The Comparison of Local and Global Early Fake News Detection Methods Project Final Report for NLP Course, Winter 2023/24

Hubert Ruczyński
WUT
01151402@pw.edu.pl

Bartosz Siński
WUT
01151411@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Maciej Pawlikowski
WUT
01151389@pw.edu.pl

Adrian Stańdo
WUT
01151435@pw.edu.pl

Abstract

With the increasing impact of social media on our lives, scientists put more and more focus on the dangers connected to it. Two crucial areas regarding those matters, present in Natural Language Processing are topic and fake news detection methods. The first one attempts to grasp the idea about the matters discussed online, whereas the latter one, focuses on detecting untrue information, which yields a negative impact on millions of people. Both fields have become more important in recent years, and even though they are closely related, no one decided to merge them.

In our work, we attempt to bridge the gap between those areas, as they might benefit from one another. One of the major issues connected to fake news detection is the sheer amount of data regarding the discussions on social media. We propose a framework, where we first conduct a topic modeling and then train and evaluate the models on smaller topics, which lets the models be fine-tuned for a particular discussion. In our work, incorporate, and evaluate this approach, and discuss its opportunities, and shortcomings.

1 Introduction

Nowadays most people learn about the world around them from the resources on the internet. Therefore not surprisingly it is used by the majority of the current news media companies. Unfortunately sometimes among the information provided by several reliable news outlets, there are a few misleading articles in which authors want to fool the reader. Their intentions could be to just grasp the attention of the user or in some cases to manipulate the user for personal gains. This misleading information is called fake news. Detecting such articles and differing them from real ones is a very important initiative. Creating a solution that would effectively recognize fake content could prevent the spread of disinformation and prevent people from harm.

As the solution, our team proposes, and evaluates a novel technique of rumor recognition that finds suspicious content at the early stage of its propagation, that combines topic, and fake news detection. The proposed framework, at first detects topics present in given dataset, in order to later train the fake news classifiers for particular topics which is much faster, enables topic-specific models explainability, and might result in higher quality results. In this work, we evaluate such approach, by comparing those local models (trained on a topic) to global solutions (trained on all data), and analyse them with the usage of eXplainable AI (XAI) methods.

2 Related Works

In this section we will describe the papers published in three domains covered by our work, which influenced our research in terms of methodology.

2.1 Topic Detection

The first work regarding topic detection (Leo et al., 2023) focuses on finding the clusters of tweets, describing similar discussion areas. This task is extremely important, as Twitter is currently the biggest platform enabling free, and uncensored thoughts exchange. We can clearly underline two major contributions of this work: the introduction of a stable clustering, and semantical enhancement of short messages (tweets). The first one tackles a major issue of topic detection, which is a machine learning (ML) task with fairly unstable results, especially because of issues with select-

ing a proper number of clusters, which results in chaotic transfers of observations from one cluster to another. An answer to this problem is the usage of Non-Negative Matrix Factorization (NMF) with consensus clustering. The idea behind consensus clustering is that by repeating the clustering operation many times with varying NMF regularization parameters, the words that will stay most of the time in the same cluster are likely to be the correct cluster members. The authors additionally point out various important attributions of tweets, connected to their length. As their maximum length is 280 characters (before - 140), we can assume that a single tweet can carry only one topic, which is a very small assumption. However, it also indicates some drawbacks, as a singular tweet corpus is rather small, and it is hard to carry its true meaning. The paper introduces a semantical enrichment strategy, where we select the most important words, and with the usage of embeddings add similar variations of them, so a singular tweet can carry more information.

Another important work (Lossio-Ventura et al., 2019) in this area, compares various LDA approaches and suggests the data preprocessing options applicable to topic detection for tweets task. The paper presents how to efficiently use Calinski-Harabasz index (Caliński and JA, 1974), and Silhouette Coefficient (Rousseeuw, 1987) for clustering evaluation, and shows us, that for this kind of data, GibbsLDA (Wei and Croft, 2006), and Online Twitter LDA (Lau et al., 2012), prove to be better than their counterparts.

2.2 Fake News Detection

There is a multitude of works dedicated to fake news detection describing a lot of ways to approach this subject. For example in (Kasra Majbouri Yazdi, 2020) authors focus on feature selection based on computing similarity between primary features in the fake news dataset, clustering obtained features using K-means (Guo et al., 2004), and selection of final attributes of all clusters. The paper describes in detail how all algorithms used are calculated and presents the value of the proposed feature selection method using it combined with SVM (Evgeniou and Pontil, 2001) to achieve very good results when it comes to fake news detection.

Another approach proposed in (Tian and Baskiyar, 2021) not only allowed authors to achieve high accuracy by utilizing Genetic and evolutionary Feature Selection and KNN in the fake news detection but also tested the quantum version of k- nearest neighbors. This research went in depth when it comes to testing the above-mentioned methods on the BuzzFace (Williams and Santia, 2018) dataset which consists of 2282 news articles and posts about the 2016 election from Facebook, which was divided into several categories: mostly fake, fake, mostly true, true, and mixed true and fake.

2.3 Explainability

Most of the SOTA models developed for the fake news detection task aim to have the greatest performance and accuracy on selected datasets. Lately however as shown by (A.B. et al., 2023) new techniques and methods have been created that focus on gaining better insight into the model decision-making process. Authors argue that explaining model predictions is the key gate-away to achieving better results. The authors present 11 SOTA explainable fake news detection methods of which 7 are attention-based approaches.

One example (Kurasinski and Mihailescu, 2020) of the attention-based method visualizes attention weights as the color-coded text to show the impact of the particular words on the prediction. For the detection task authors use two deep learning models. First is the BiDir-LSTM-CNN which is the architecture that combines convolutional neural networks and bidirectional recurrent neural networks. The second one is bidirectional encoder representation from transformers (BERT (Devlin et al., 2019)). Used color-coded visualization to show how models distribute attention differently. Another interesting finding was that both models showed a strong correlation between click-bait content such as *Check it out!, MOST IMPORTANT* and fake news. Models were trained on the ”Fake News Corpus” (Pathak and Srihari, 2019) which is the data set we are using in our solution. For both models preprocessing methods: *summarization, stemming* and *lemmatization* worsened the results.

3 Datasets

Online news and posts can be collected from a variety of sources via dedicated APIs or by scrapping. Nonetheless, manual annotation is a challenging task requiring annotators with domain ex-

pertise. For these reasons, we will make use of open-source data available on the Internet. As for now, not many datasets, regarding fake news detection, are available, as well as there is not one commonly used benchmark dataset. In this section, we will present a few data sources with short descriptions and a list of their shortcomings, to finally describe the dataset we will use.

3.1 Existing Resources

1. *BuzzFeedNews* - The dataset used in the paper (Tian and Baskiyar, 2021), contains data on news published on Facebook during a week before the US election in 2016. Every post was annotated by 5 people, however, the dataset contains only links to Facebook posts, so the content of articles is not available. Due to the limited time for the project, we resigned from this source, as we would require a web crawler to gather full texts.

2. *CREDBANK* - The dataset described in (Mitra and Gilbert, 2021) contains 60 million tweets that cover 96 days in 2015. Access to the data is restricted - they have to be downloaded from the AWS cloud for a small fee, which we are not able to pay. Moreover, labels are not provided for the tweets - only events were identified and annotated by 30 different people.

3. *FakeNewsCorpus* - The dataset includes around 9 million news articles (around 30 GB of data). It contains, among other things, the text of the articles, their title, and source. It also provides labels for 11 different types of misinformation with the addition of class *Reliable*. It was created by scrapping text from more than 1000 different Internet domains. Each article has been attributed the same label as the label associated with its domain.

There are many other data sources, however, they are smaller and, hence, it may be difficult to create a reliable model using them. All things considered, we decided to use the last dataset - *FakeNewsCorpus* - as it is the most extensive one and contains long enough article content. In our study, we will use the annotation information provided by this source as a ground truth, and the content of crawled websites in order to extract meaningful text features.

