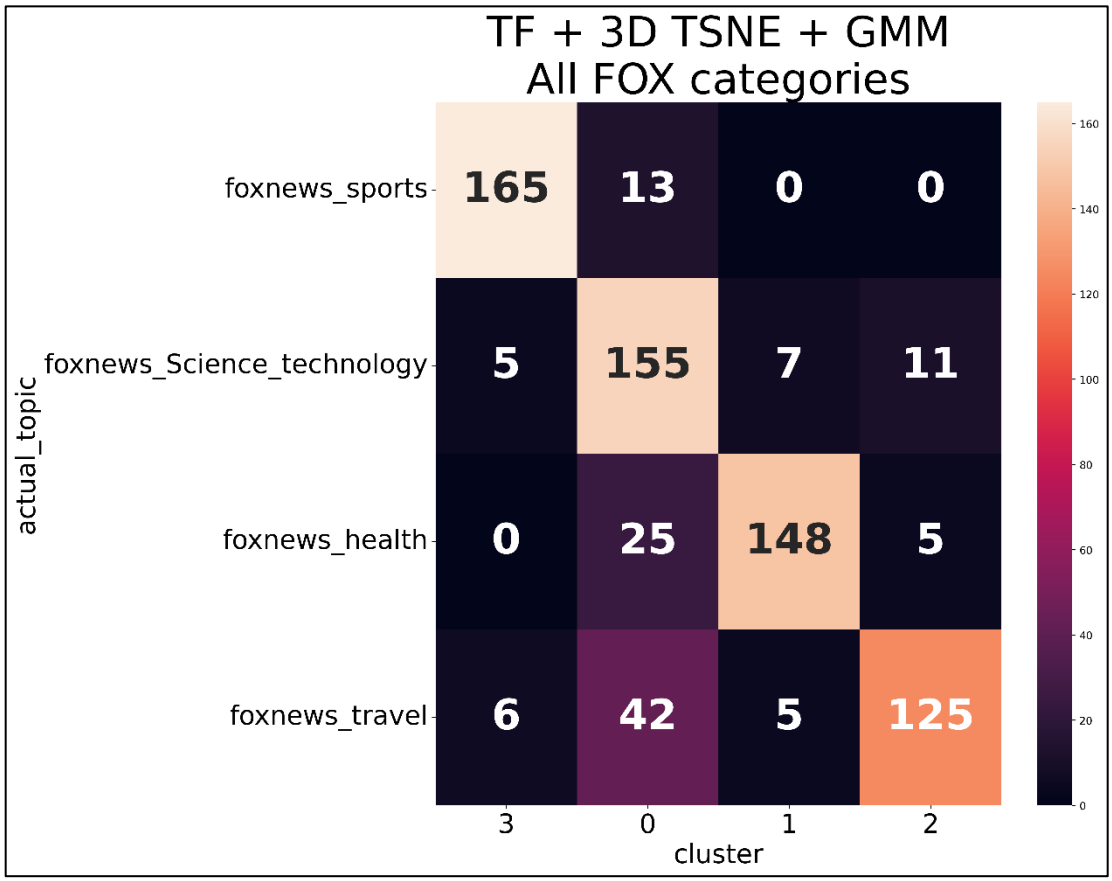


When applying t-SNE + clustering to all 4 FOX News clusters, the recall score achieved was a little lower than for NMF (0.79 vs 0.82), but unlike for NMF, the most data points belonged to their own cluster for all 4 categories. The confusion matrix and the clustering plot for GMM clustering are shown below.





When applying GMM clustering to 3-dimensional t-SNE output, the recall score reached 0.83, slightly higher than with all other approaches.

