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 (NLP) 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 modelling and then train and evaluate the models on smaller topics, which lets the models be fine-tuned for a particular discussion. In our work, we propose the `local` approach and compare the results to the `global` method, where we perform topic detection tasks for the whole dataset. The results show that `local` models turned out to be worse than their counterparts, however, we advocate for further studies in this domain due to the limitations encountered by our team.

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 just to 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 a given dataset in order to train later 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 an approach by comparing those `local` models (trained on a topic) to `global` solutions (trained on all data) and analyze them with the usage of eXplainable AI (XAI) methods.

2 Related Works

To better understand the topic of our study and choose appropriate methods for each part of the proposed pipeline, we had to conduct a thorough study of state-of-the-art (SOTA) solutions from the literature. The main goal of that was the acknowledgement of how to do something and which tools can help us achieve promising results. In this section, we describe the most important scientific works, which inspired us during the development of this study.

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 selecting 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 light assumption. However, it also indicates some drawbacks, as a singular tweet corpus is rather small, containing, on average, 60 words that do not have to be unique, 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 the short messages, 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 (D'Ulizia et al., 2021; Zhou and Zafarani, 2020). 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 are calculated and presents the value of the proposed feature selection method combined with SVM (Evgeniou and Pontil, 2001) to achieve very good results regarding 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 neighbours. 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 2,282 news articles and posts about the 2016 election from Facebook, which was divided into several categories: mostly fake, fake, mostly true, true, and mixed.

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 colour-coded text to show the impact of the particular words on the prediction. For the detection task, the 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 strongly correlated with clickbait 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 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 description of Figure 1 (described in Section 4). Those datasets are later used during the topic detection phase, where we test the multitude of clustering approaches (described in Section 3.1), evaluate them, and two of the best clusterings are forwarded to the next phase, being fake news detection. At this point, we are training the models on the whole dataset and choosing the best ones, which is called a `global` approach. After that, the best models 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 by comparing the performance of `global` and `local` approaches. Finally, the models are explained with the usage of XAI methods to provide some insights and outline the differences between the most important aspects of both approaches.

3.1 Topic Detection

Topic detection was performed separately on both lemmas and noun chunks extracted from the documents in the chosen dataset to later compare the two approaches. Firstly we have used the K-means (Jin and Han, 2010) algorithm with TD-IDF data representation and Doc2Vec (Le and Mikolov, 2014) to perform the standard clustering, which we treat as a baseline. Subsequently, we have used the topic detection models. We have chosen three well-established methods: 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 modelling, 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 (CHI) (equations in Appendix B.1) and Silhouette Coefficient (SC) (equations in Appendix B.2) were utilized along with the Coherence Score, which is specific to LDA and LSI algorithms. For k-means clustering, CHI and SC were calculated on the TF-IDF and Doc2Vec representation with cluster labels assigned to each row. However, for the topic modelling methods, after performing topic detection, we create a feature vector for each document. Within this vector, each element represents the probability of the observation being associated with a particular topic. Then SC and CHI are calculated on those obtained vectors. When determining the number of clusters for K-means, we have also examined the within-cluster sum of square distances metric. We examine the results for a number of clusters between 2 and 20. After obtaining a correct number of clusters for each of the clustering methods, we have compared their results once again using the CHI and SC. Two best clustering were passed to group data for `local` fake news detection.

3.2 Fake News Detection

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 (Generalized wrapper-based Feature Selection) (Sahebi et al., 2020).

Eventually, all clustering methods only worsened the models' performance and were computationally expensive. The Passive Aggressive Clas-