3.2 Data Preparation

As the most important data for us are the raw texts included in the dataset, we have to cope with very difficult type of data and put additional care into the preparation step. The *FakeNewsCorpus* is extremely large, as it contains over 20GB of data, thus we had to limit ourselves to a smaller number of examples. For the final subset, we randomly selected 12,000 observations, with 1000 records per class. This way we can ensure that our solution will be able to work properly for each type of news, which would not be possible with original distribution, where most of the classes are heavily underrepresented.

In our work, we relied mostly on the features provided by the SpaCy package for Python, which was used to create our in-house EDA for the NLP (Exploratory Data Analysis for Natural Language Processing) package. For that case, we incorporated a pipeline called *en_core_web_md*, which is a medium-sized model for analyzing English texts. The full list of preprocessing steps is listed below.

1. We removed all numbers and formatting artifacts, such as '$\backslash n$' from the raw content of articles, in order to ensure higher data quality.

2. We modified the basic stop-words of the used model by removing the negation stop-words, as they are extremely important to the sentence's meaning.

3. For each article, we created a SpaCy document object, which was used for further preprocessing methods, and saved for later usage.

4. We enriched the dataset by calculating statistics, such as the word counts, character counts, word density (word count/character count), and sentiment features, namely the polarity, and subjectivity.

5. For the EDA, and modeling, for each document we additionally prepared the lists of nouns, noun chunks, named entities, and lemmas.

6. We calculated tf-idf scores, and prepared the top tf-idf table. The table contains the top 10 phrases with the highest tf-idf scores for each article.

As a result, we obtained a dataset with 1200 records, whose structure is presented in Table 1. Additionally, we also saved the top tf-idf table, used en model, and the documents created with it. To know more about the dataset characteristics, take a look at Appendix A

Column	Description
id	Article ID.
type	Type of news (ex. fake news).
domain	Scrapped web page.
scraped_at	Date of scrapping.
url	URL of the web page.
authors	Articles authors.
title	Article title.
content	Articles content.
word_count	Number of words.
char_count	Number of characters.
word_density	$\frac{word\_count}{char\_count}$.
polarity	Sentiment polarity score.
subjectivity	Sentiment subjectivity score.
nouns	A list of nouns.
noun_chunks	A list of noun chunks.
entities	A list of named entities.
lemmas	A list of lemmas.

Table 1: **The description of the dataset created after running our preprocessing pipeline.** The table contains 8 columns (first 8) that were also present inside the original dataset, as well as the values calculated during the preprocessing stage (next 5), and features extracted with SpaCy (last 4).

4 Methodology

The next step is the description of the methods used for each part of the project. The overall pipeline is presented in Figure 1. At first, we start with the full FakeNewsCorpus, which contains over 9,000,000 articles, with 12 assigned classes, which weigh around 20GB. After the data preprocessing, we are limiting it to 100MB with 12,000 articles, and we are also enhancing it by adding the columns mentioned in the previous section. Those datasets are later used during the topic detection phase, where we test the multitude of clustering approaches, evaluate them, and the best ones are forwarded to the next phase, being fake news detection. At this point, we are training the models on all datasets, and choosing the best ones, which is called a global approach. After that, the best ones are trained and evaluated on the clusterings from the previous step, which is called a local approach. At this point, we will be able to say, whether this method is feasible or not. Finally, the models are explained with the usage of XAI methods.

4.1 Topic Detection

Topic detection was performed on the extracted lemmas and noun chunks from the dataset. We have created several different clusterings of fake news data by testing the performance of both topic modeling methods and standard clustering methods. In order to perform standard clustering we have used the K-means (Jin and Han, 2010) algorithm with TD-IDF data representation and Doc2Vec (Le and Mikolov, 2014). As for the topic detection model, we have tried Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Latent Semantic Indexing (LSI) (Kontostathis, 2007), and Gibbs Sampling Dirichlet Multinomial Mixture (GSSDMM) (Yin and Wang, 2014). To obtain the clusters from topic modeling, the cluster label of each document is determined by selecting the topic with the highest probability for that document. Additionally, LSI and LDA were tested on raw lemmas and noun chunks, as well as their TF-IDF representations.

For each of the tested methods, the number of clusters had to be determined. For this purpose, the Calinski-Harabasz index, and Silhouette Score were utilized along with the Coherence Score which is specific to LDA and LSI algorithms. Moreover, for topic modeling results we have calculated the above metrics on a feature vector where each element of the vector represents the probability of the document being associated with a particular topic. When determining the number of clusters for K-means, we have also examined the within-cluster sum of square distances metric. After obtaining a correct number of clusters for each of the clustering methods we have compared their results using once again the Calinski-Harabasz index and Silhouette Score. Two best clustering were passed to group data for local fake news detection.

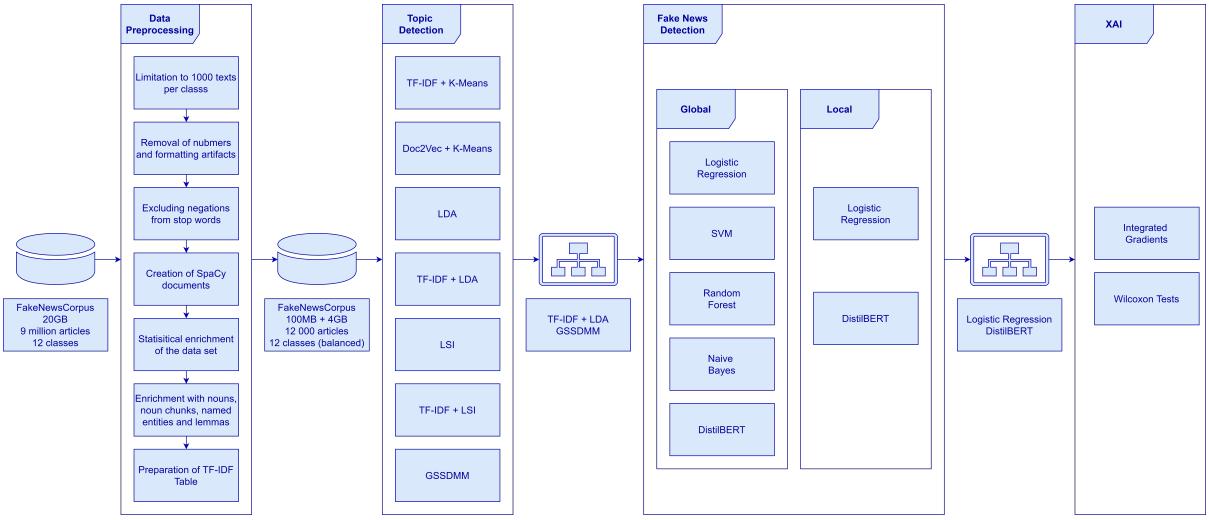

Figure 1: **Experiments pipeline.** The graph represents the pipeline of experiments conducted in this paper. It goes from the initial dataset, through, data preprocessing, topic detection, and fake news detection, to XAI outcomes.

4.2 Fake News Detection

In order to build a successful Fake News Detection model we first need to preprocess our dataset. We began by encoding the data and separating it into training and validation subsets. As there is no one best encoding that fits all NLP tasks we tested several different techniques including Count Vectorizer, Hashing Vectorizer, and TF-IDF. We tested those vectorizers as a proof of concept on a subset of our data, which contained only 2 classes, called real and fake, using SVM and Passive Aggressive Classifier (Crammer et al., 2006). During PoC we also tested if feature selection methods could enhance our models by analyzing models' performance after K-means feature selection and GeFeS (Sahebi et al., 2020).

Eventually, all clustering methods only worsened the models' performance and were computationally expensive. The Passive Aggressive Classifier based on TF-IDF had the best performance, thus we tested it on an entire dataset with 12 classes with good results. We also tested DistilBERT, where we first used its transformer to encode our data and test that encoding using statistical models. Because those models outperformed their counterparts from PoC we decided to perform experiments using only this encoding. During the final experiments, we examined Logistic regression (Cox, 1958), SVM, Random Forest (Ho, 1995), Naive Bayes (Webb, 2010), and DistilBERT (Sanh et al., 2019). We trained all those models on the entire training data and validated their performance on the testing dataset.

The main goal of our project is to examine if creating models for each topic separately is better than training one big model on the entire dataset, we took the two best topic detection methods as described in 5.2.2 and trained all machine learning models on each topic.