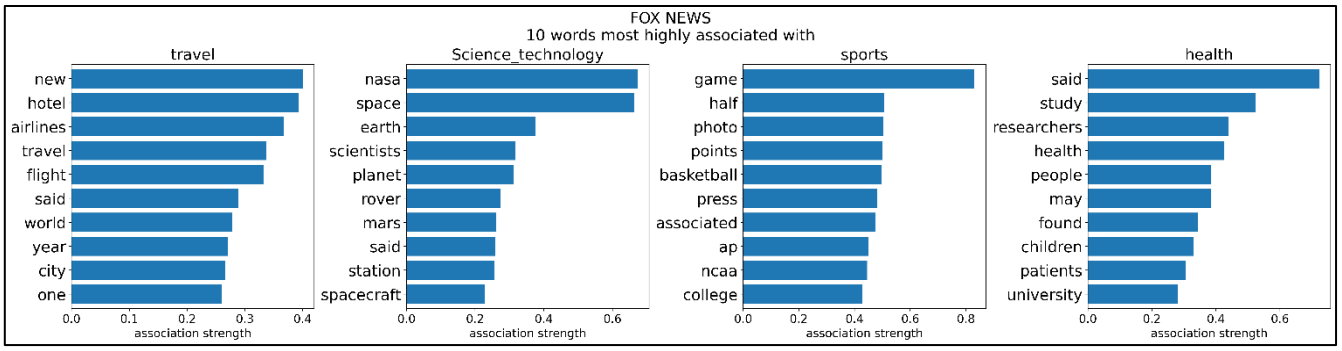


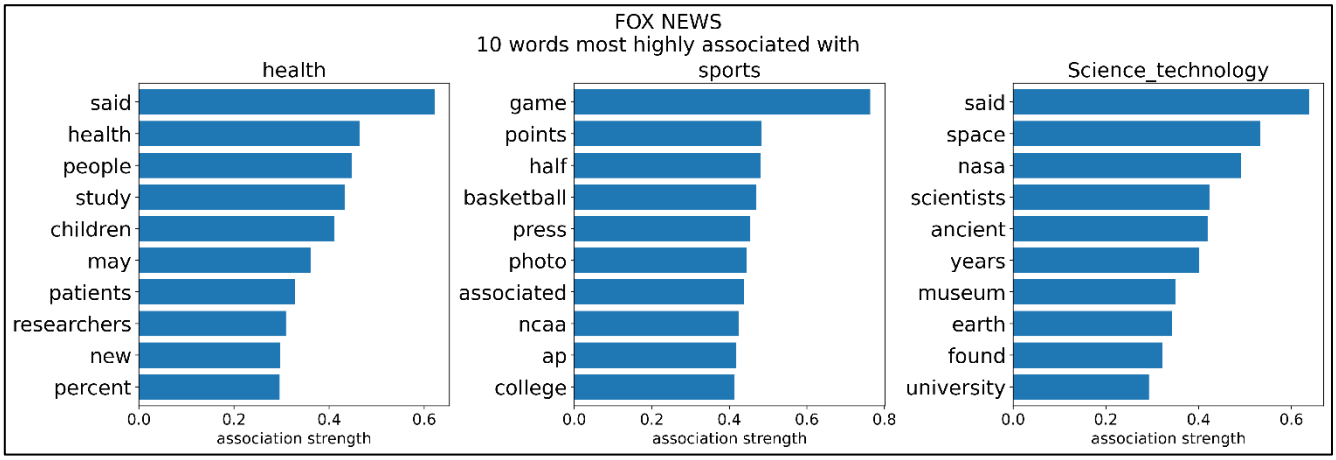
When applying t-SNE + clustering on TF-IDF data, some improvements over NMF scores were also achieved.