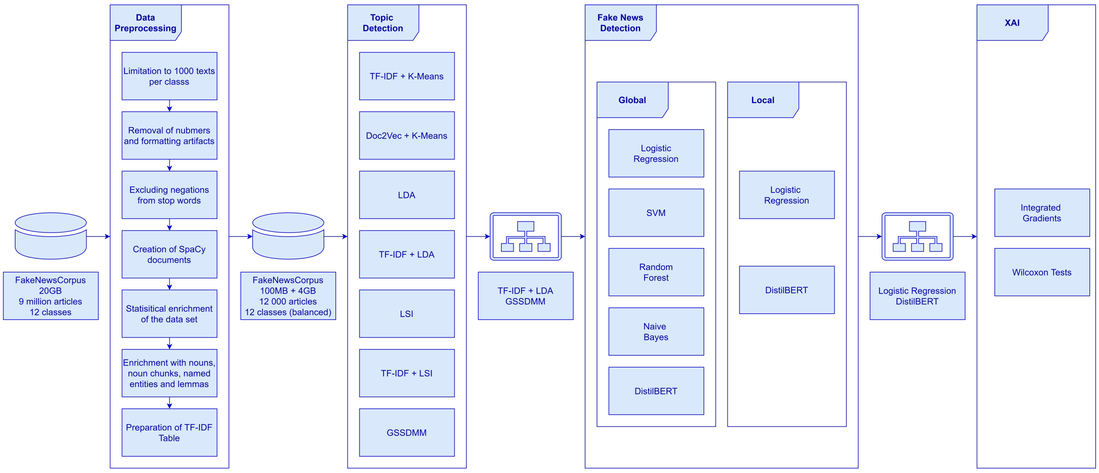

Figure 1: **Experiments pipeline.** The graph represents the pipeline of experiments conducted in this paper. It goes from the initial dataset, through data preprocessing, topic detection, and fake news detection, to XAI outcomes.

sifier 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.

3.3 Explainability

In the project, we applied an explainability method to the DistilBERT model called Integrated Gradients (IG) proposed 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.

3.3.1 Integrated Gradients

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 pad tokens in the case of neural networks for language processing), x is the input to the model (for which the explanation is to be calculated), and i is the dimension of the input.

The intuition behind this method is that firstly the vectors are interpolated between the baseline and input vector - this is represented by the $(x' + \alpha \times (x - x'))$ term. In the next step, the gradients of the output predictions of the model F, with respect to the input feature, are calculated - this is represented by the formula $\frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i}$. Finally, the gradients are summed over the interpolated vectors (what is represented by the integral) and scaled by the $(x_i - x_i')$ formula.

One important feature of the IG method is *completeness*. This means that attributions for all input dimensions sum up to the difference between the model output for input vector x and the baseline vector x'.

In this project, the above parameters can be understood as follows: F is the trained DistilBERT model, x' is a vector of pad tokens (special tokens which ensure that shorter sequences will have the same length as either the longest sequence in a batch or the maximum length accepted by the model), x is the input text, and i is the index of a token in the input text.

The IG method assigns to each token a value called attribution. If this value is positive, it indicates a positive influence on the prediction, on the other hand, negative values show a negative impact on the model estimate. The absolute value of the attribution determines the power of the given token on the final prediction.

3.3.2 Wilcoxon statistical test

The Wilcoxon signed-rank test (Wilcoxon, 1992) is a non-parametric statistical test used to assess whether there is a significant difference between two paired groups. In other words, it tests the null hypothesis that two related paired samples come from the same distribution. It is an alternative to the paired t-test when the assumptions of normality are not met.

The test statistic is calculated using the differences in scores between each pair of observations. In the next step, the absolute differences are ranked, and later the sums of positive and negative differences are considered to calculate the final test statistic.

One common use case of the test is to compare whether, e.g., there was a change in the results of exams after and before a certain training. For example, assume there is a group of students who wrote a certain exam. After training, the same group of students retook the same exam. In such a situation, there are two groups of observations (before and after the training) which are paired (the results are paired by a student), and the Wilcoxon statistical test can be used to compare the results. The output of the test can say whether the number of achieved points by students has changed after the training.

3.3.3 Comparison of explanations

In the project, the explanations created by the two models (global and local) will be compared with each other. This will be done by comparing the attribution values for the same input tokens. The verdict on whether the explanations differ from each other will be provided by the Wilcoxon statistical test.

Referring back to the example with the results of exams, we assumed that the students are represented by tokens in the input text and observations in two groups, after and before training, are depicted by having a global and local model for fake news detection.

In the approach presented in the project, the attribution values for the input tokens are assumed to come from a certain, unknown probability distribution. The aim of the usage of the Wilcoxon statistical test is to see whether the explanations created by the two models differ from each other, or in other words, whether the distribution of the attribution scores is different. This is done by calculating the p-value and rejecting the null hypothesis if it is smaller than, e.g., 0.05.

The described above approach was proposed in (Alarab and Prakoonwit, 2022; Stando et 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)).

4 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 expertise. 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.