4.3 Explainability

In the project, we applied an explainability method to the DistilBERT model, which is called Integrated Gradients (IG) propsoed by Sundararajan et al. (2017). This approach is based on the gradient operator and can be applied to any neural network, even though the authors of the original paper applied it to networks working with images.

The values of Integrated Gradients are calculated as follows:

$$IG_i(x) = (x_i - x_i') \times \int_0^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha,$$

where F represents a neural network, x' is a baseline input (e.g. black image in the case of images or neutral tokens in the case of neural networks for language processing, i is the dimension of the input.

In the project, the explanations created by the two models will be compared with each other. This will be done with the Wilcoxon statistical test (Wilcoxon, 1992) that tests the null hypothesis that two related paired samples come from

the same distribution. This approach was proposed in (Alarab and Prakoonwit, 2022; Standoet al., 2023). Additionally, because of multiple tests in the series of experiments, the results of the Wilcoxon test are modified with False Discovery Rate (FDR) correction (Benjamini–Hochberg procedure, (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001)).

5 Experiments and Results

In this section we will discuss the results of conducted experiments, step by step. At first we will take a closer look at the comparison of topic detection methods, evaluate them, and choose two best performing ones, which will be used in the next steps. Afterwards, we will reminisce the results obtained during the Proof of Concept to show that feature selection strategies are of no use in our situation. Eventually we will compare the results of models trained on whole dataset, and the topics provided by the clustering methods from previous step. In the end we will analyse the explanations of the best model for both clusterings.

5.1 Topic Detection Comparison

We determined the optimal cluster count by visualizing Calinski-Harabasz and Silhouette scores for all methods with Within-cluster square distances for K-means algorithms. Additionally, we considered Convergence scores for topic modeling methods. All metrics were plotted for different numbers of clusters. Then we looked for which number both of the metrics are the highest or at least have local maximum in the form of peaks. An example of this plot can be found in Figure 2 where both metrics indicate the number of clusters to be four.

In total, we ended up with 14 different clustering results to compare. Values of Calinski-Harabasz and Silhouette scores were calculated and used to find the best clustering. Results of clustering on lemmas are displayed in Table 2 and on noun chunks in Table 3.

Clustering Algorithm	Silhouette Score	Calinski-Harabasz Score
TF-IDF + K-means	0.037988	293
Doc2Vec + K-means	0.133619	449
LDA	0.607199	18668
TF-IDF + LDA	**0.872787**	**91490**
LSI	-0.319640	49
TF-IDF + LSI	0.468400	1655
GSSDMM	0.714127	529

Table 2: **Lemmas clustering** The Table presents the Silhouette and Calinski-Harabasz Scores calculated as an evaluation of different clustering algorithms on lemmas only.

Clustering Algorithm	Silhouette Score	Calinski-Harabasz Score
TF-IDF + K-means	0.067379	314
Doc2Vec + K-means	0.386183	3473
LDA	0.883480	110432
TF-IDF + LDA	**0.929051**	**323993**
LSI	-0.512264	131
TF-IDF + LSI	-0.289569	393
GSSDMM	0.866992	15681

Table 3: **Noun chunks clustering** The Table presents the Silhouette and Calinski-Harabasz Scores calculated as an evaluation of different clustering algorithms on noun chunks only.

We can see that clustering on noun chunks gave overall better results than clustering on lemmas. Among particular algorithms, the best results were obtained by the LDA with TF-IDF representations. It achieved the highest values for both metrics. Moreover, it produced balanced clusters which can be seen in Figure 3. LSI and K-means assigned most of the observations to a single cluster and left the clustering unusable in further parts of the project. Another algorithm that performed well was the GSSDMM, for both lemmas and noun chunks it had second second-best result after LDA. The worst-performing topic modeling method was the LSI which in most cases scored below the baseline K-means clustering. Number of clusters which was most often indicated as optimal was 4. For further processing and fake news detection, we have chosen LDA on TF-IDF representation with 4 clusters and GSSDMM with 7

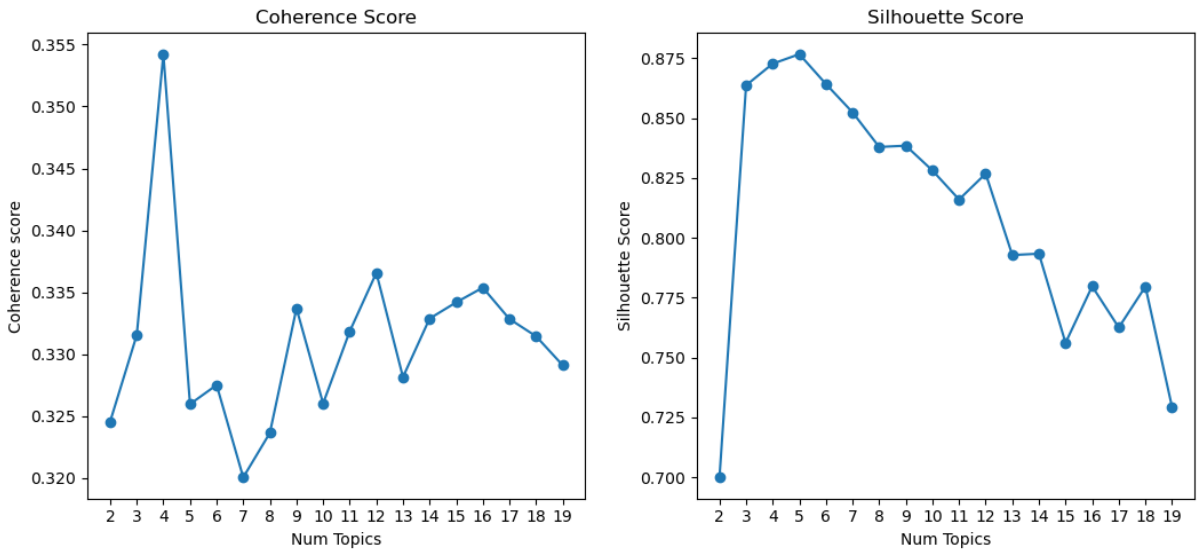

Figure 2: **Coherence and Silhouette scores** for a different number of clusters for the LDAs on noun chunks after TF-IDF. These plots were used to choose a proper number of topics, which in this case is equal to four.

clusters, to encourage more varied approaches.

5.2 Fake News Detection Comparison

5.2.1 Proof of Concept

As described in Section 4.2 we began our work by comparing encoders and different feature selection methods. We trained SVM and Passive Aggressive Classifier on data containing two classes and estimated the performance on validation data. We measured accuracy, recall, f1, and time of training and gathered the results in Table 4.

Encoder	Model	Accuracy	Recall	F1	time
CV	K-means	0.75	**0.99**	0.80	58
CV	GeFeS	0.67	**0.99**	0.75	195
CV	Full	0.86	0.95	0.87	**2**
HV	K-means	0.64	0.95	0.72	402
HV	GeFeS	0.49	0.97	0.65	3888
HV	**Full**	**0.93**	0.95	**0.92**	**2**
TF-IDF	K-means	0.54	0.59	0.56	26
TF-IDF	GeFeS	0.62	0.74	0.66	222
TF-IDF	Full	0.91	0.96	0.91	3

Table 4: **The results from the SVM model.** In the Encoder columns, CV means Count Vectorizer, HV - Hashing Vectorizer, and TF-IDF - TF-IDF Transformer.

The most important outcome is that all feature selection methods negatively impact the model's performance. Not only did model metrics decrease, but also feature selection alone took significantly more time to calculate than for the entire model to train. Overall Passive Aggressive Classifier with Hashing Vectorizer proved to be the most successful according to most metrics. We then used this model on the entire dataset containing all 12 classes. We were able to achieve quite good results with an accuracy of 0.613. Overall results seem great, but before setting a random state we encountered models that had significantly lower accuracy scores, which made this method unreliable.