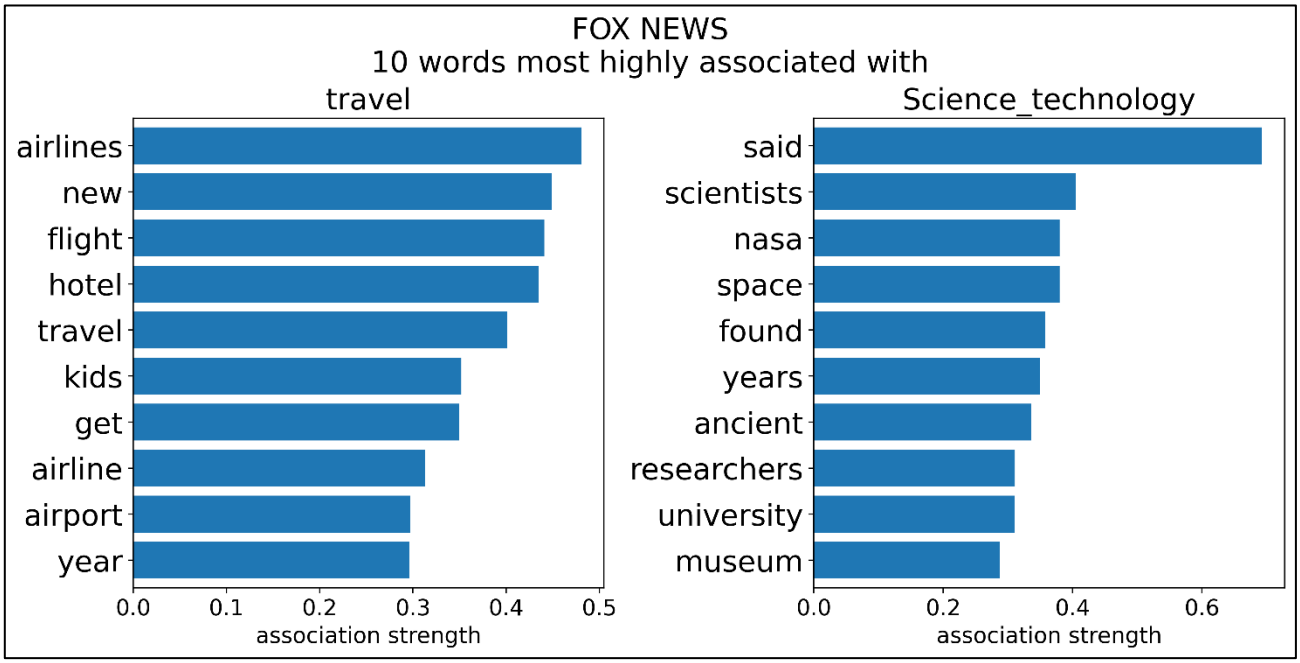
VI. NMF topic analysis

In order to determine what words are most characteristic of the topics in the original articles, NMF models on TF-IDF data with topics best matching the original classes were selected. Then, for each NMF topic, top 10 words were determined.

For FOX News data, the NMF model on all 4 classes was analyzed first. It performed well with a recall score of 0.823. Then, two more well-performing models were analyzed: a 3-class model with a recall score of 0.958, and a 2-class model with an accuracy score of 0.949. Lists of top 10 words for each NMF topic in those models are visualized below, with the NMF topic names replaced by the names of the original classes, to which the topics were most similar in the confusion matrices.