4.1 Existing Resources

1. *BuzzFeedNews* - The dataset used in the paper (Tian and Baskiyar, 2021) contains data on news published on Facebook 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 9,000,000 news articles with 12 assigned classes, which weigh around 20GB. It contains, among other things, the text of the articles, their title, and source. It also provides labels for 11 different types of misinformation (*fake, satire, conspiracy, hate, etc.*) 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, e.g. described in (D'Ulizia et al., 2021), 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, the *FakeNewsCorpus*, as it is the most extensive one of them. Additionally, the news article content is much longer than in the case of tweets, as the median length oscillates around 400 words per text. It is a very useful feature, as each text is long enough to carry plenty of information and provide a vast text corpus, and thus its characterization by topic or news category is easier. In our study, we will use the annotation information provided by this source as ground truth and the content of crawled websites in order to extract meaningful text features. The exploratory analysis of *FakeNewsCorpus* subset containing 12,000 observations is presented in Appendix I.

4.2 Data Preparation

As the most important data for us are the raw texts included in the dataset, we had to deal with plenty of difficulties. In most cases, the data set contained the results obtained from scrappers, which resulted in lower text quality, as they included page numbers or control characters such as '$\backslash n$'. In spite of that, we had to 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 1,000 records per class. This way, we can ensure that our solution will be able to work properly for each type of news, which, intuitively, might 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 (Honnibal and Montani, 2017), 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 SpaCy pipeline called *en_core_web_md* (further called *en* in this article), 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 artefacts, 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 modelling, 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.

Table 1 presents the results of the aforementioned preprocessing, which resulted in a dataset with 12,000 records. We also saved the top TF-IDF table, used a modification of *en* model, and the documents created with it. As a result of these limitations, we managed to diminish the dataset size from 20GB to only 100MB, which makes the modelling task feasible. To know more about the dataset characteristics, take a look at Appendix I, where we present a short exploratory data analysis (EDA).

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).

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 about the results obtained during the Proof of Concept (Section 5.2.1) to show that feature selection strategies are of no use in our situation. Eventually, we will compare the results of models trained on the whole dataset and the topics provided by the clustering methods from the previous step. In the end, we will analyze the explanations of the best model for both clusterings.

5.1 Topic Detection Comparison

We determined the optimal cluster count by visualizing CHI and SC values for all methods with Within-cluster square distances for K-means algorithms. Additionally, we considered Convergence scores for topic modelling methods. We plotted these metrics for the different cluster counts and searched for the point where both metrics were either highest or had local 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	Cluster Count	Silhouette Coefficient (SC) ↑	Calinski-Harabasz Index (CHI) ↑
TF-IDF + K-means	4	0.04	293
Doc2Vec + K-means	6	0.13	449
LDA	4	0.61	18668
TF-IDF + LDA	**4**	**0.87**	**91490**
LSI	7	-0.32	49
TF-IDF + LSI	3	0.47	1655
GSSDMM	4	0.71	529

Table 2: **Lemma's clustering** The Table presents the SC and CHI values calculated as an evaluation of different clustering algorithms on lemmas only. SC is in the range (-1, 1), while CHI does not have a specified range. ↑ means that a higher score indicates better-defined clusters. .

Clustering Algorithm	Cluster Count	Silhouette Score ↑	Calinski-Harabasz Index ↑
TF-IDF + K-means	5	0.07	314
Doc2Vec + K-means	4	0.39	3473
LDA	4	0.88	110432
TF-IDF + LDA	**4**	**0.94**	**323993**
LSI	4	-0.51	131
TF-IDF + LSI	4	-0.29	393
GSSDMM	7	0.87	15681

Table 3: **Noun chunks clustering** The Table presents the SC and CHI values calculated as an evaluation of different clustering algorithms on noun chunks only. SC is in the range (-1, 1), while CHI does not have a specified range. ↑ means that a higher score indicates better-defined

We can see that when considering chosen metrics, topic detection performed on the noun chunks gave overall better results compared to the topic detection on lemmatized text. 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.