5.2.2 Global vs Local Methods

Global

For our main method of fake news detection, we decided to train one SOTA transformer model-DistilBERT. We encoded the data with DistilBERT's pretrained tokenizer, which we first tested using statistical machine learning methods as a baseline. The results are presented below:

- Logistic Regression - Accuracy = 0.598
- Support Vector Machines - Accuracy = 0.518
- Random Forest - Accuracy = 0.491
- Naive Bayes - Accuracy = 0.444

In comparison to the Passive Aggressive Classifier, we achieved consistent results of quite high quality. Overall all the models recognized all classes, as the diagonal of the confusion matrix presented on Figure 4 is always significantly

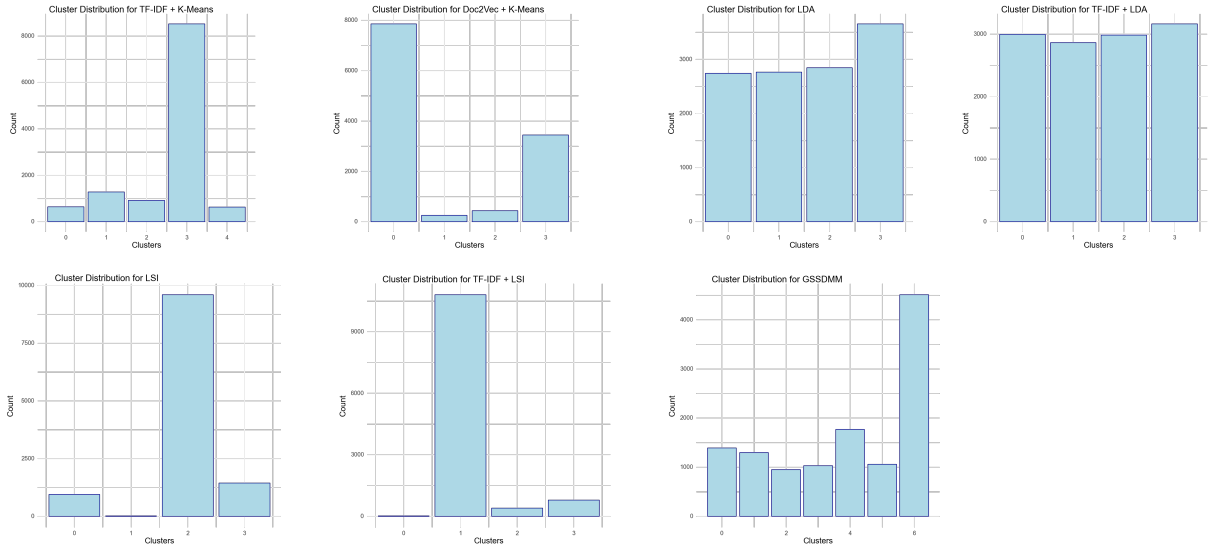

Figure 3: **Clusters comparison** for noun chunks clustering. The bar plots represent the number of documents in each cluster, for 5 different methods. We can see that the most reasonably performing methods, which does not involve a single, huge cluster are LDA, LDA with TF-IDF, and GSSDMM.

darker than the rest of the matrix. We also noted problems with classes *political* and *unknown* as they are the weakest points of all the models. We expected problems with *unknown* classes as it is hard to know what you do not know, but *politics* was surprising. Our models most often confused this class with clickbait, bias, and hate classes, which is also interesting result, as it shows what kind of language is used to describe the political scene of the USA.

After calculating our baseline models we trained DistilBERT on training data. This model achieved accuracy reaching up to 0.71 on the testing dataset. Even though this model achieved better results than baseline models it still struggles with political class (Figure 5). Because this class is so problematic we calculated each model accuracy with this class excluding what greatly boosted our score as shown in Table 5.

Local

Moving on to topic detection in model training, we take clustering achieved using LDA (4 clusters) and GSSDMM (7 clusters). Then we train models for each cluster on the training dataset and evaluate them on each cluster separately on testing data. We measure the model's performance in two ways:

1. We calculate number weighted accuracy between clusters $\left(\frac{\sum_{cluter} cluster\_size * cluster\_accuracy}{\sum_{cluter} cluster\_size}\right)$.

Model	Accuracy	Accuracy*
Logistic Regression	0.581	0.628
SVM	0.518	0.577
Random Forest	0.491	0.540
Naive Bayes	0.444	0.501
DistilBERT	**0.710**	**0.790**

Table 5: **Global approach results.** The accuracy scores for the models trained before the separation into topics. The last column called Accuracy* shows the scores if we remove the most problematic class, being the political one.

2. We analyzed values on diagonals of confusion matrices calculated for each cluster separately. We also calculated accuracy, and accuracy without political class for each cluster.

GSSDMM results

- Logistic regression - Weighted Acc. = 0.54
- SVM - Weighted Acc. = 0.46
- Random Forest - Weighted Acc. = 0.47
- Naive Bayes - Weighted Acc. = 0.44
- DistilBERT - Weighted Acc. = 0.60

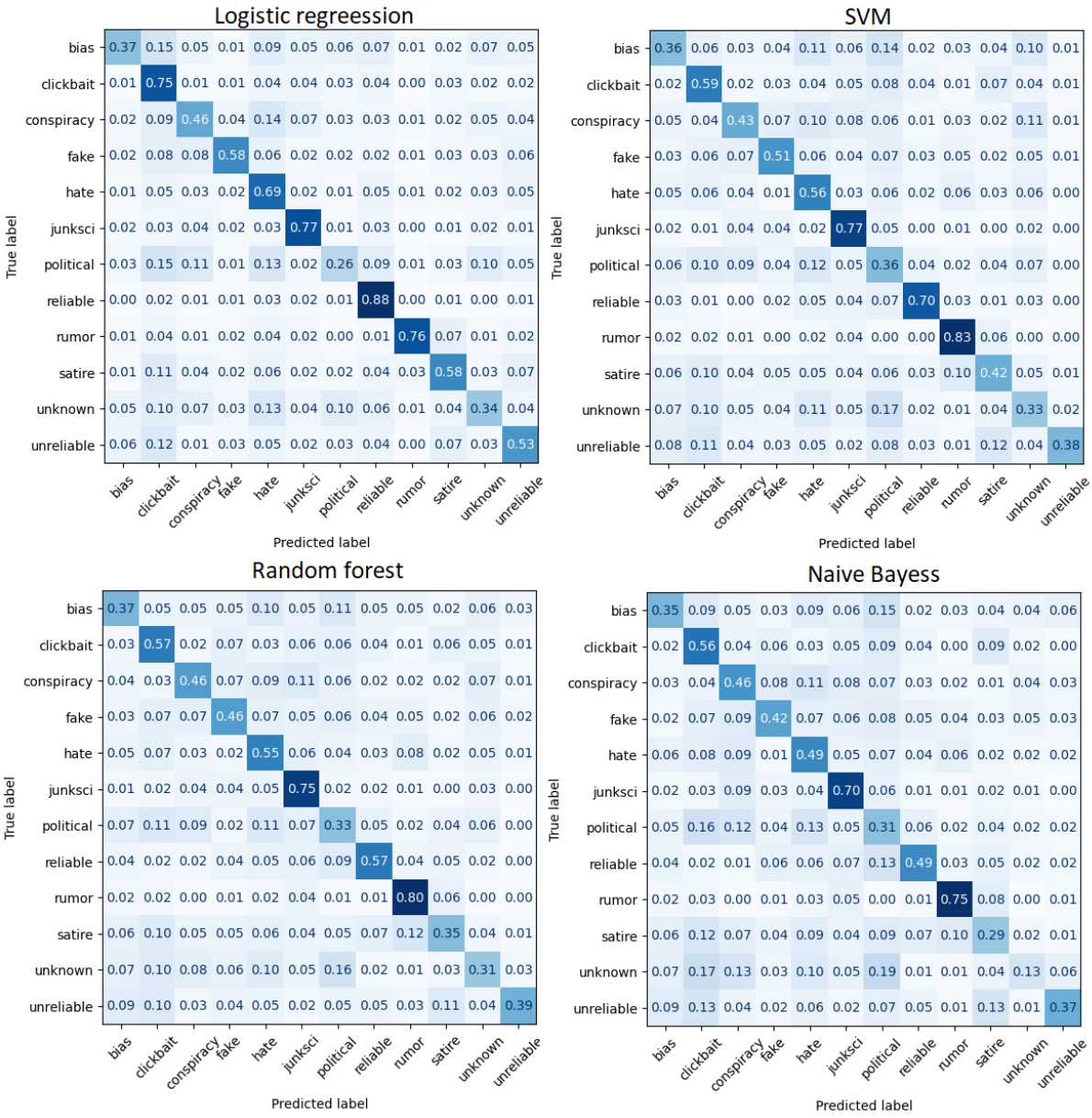

Figure 4: **Confusion matrices** for all models, when they were trained on all topics.