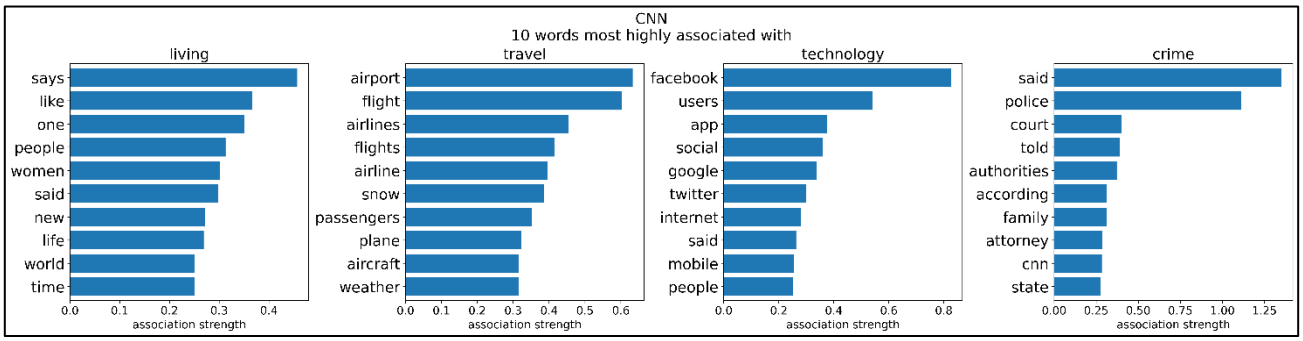


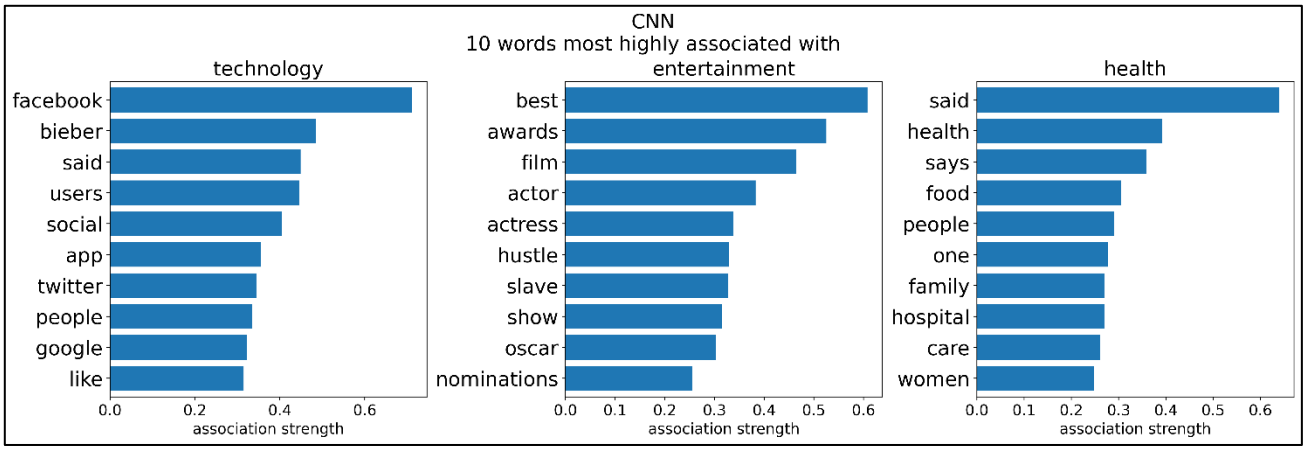


Overall, for FOX News data each set of top 10 words per NMF topic seems to fit very well with the presumed actual topic of the texts. It can be inferred that articles in the **'foxnews_travel'** section talk about flying on the airlines, hotels, cities, new ideas for travel; articles in the **'foxnews_Science_technology'** section talk about space exploration, scientists, history; articles in the **'foxnews_sports'** section talk about games and sports leagues; articles in the **'foxnews_health'** section talk about patients, researches, new treatments, children. Moreover, for any class the lists of top 10 words per topic are quite similar across different models.

The only downside is the presence of the word “said”, which is commonly used in all kinds of news articles to quote someone. It shows up in different topics without adding much information. It should have been added to the list of ignored words during text processing.

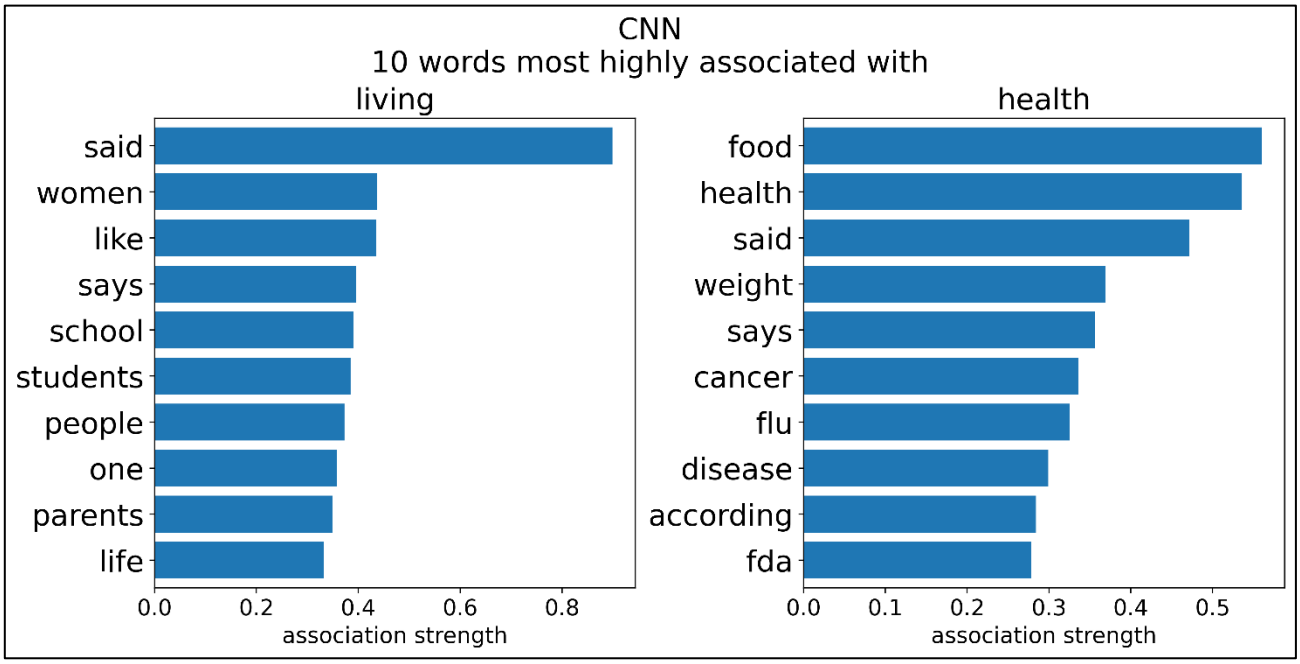
For CNN data, two well-performing models were analyzed first: a 3-class model with a recall score of 0.808, and a 4-class model with an accuracy score of 0.766. Lists of top 10 words for each NMF topic in those models are visualized below, with the NMF topic names replaced by the names of the original classes, to which the topics were most similar in the confusion matrices.