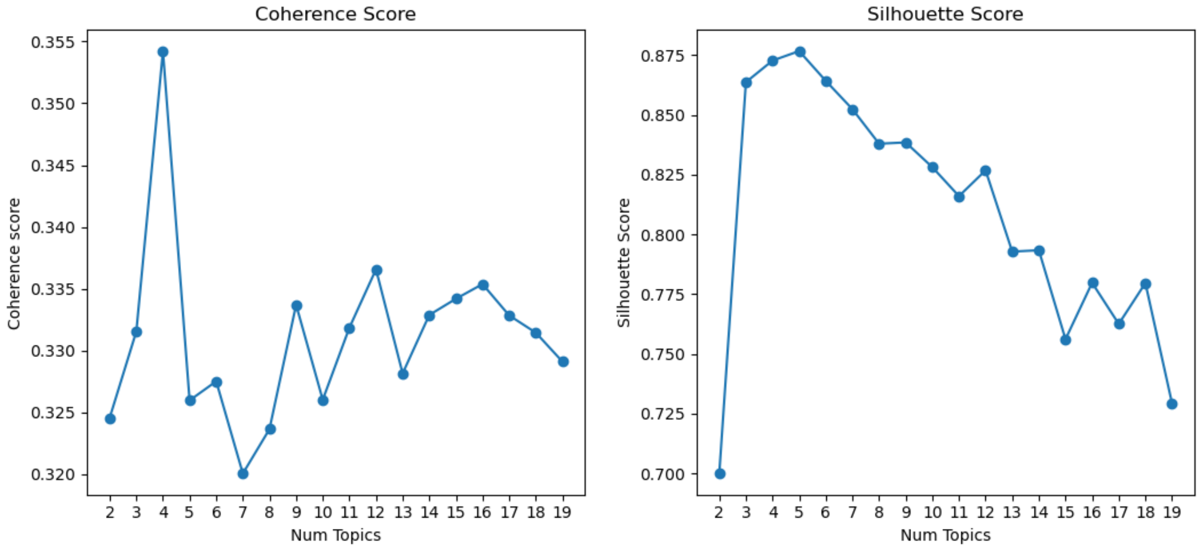

Figure 2: **Coherence and Silhouette scores** for a different number of clusters for the LDAs on noun chunks after TF-IDF. These plots were used to choose a proper number of topics, which in this case is equal to four.

Another algorithm that performed well was the GSSDMM, for both lemmas and noun chunks, it had the 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 clusters to encourage more varied approaches.

5.2 Fake News Detection Comparison

5.2.1 Proof of Concept

As described in Section 3.2 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.

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.

Encoder	Model	Accuracy ↑	Recall ↑	F1 ↑	time [s]
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.

5.2.2 Global vs. local methods

Global methods

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 covers: lowering the case of the entire string, adding special tokens to mark the start and end of the string, splitting uncommon words into several tokens, removing punctuation, padding/ truncating, and embedding the data. We first tested this encoding using statistical machine-learning methods as a baseline.