LDA results

- Logistic regression - Weighted Acc. = 0.53
- SVM - Weighted Acc. = 0.48
- Random Forest - Weighted Acc. = 0.46
- Naive Bayes - Weighted Acc. = 0.45
- DistilBERT - Weighted Acc. = 0.62

Comparing the weighted accuracy of models trained on separate topics to the accuracy of models trained on the entire data its easy to conclude that the proposed approach generates models of worse quality. Again DistilBERT was the best-performing model in all examined scenarios. For that reason, we choose those models to further analyze their behavior on separate clusters. We focused on values on the diagonal of confusion matrices, which comprise the first 12 columns of Table 6 and Table 7.

Topics based models not only performed worse when it comes to overall accuracy, but also some models didn't learn all target classes, what is evident on 6, where each model except the last one hasn't learnt all the classes and has zeros on diagonal of confusion matrix. Models based on LDA clustering performed better in this regard, but they

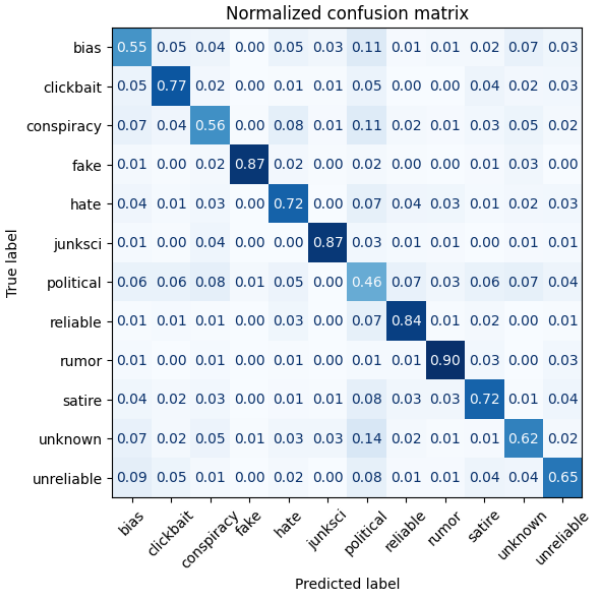

Figure 5: **DistilBERT's confusion matrix** when the model was trained on all topics.

Topic	bias	clickbait	conspiracy	fake	hate	junksci	political	reliable	rumor	satire	unknown	unreliable	**accuracy**	**accuracy***	size
1	0.53	0.70	0.52	0.92	0.71	0.78	0.52	0.60	0.87	0.54	0.49	0.63	0.65	0.70	2995
2	0.37	0.74	0.61	0.89	0.55	0.80	0.17	0.72	0.82	0.71	0.57	0.47	0.63	0.69	2862
3	0.49	0.64	0.56	0.82	0.57	0.80	0.34	0.65	0.76	0.70	0.43	0.46	0.59	0.67	2981
4	0.30	0.79	0.33	0.88	0.74	0.91	0.54	0.78	0.78	0.37	0.48	0.42	0.61	0.71	3162

Table 6: **DistilBERT trained on GSSDMM topics.** The results for clusters produced with DistilBERT on GSSDMM clustering. The first columns provide us with the proportion of correct hits for a given class in the selected cluster (the diagonal from the confusion matrix). The bold ones adhere to the accuracy calculated for the whole cluster, and the accuracy* column translates to the accuracy calculated without political class.

also had lower values on diagonals of confusion matrices compared to model trained on the entire dataset. This approach also didn't help with the problematic class *political*, as accuracy without this class was better on all clusters.

5.3 Explanations

In the following experiments, we compared a sample of articles for each class for each of the created clusters by the two considered methods. Figures 6, 7 illustrate the rate of rejected Wilcoxon tests at the confidence level of $\alpha = 0.05$ with FDR correction. The results prove that the LDA clustering method caused less change in the explainability than GSSDMM. In the case of the latter method, there are cases where more than 90% of the tests were rejected, whereas, in the case of LDA, there are two cells with a rejection rate as low as 7%.

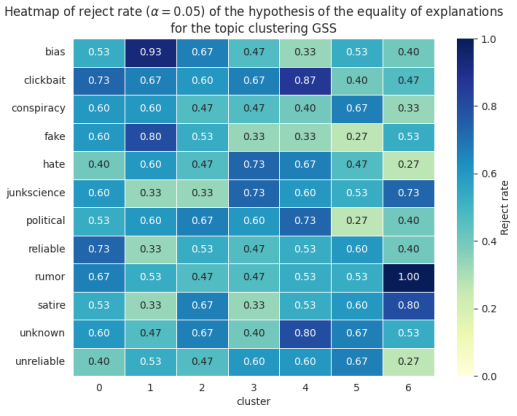

Figure 6: Rate of rejected tests for clustering created by GSSDMM clustering.

6 Discussion and Further Works

Presented results provide us with interesting insights concerning the global, and local fake news

Topic	bias	clickbait	conspiracy	fake	hate	junksci	political	reliable	rumor	satire	unknown	unreliable	**accuracy**	**accuracy***	size
1	0.62	0.33	0.26	0.88	0.47	0.00	0.00	0.41	0.94	0.67	0.46	0.29	0.52	0.56	1393
2	0.00	0.42	0.55	0.97	0.67	0.97	0.60	0.62	0.75	0.27	0.11	0.00	0.57	0.64	1296
3	0.76	0.14	0.00	0.89	0.00	0.00	0.00	0.78	0.95	0.69	0.00	0.00	0.58	0.59	951
4	0.00	0.86	0.27	0.76	0.17	0.89	0.06	0.61	0.70	0.73	0.00	0.00	0.55	0.60	1028
5	0.22	0.76	0.10	0.81	0.50	1.00	0.81	0.05	0.00	0.52	0.55	0.17	0.53	0.67	1766
6	0.15	0.77	0.65	0.67	0.50	0.20	0.00	0.90	0.77	0.69	0.00	0.00	0.57	0.61	1057
7	0.45	0.68	0.47	0.94	0.76	0.85	0.38	0.73	0.64	0.58	0.60	0.79	0.67	0.75	4509

Table 7: **DistilBERT trained on LDA topics.** The results for clusters produced with DistilBERT on LDA clustering. The first columns provide us with the proportion of correct hits for a given class in the selected cluster (the diagonal from the confusion matrix). The bold ones adhere to the accuracy calculated for the whole cluster, and the accuracy* column translates to the accuracy calculated without political class.

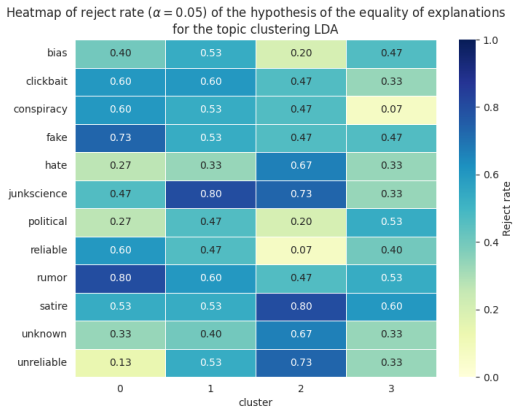

Figure 7: Rate of rejected tests for clustering created by LDA clustering.

detection methodology, so in this section, we will further discuss them.

The most important results to discuss, are the ones of local methods, which quite surprisingly for us were significantly worse than the ones of the global models. The gap between 71% and 60% for the best models is undoubtedly large, and it compromises the quality of the proposed framework. Although it would seem that this approach is unfeasible, we should try explaining the outcoming results. We should bear in mind, that in fact, the global models used a 12 times larger dataset, namely 12,000 observations, whereas the ones for local models had only 1/4th or even 1/7th of it. Additionally, the task of detecting 12 different labels, was hard even on the global dataset, which only gets worse for the local approach. We can suspect that some classes were heavily underrepresented in particular clusters, which additionally negatively impacts the final outcomes.