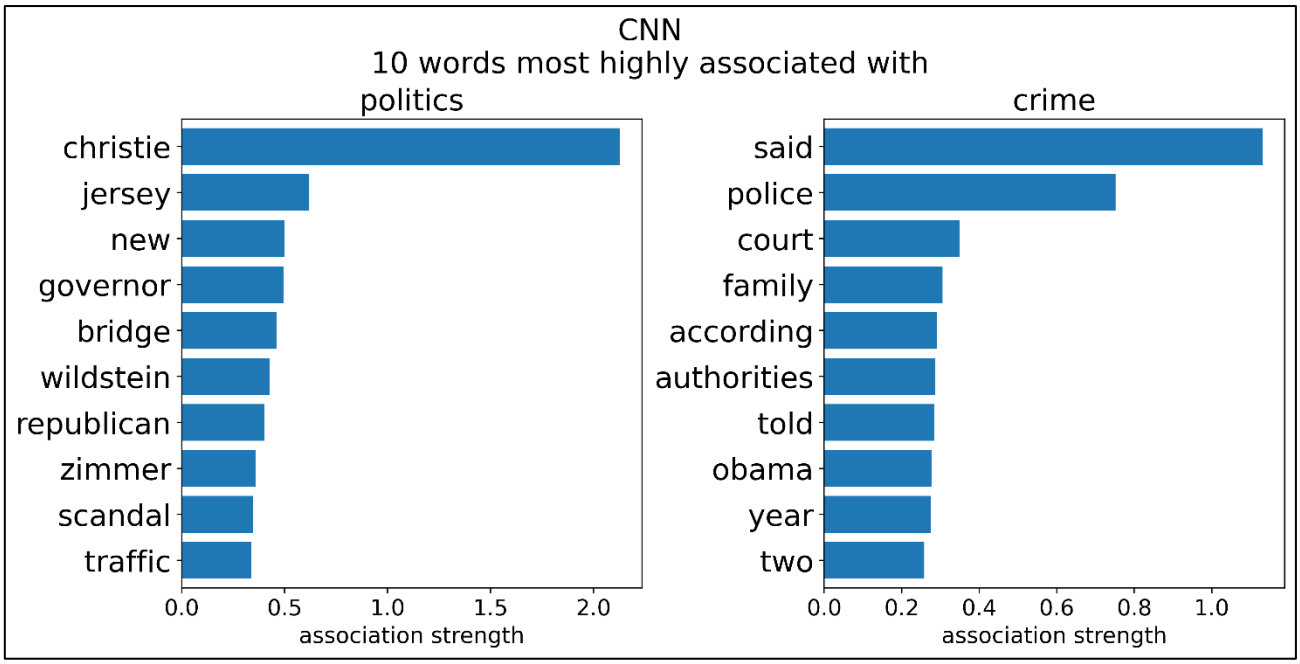
These two NMF models cover 6 of the 7 CNN topics, and, similarly to FOX News data, each set of top 10 words per NMF topic seems to fit very well with the presumed actual topic of the texts. The only downsides to these results are, similarly to FOX News data, presence of the words “said” and “says”, and some similarity in the sets of top 10 words for the classes **'cnn_health'** and **'cnn_living'**.