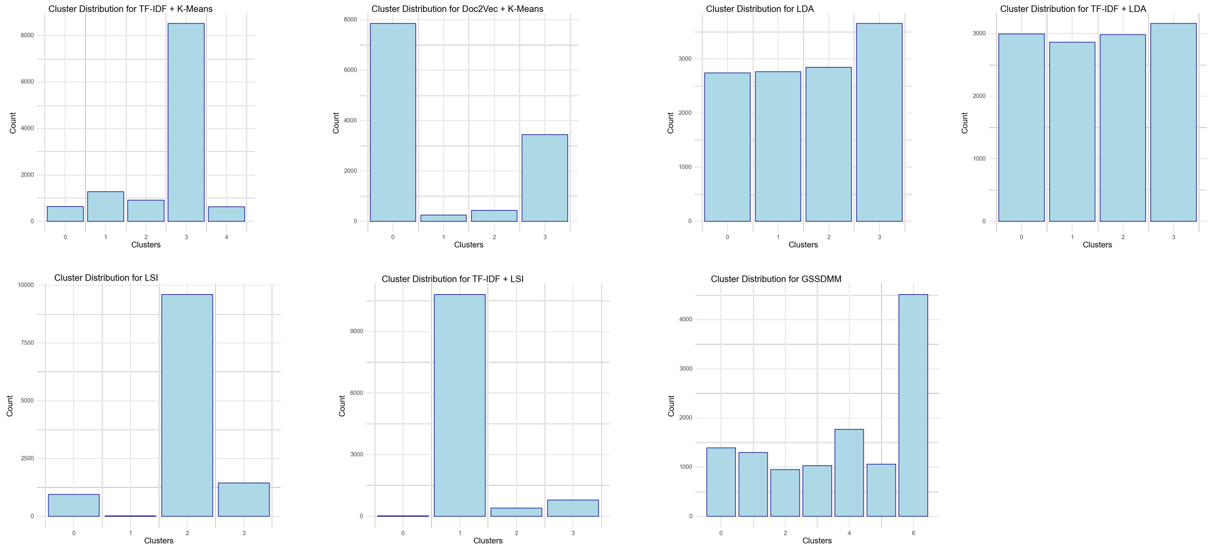

Figure 3: **Clusters comparison** for noun chunks clustering. The bar plots represent the number of documents in each cluster for 5 different methods. We can see that the most reasonably performing methods, which do not involve a single, huge cluster, are LDA, LDA with TF-IDF, and GSSDMM.

Model	Accuracy ↑
Logistic Regression	0.598
SVM	0.518
Random Forest	0.491
Naive Bayes	0.444

Table 5: **Baseline results for DistilBERT tokenizer.** The accuracy scores for the models trained before the separation into topics.

The results are presented below 5. We decided to analyze accuracy as we have balanced classes and want to treat all classes equally.

In comparison to the Passive Aggressive Classifier, we achieved consistent results of quite high quality 5.

Overall, all the models recognized all classes, as the diagonal of the confusion matrix presented in Figure 4 is always significantly 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 an 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 (see Figure 5). Because this class is so problematic, we calculated each model's accuracy with this class, excluding what greatly boosted our score, as shown in Table 6.

Model	Accuracy ↑	Accuracy w/o politics ↑
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 6: **Global approach results.** The accuracy scores for the models trained before the separation into topics. The last column, called Accuracy w/o politics, shows the scores if we remove the most problematic class, the political one.

Local methods

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 the number weighted mean of accuracies achieved on all clusters
   Weighted Accuracy:

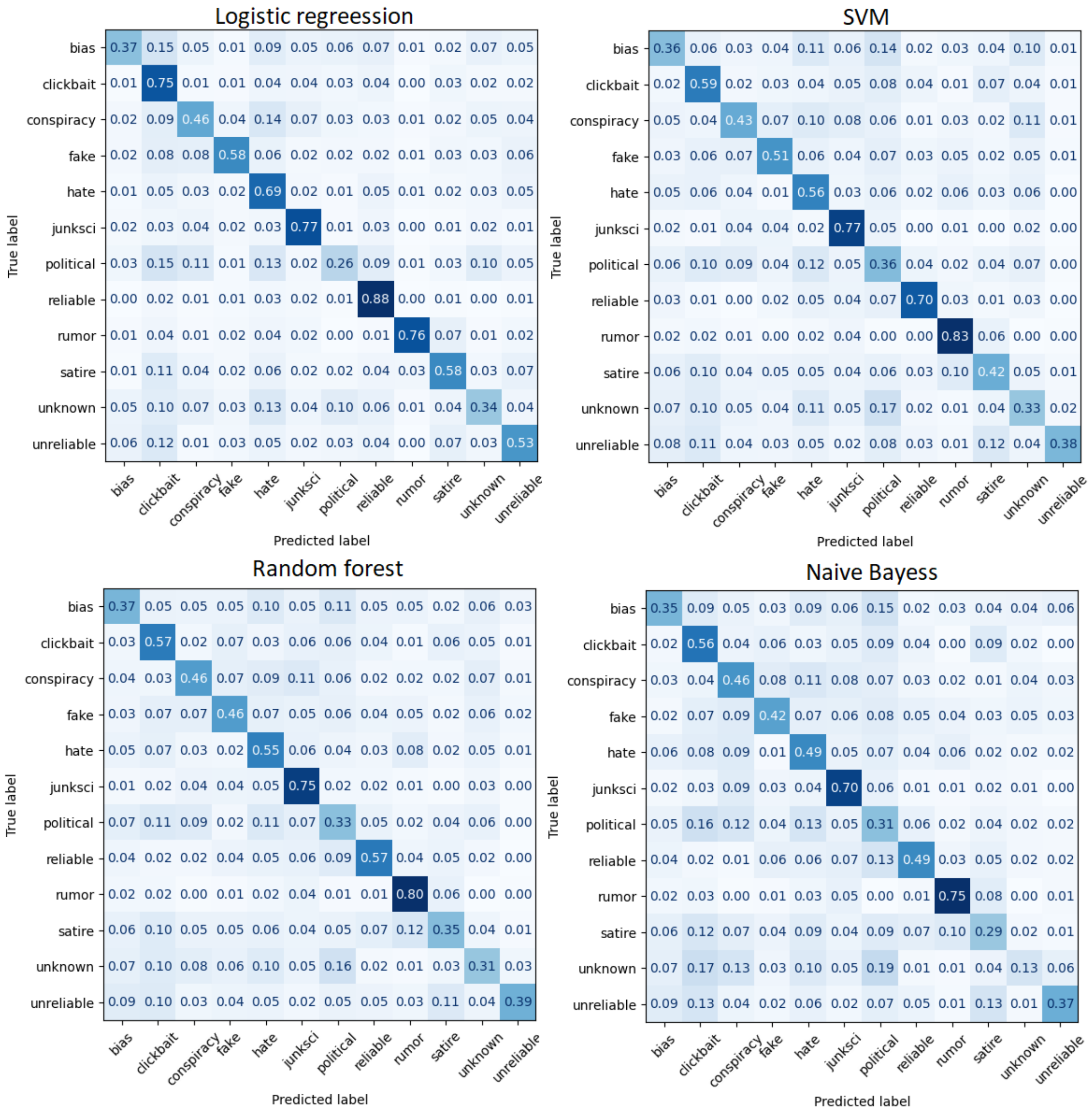

Figure 4: **Confusion matrices** for all models when they were trained on all topics.

$\frac{\sum_{cluter} cluster\_size \cdot cluster\_accuracy}{\sum_{cluter} cluster\_size}$. We choose this metric to more accurately compare global and local approaches.

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

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

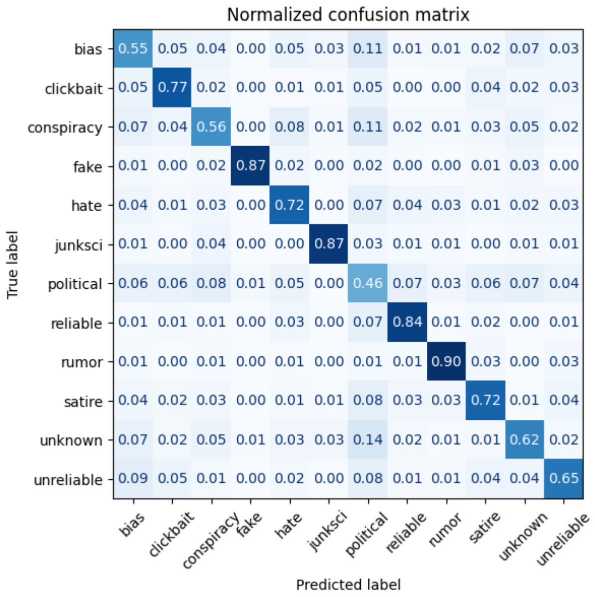

Figure 5: **DistilBERT's confusion matrix** when the model was trained on all topics.

models trained on the entire data we 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 analyze their behaviour on separate clusters further. We focused on values on the diagonal of confusion matrices, which comprise the first 12 columns of Table 7 and Table 8.

Topics-based models not only performed worse when it comes to overall accuracy but also some models did not learn all target classes, which is evident in Table 7, where each model except the last one hasn't learnt all the classes and has zeros on the diagonal of the confusion matrix. Models based on LDA clustering performed better in this regard, but they also had lower values on diagonals of confusion matrices compared to models 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 the explanations created by the global and local models. The comparisons were made using a sample of articles for each class and for each of the created clusters by the two considered methods, namely the GSSDMM and LDA clustering.

As a reminder, the comparison procedure is based on the Wilcoxon statistical test. It rejects the null hypothesis when the estimated p-value is smaller than the adopted confidence level α. In other words, the test rejects the null hypothesis if the two explanations are statistically significantly different from each other. Hence, the perfect and expected result is that the rejection rate in each case is as close to 0 as possible - this would mean that the models provided statistically the same explanations. If it does not hold, it means that each of the two models produces statistically different explanations and probably, puts focus on different tokens while creating the final prediction.