We advocate for further evaluation of this approach with the usage of larger computational resources. Unfortunately, in our case, we had to limit ourselves to those 12,000 texts, due to the lack of immense computing power. Feasible steps, to better evaluate this approach could involve enlarging the dataset twice and limiting the number of labels to 4 of them, e.g. reliable, fake, rumor, and junk science, or even into a binary classification regarding reliable, and fake classes only. This way we could evaluate the proposed solution in different environments and further assure its applicability.

7 Conclusion

In this work we proposed a novel framework for fake news detection, which incorporates topic, and fake news detection methods with XAI. For the clustering phase, we tested multiple approaches, and used the best of them in next steps. Additionally, we evaluated multitude of fake news detection models, and outlined the best performing one. Eventually, we compared the global, and local fake news detection methods, showing, that in our case the local methodology did not work, possibly due to too small amount of observations. In this matter we advocate for further research, and propose solutions which might improve the studies results. Finally, the models were also compared based on the explainability technique, called Integrated Gradients. This approach proved that topic-specific models put focus on different words than a global model. Moreover, the Wilcoxon test results showed that the differences are real.

References

A.B., A., Kumar, S. M., and Chacko, A. M. (2023). A systematic survey on explainable ai applied to fake news detection. *Engineering Applications of Artificial Intelligence*, 122:106087.

Alarab, I. and Prakoonwit, S. (2022). Effect of data resampling on feature importance in imbalanced blockchain data: comparison studies of resampling techniques. *Data Science and Management*, 5(2):66–76.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165 – 1188.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Caliński, T. and JA, H. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Evgeniou, T. and Pontil, M. (2001). Support vector machines: Theory and applications. volume 2049, pages 249–257.

Guo, G., Wang, H., Bell, D., and Bi, Y. (2004). Knn model-based approach in classification.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Jin, X. and Han, J. (2010). *K-Means Clustering*, pages 563–564. Springer US, Boston, MA.

Kasra Majbouri Yazdi, Adel Majbouri Yazdi, S. K. J. H. W. Z. S. S. (2020). Improving fake news detection using k-means and support vector machine approaches.

Kontostathis, A. (2007). Essential dimensions of latent semantic indexing (lsi). In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 73–73.

Kurasinski, L. and Mihailescu, R.-C. (2020). Towards machine learning explainability in text classification for fake news detection. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 775–781.

Lau, J. H., Collier, N., and Baldwin, T. (2012). On-line trend analysis with topic models: #twitter trends detection topic model online.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Leo, V. D., Puliga, M., Bardazzi, M., Capriotti, F., Filetti, A., and Chessa, A. (2023). Topic detection with recursive consensus clustering and semantic enrichment. *Palgrave Communications*, 10(1):1–10.

Lossio-Ventura, J. A., Morzan, J., Alatrista-Salas, H., Hernandez-Boussard, T., and Bian, J. (2019). Clustering and topic modeling over tweets: A comparison over a health dataset. *Proceedings. IEEE International Conference on Bioinformatics and Biomedicine*, 2019:1544–1547.

Mitra, T. and Gilbert, E. (2021). Credbank: A large-scale social media corpus with associated credibility annotations. *Proceedings of the International AAAI Conference on Web and Social Media*, 9(1):258–267.

Pathak, A. and Srihari, R. (2019). BREAKING! presenting fake news corpus for automated fact checking. In Alva-Manchego, F., Choi, E., and Khashabi, D., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Work-*

shop, pages 357–362, Florence, Italy. Association for Computational Linguistics.

Rousseeuw, P. (1987). Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sahebi, G., Movahedi, P., Ebrahimi, M., Pahikkala, T., Plosila, J., and Tenhunen, H. (2020). Gefes: A generalized wrapper feature selection approach for optimizing classification performance. *Computers in Biology and Medicine*, 125:103974.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Stando, A., Cavus, M., and Biecek, P. (2023). The effect of balancing methods on model behavior in imbalanced classification problems. *arXiv preprint arXiv:2307.00157*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Tian, Z. and Baskiyar, S. (2021). Fake news detection using machine learning with feature selection. pages 1–6.

Webb, G. I. (2010). *Naïve Bayes*, pages 713–714. Springer US, Boston, MA.

Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 178–185, New York, NY, USA. Association for Computing Machinery.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

Williams, J. and Santia, G. (2018). Buzzface: A news veracity dataset withfacebook user commentary and egos.

Yin, J. and Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 233–242, New York, NY, USA. Association for Computing Machinery.

A EDA

To better know our dataset we decided to conduct an Exploratory Data Analysis. Firstly, we analyzed the word counts of given texts and compared them between all 12 classes. As we can see from Figure 10, text length distributions differ from each other, although not too much. We can see that the distribution of reliable articles is slightly different from the global distribution. Quite interestingly it is very close to bias, and political texts. We can see that major differences occur for hate, junk science, fake, and unreliable articles. Finally, let's notice that the unknown class follows the global distribution of word counts.

In the case of word density, we can see that the global distribution follows normal distribution as presented in Figure 11 and that the reliable class is similar to the general trend. This time the most similar to reliable class are hate, political, and fake articles, whereas the biggest differences we can see from bias, conspiracy, junk science, and satire. As before, the unknown class follows the global distribution. Considering those two plots we might want to combine some classes, to simplify the final task. It seems like unknown, political, and reliable classes are fairly similar to each other in this case.

Later we analyzed the sentiment in terms of polarity and subjectivity, presented in Figure 12, and Figure 13 respectively. Interestingly, the reliable sources have one of the highest median polarity values, very similar to the fake news, whereas the categories like bias, or hate are closer to 0. It is much different in terms of subjectivity, where reliable news sources have almost the lowest scores, despite the bias category. All in all, the sentiment analyses might not be so useful as the scores are fairly similar among the groups, and do not necessarily include reliable values.

Named entities presented in Figure 8 can clearly show us the big players in the debate from our dataset. It is mostly dominated by the American government (14/30), and the rest represent other countries. Additionally, it shows that our data source is a heavily politically driven dataset, as it was scrapped around 2017 when the presidential elections happened in the USA.

The noun chunks presented in Figure 9, further outline the topics regarding the USA political scene, however, they additionally give us more insight into different topics, such as the impact of

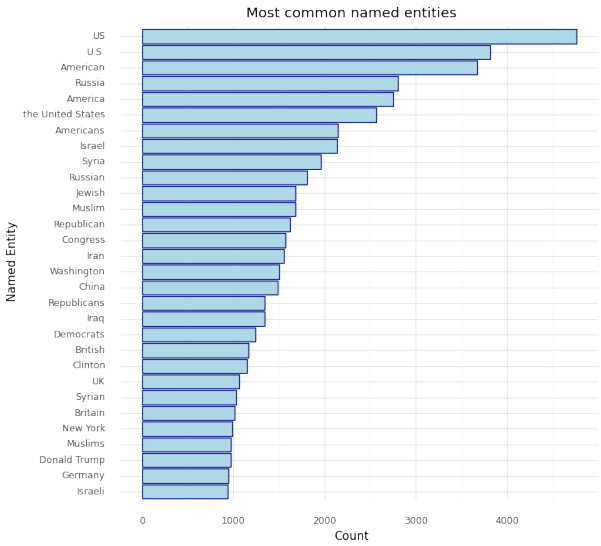

Figure 8: **Top 30 most frequently occurring named entities.**

social media, World War II, climate changes, Wall Street, the importance of free speech or issues regarding the police officers. Unfortunately, they are still suppressed by the most common political scene.

The final visualization from our EDA, presented in Figure 14, compares the top tf-idf terms for each type of news article. This way we can see that all groups were focused on the presidential elections, however, some of them had particular interest areas. Satirical texts focused mostly on Donald Trump, conspiracy theories definitely suspected that the voting was illegal, whereas the fake news focused mostly on Russia.