To resolve this issue, the model looking only at these two classes was also analyzed. Lists of top 10 words for each NMF topic in that model are visualized below.



Now, the difference between the two topics can be seen clearly. It seems that articles in the **'cnn_living'** section talk about human relationships and **'cnn_health'** talks about diseases.

As for **'cnn_politics'** data, no good NMF model had been constructed using it. For analysis, one model on **'cnn_politics'** and **'cnn_crime'** data was analyzed. Lists of top 10 words for each NMF topic in that model are visualized below.



As can be seen in the confusion matrix for this model shown above, NMF separated a minority of the **'cnn_politics'** data in one topic, heaping the rest of the data together in the other topic. Upon further analysis, the part of the **'cnn_politics'** data used for this project contained many stories about one thing: "Bridgegate", a political scandal in the U.S. state of New Jersey involving Chris Christie (governor of New Jersey at the time). NMF tends to separate stories about "Bridgegate" into its own topic. On the other hand, words that presumably would be important to politics, such as `obama`, ended up in a different topic. This issue should be solvable with better sampling and/or a higher sample size.

VII. Results summary and conclusions

In this project, texts from 11 different topics were analyzed using Non-negative Matrix Factorization. For articles of each topic, lists of most prevalent words were found, which clearly match the presumed topics of the articles. It was shown that NMF can be a powerful tool that, while using text data correctly transformed via TF-IDF, correctly separates texts in a sample in different groups, which are easy to analyze.

However, since NMF is not an exact model, for certain values of random initializations and/or orders of rows in the dataset, it can fail to split the texts of different topics well even when such a split is possible. And for other data samples, it can fail to produce a good split altogether. That's why for this task of exploring already existing text categories, it's important to check the relationship between NMF topics and presumed text categories in a confusion matrix every time before analyzing NMF topics.

It makes sense to use NMF only on TF-IDF data, which is unsurprising, since TF data doesn't take the influence of lengths of documents into account.

If the goal is to cluster existing groups of texts, NMF can also be used, but clustering after dimensionality reduction generally produces better results. t-distributed Stochastic Neighbor Embedding is a powerful technique, that can negate the impact of lengths of documents by using cosine distance on TF data.

In most cases, t-SNE produced nicely separable data in only 2 dimensions. However, t-SNE required correct hyperparameters to work best on a given data sample.

When clustering on t-SNE outputs, k-means with the default k-means++ initialization was a good enough option in most cases. GMM clustering could further improve results, but it required hyperparameter tuning of its own.

Overall, t-SNE + clustering produced topic splitting results at least as good as NMF, and in most attempted cases, even better results.

VIII. Suggestions for future work

As a result of this project, these ideas for further exploration are proposed.

1) **Use other algorithms for topic extraction and comparing the results**. For example, Latent Dirichlet Allocation.
2) **Use other dimensionality reduction methods**. For example, Multidimensional Scaling.
3) **Compare similar categories from different news organizations against each other.** This could point out the differences between different news organizations, not just categories.
4) **Get more text data.** For more ambitious projects, it would be a good idea to use datasets with at least thousands of texts, not hundreds, as in this dataset.
5) **Try to use text clustering as a classification method.** Since clustering methods such as k-means and GMM can be trained on one data and applied to other data, it could be possible to use it as a prediction model, with the cost function depending on correctness of classification as well as clustering-specific performance metrics such as distortion, inertia, or silhouette score.

Links:

- Project source and full code: https://github.com/vectorkoz/my-nlp.git
- Online course: https://www.coursera.org/learn/ibm-unsupervised-machine-learning/
- Data source: https://sites.google.com/site/qianmingjie/home/datasets/cnn-and-fox-news