Figures 6 and 7 illustrate the rate of rejected Wilcoxon tests ($rejected\_tests/all\_tests$) at the confidence level of $\alpha = 0.05$ with False Discovery Rate (FDR) correction (as described in the previous section, the correction is needed because of the series of performed statistical tests).

Figures 6 and 7 show that the rejection rates were high in almost all cells. Nevertheless, by comparing the values on the two plots, it can be noticed that the explanations created by the models trained on the GSSDM clusters differ much more from the original ones than the explanations created by models trained on LDA clusters.

Another interesting observation is that in the case of the LDA clustering, there were cells where the rejection rate was as small as 0.07. On the other hand, some clusters in the case of GSSDM clustering had a rejection rate higher than 0.8, which means that 80% of produced explanations

Topic	bias	clickbait	conspiracy	fake	hate	junksci	political	reliable	rumor	satire	unknown	unreliable	**accuracy** ↑	**accuracy w/o politics** ↑	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 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 w/o politics column translates to the accuracy calculated without political class.

Topic	bias	clickbait	conspiracy	fake	hate	junksci	political	reliable	rumor	satire	unknown	unreliable	**accuracy** ↑	**accuracy w/o politics** ↑	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 8: **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 w/o politics column translates to the accuracy calculated without political class.

were different from the ones created by the global model.

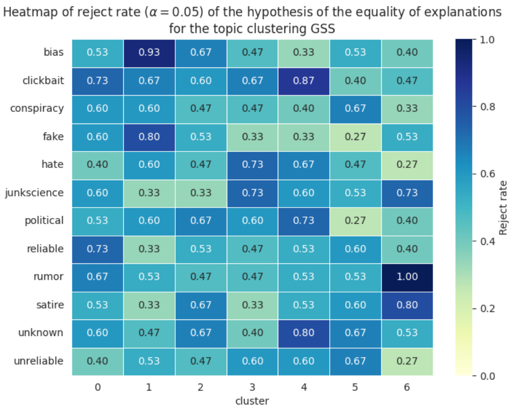

Figure 6: Rate of rejected tests for clustering created by GSSDMM clustering.

6 Discussion

Presented results provide us with interesting insights concerning the `global` and `local` fake news 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 few times larger dataset, namely 12,000 observations, whereas the ones for `local` models had only 1/4th or even 1/7th of it, although all approaches used the same amount of columns. Additionally, the task of detecting 12 different labels was hard even on the `global` dataset, which only gets worse for

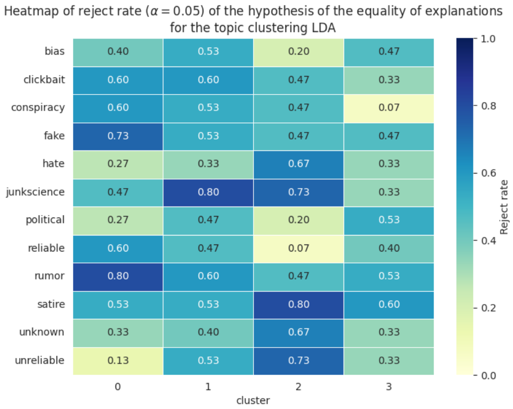

Figure 7: Rate of rejected tests for clustering created by LDA clustering.

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, 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.

As far as the explainability is concerned, we proved that the models trained on the clusters provide different explanations than the global model. Such a situation was expected, as by creating clusters, we removed some variability from each subset of the dataset. Nonetheless, one of the aims of the project was to measure the extent of changes in explanations. The experiments showed that it depends on the class and used clustering algorithm - some produce similar, but many much different explanations than the original model. The answer to the question of which explanations are better or more reliable is out of the scope of the project - nonetheless, such a study can be performed as a future work.

7 Further Works

As the further works, we can see a number of possible paths to improve the obtained results:

- the classification task could be reduced to two or four classes in order to reduce complexity and make it easier to compare local and global methods,
- increasing the size of the training data - because of hardware limitation, we made use of a small subset of available data,
- similarly to the previous point, increasing the number of samples considered in the explainability section,
- developing new methods which aim to compare explanations created by different models in the field of NLP,
- exploring the differences in explanations in more detail by comparing explanations created by the local and global methods visually, assessing the explanations, and choosing the more reliable one,
- studying more advanced topic detection methods,
- studying more advanced text classification methods, e.g. utilising Large Language Models (LLMs),
- transforming the created batch-processing pipeline into an online learning pipeline by making use of, e.g., stream clustering algorithms like TextClust.