B Work Division

In the Table 8 we present a workload distributed among all team members throughout the project. Additionally we state, that all of us declare that the workload distribution was fair, reasonable, and everyone contributed as he was supposed to.

Name	Task	Time
Hubert Ruczyński	Topic detection literature review	4 hours
	Project idea, and proposed solution design	2 hours
	Topic 15 review (1st review)	1 hour
	Editor of 1st presentation	3 hours
	Data preprocessing and preparation	12 hours
	Exploratory Data Analysis	6 hours
	Editor of final presentation	8 hours
	Editor of final report	12 hours
Bartosz Siński	Explainability literature review	4 hours
	Topic 15 review (1st review)	1 hour
	Topic detection code	16 hours
	Topic detection section in report	2 hours
	Topic detection section in presentation	2 hours
Maciej Pawlikowski	Fake News Detection literature review	4 hours
	Topic 10 review (1st review)	1 hour
	Methods Review (PoC)	2 hours
	Fake news detection code	20 hours
	Fake news detection section in report	2 hours
	Fake news detection section in presentation	2 hours
Adrian Stańdo	Editor of PoC	4 hours
	Dataset description	3 hours
	Topic 10 review (1st review)	1 hour
	Explainable AI code	10 hours
	Explainable AI section in report	2 hours
	Explainable AI section in presentation	2 hours
	Repository structure	2 hours

Table 8: Work division regarding this document and additional deliverables.

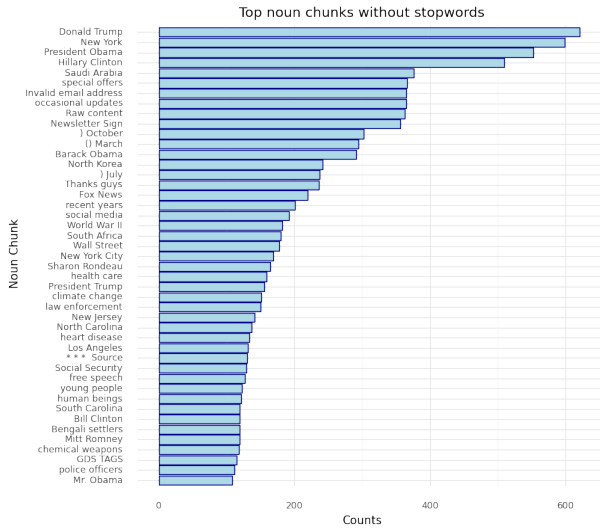

Figure 9: **Top 40 most frequently occurring noun chunks, with removed stop-words.**

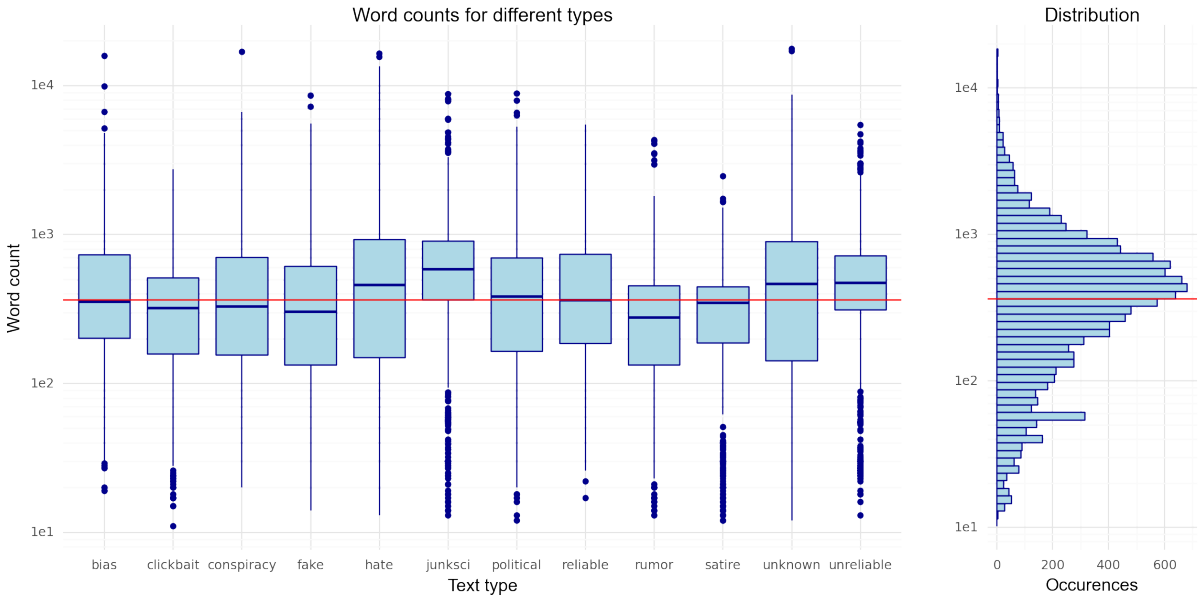

Figure 10: **Word Count analysis** The distribution of text word counts for different article types compared to the global distribution of all news on the right. The red line indicates the median for the reliable class.

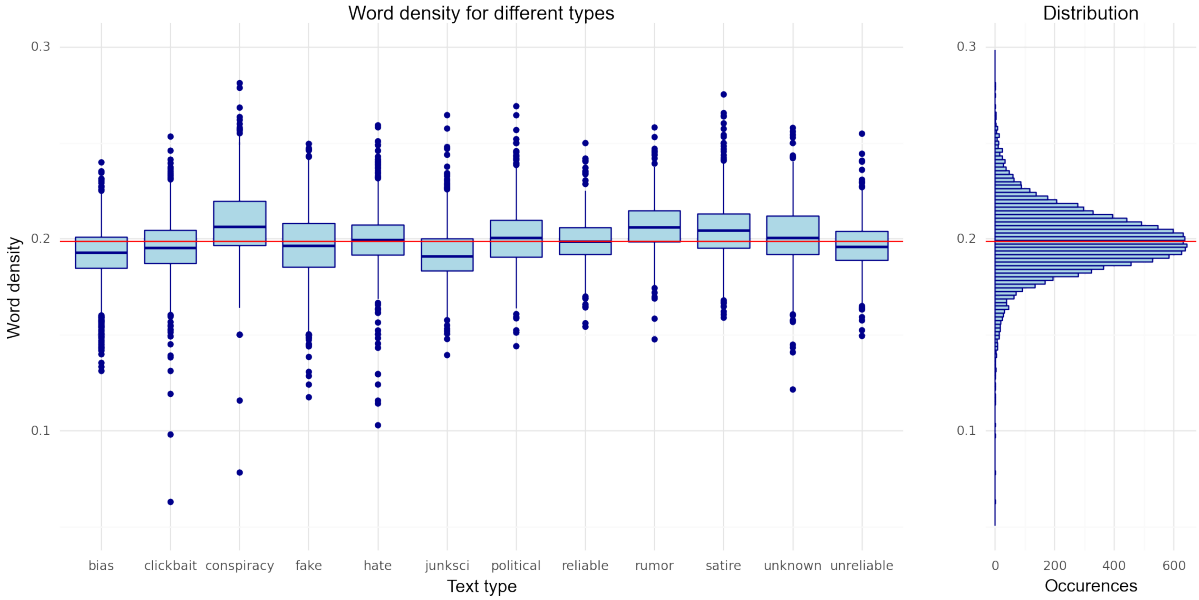

Figure 11: **Word Density analysis** The distribution of text word density for different article types compared to the global distribution of all news on the right. The red line indicates the median for the reliable class.

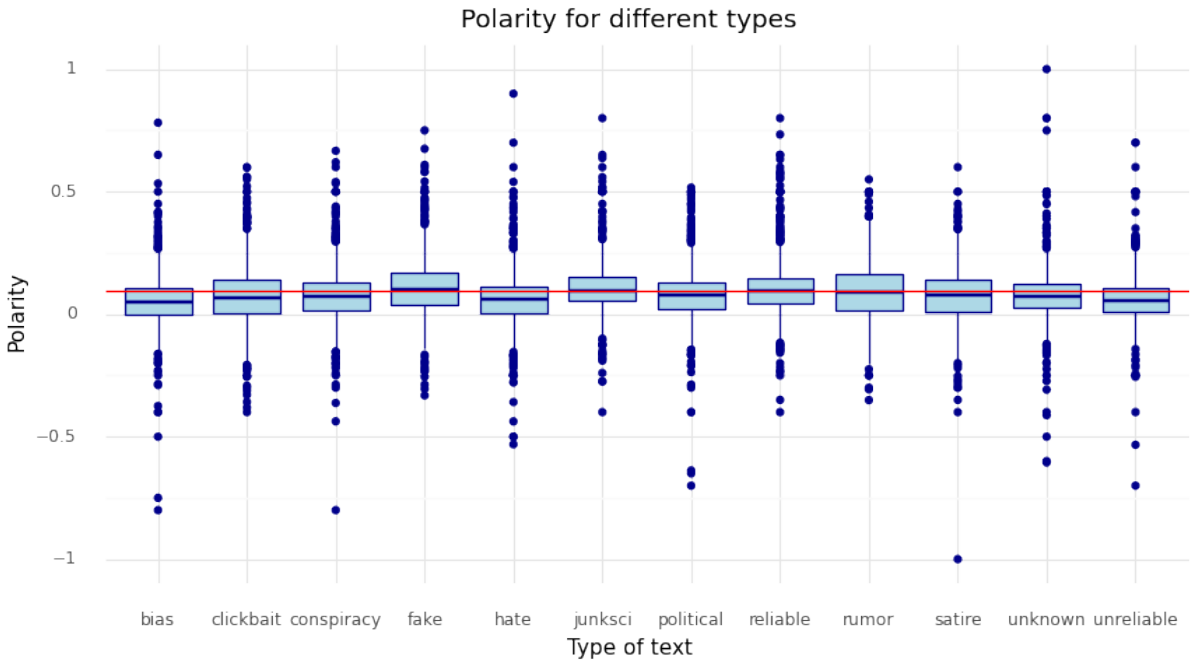

Figure 12: **Sentiment Polarity analysis** The distribution of text polarity for different article types. The red line indicates the median for the reliable class.

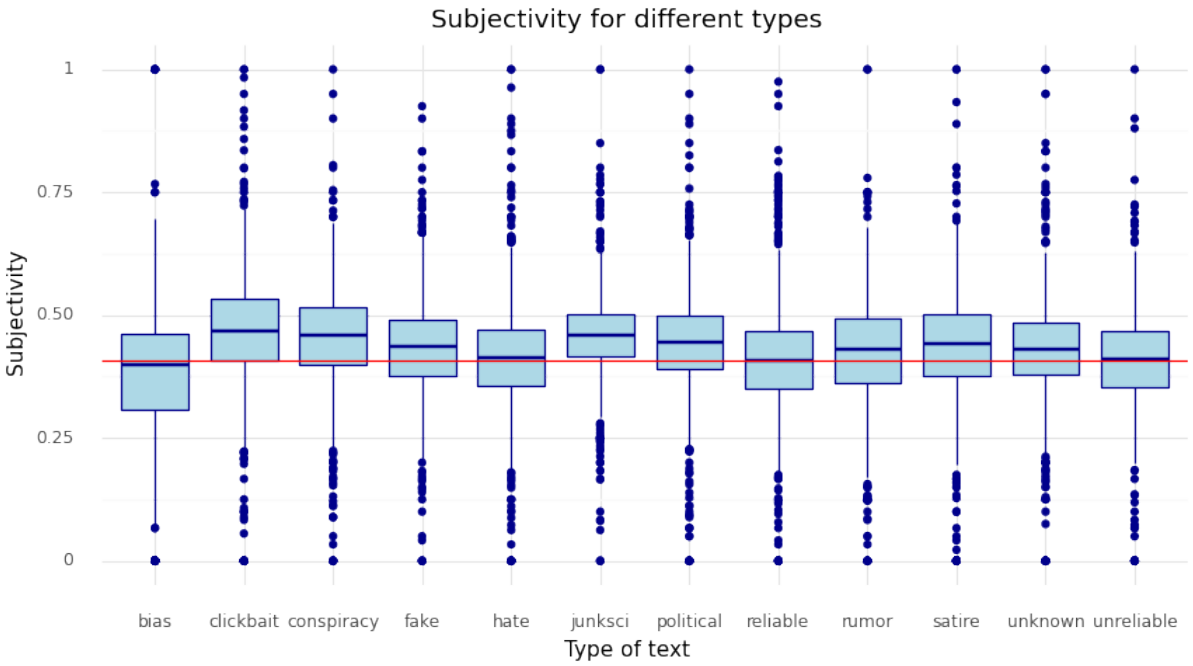

Figure 13: **Sentiment Subjectivity analysis** The distribution of text subjectivity for different article types. The red line indicates the median for the reliable class.

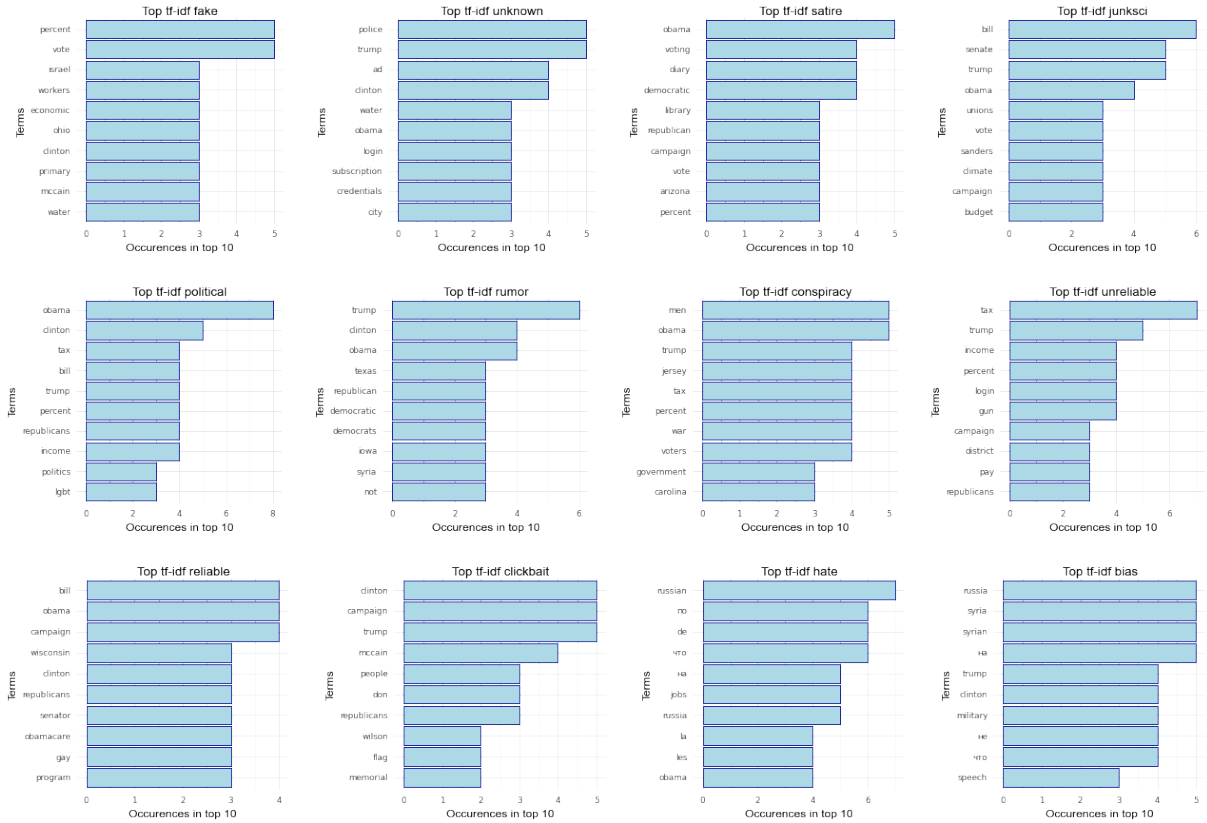

Figure 14: **Conditional TF-IDF analysis** The set of plots, representing the top 10 terms according to tf-idf for different article types.