8 Conclusion

In this work:

1. We proposed a novel framework for fake news detection, which combines topic and fake news detection methods with XAI.
2. We distinguished the concepts of `global` fake news detection, where we perform the classification based on whole datasets, and the `local` fake news detection, which performs the task on the clusters representing the topics present in the used dataset.
3. We tested and evaluated multiple clustering/topic detection approaches in order to use the best of them in the next steps.
4. We evaluated a multitude of fake news detection models and outlined the best-performing one for the `global` method, being DistilBERT.
5. 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.
6. Finally, the models were also compared based on the explainability technique called Integrated Gradients. This approach proved that topic-specific (`local`) models differ much from a `global` model in terms of the attribution score distribution.

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*.

D'Ulizia, A., Caschera, M. C., Grifoni, P., and Ferri, F. (2021). Fake news detection: A survey of evaluation datasets. *PeerJ Computer Science*, 7:e518.

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

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H. M., Daumé, H., and Crawford, K. (2018). Datasheets for datasets. *Communications of the ACM*, 64:86 – 92.

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.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

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 Workshop*, 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.

Zhou, X. and Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5).

A Work Division

In Table 9, 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 and reasonable, and everyone contributed as he was supposed to.

B Metrics and measures

B.1 Calinski-Harabasz Index

The equations below present the following measures: between-group sum of squares - BGSS (1), within-group sum of squares WGSS (2,3), and Calinski-Harabasz score - CH (4).

$$BGSS = \sum_{k=1}^{K} n_k \times ||C_k - C||^2 \tag{1}$$

$$WGSS_k = \sum_{i=1}^{n_k} ||X_{ik} - C_k||^2 \tag{2}$$

$$WGSS = \sum_{k=1}^{K} WGSS_k \tag{3}$$

$$CH = \frac{BGSS}{WGSS} \times \frac{N - K}{K - 1} \tag{4}$$

The notation is as follows: n_k is the number of observations in cluster k, C_k is the centroid of cluster k, C is the centroid of the dataset (barycenter), K is the number of clusters, X_ik : the $i - th$ observation of cluster k, $WGSS_k$: the within-group sum of squares of cluster k, and N is the total number of observations.

B.2 Silhouette Coefficient

The following equations present the Silhouette Coefficient for i-th observation (5) and the overall score for all clustering (6). Its values range from -1 to 1, where values below 0 represent poorly separated clusters, and the larger the value, the better the separation.

$$SilhouetteScore(i) = \frac{max(a_i, b_i)}{b_i - a_i} \tag{5}$$

$$SilhouetteScore = \frac{\sum_{i=1}^{n} SilhouetteScore(i)}{n} \tag{6}$$

The notation is as follows: a_i is the average distance of i to all other data points in the same cluster, b_i is the average distance of i to all data points in the nearest cluster, and n is the number of clusters.

B.3 Accuracy

The following equations present the accuracy measure:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

where TP states for True Positive, TN - True Negative, FP - False Positive, and FN - False Negative.

B.4 Precision

The following equations present the precision measure:

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

where TP states for True Positive, and FP - False Positive.

B.5 Recall

The following equations present the recall measure:

$$Recall = \frac{TP}{TP + FN}, \tag{9}$$

where TP states for True Positive, and FN - False Negative.

B.6 F1 Score

The following equations present the F1 score measure:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{10}$$

C Trained models

The following table shows all models trained during our research 10.

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
	Topic 15 review (2nd 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	20 hours
Bartosz Siński	Explainability literature review	4 hours
	Topic 15 review (1st review)	1 hour
	Topic 15 code reproduction (2nd review)	1 hour
	Topic detection code	16 hours
	Topic detection section in report	4 hours
	Topic detection section in presentation	2 hours
	Reproducibility checklist	2 hours
Maciej Pawlikowski	Fake News Detection literature review	4 hours
	Topic 10 review (1st review)	1 hour
	Topic 10 code reproduction (2nd review)	1 hour
	Methods Review (PoC)	5 hours
	Fake news detection code (without computation time)	20 hours
	Fake news detection section in report	3 hours
	Fake news detection section in presentation	2 hours
	Model cards	2 hours
Adrian Stańdo	Editor of PoC	4 hours
	Dataset description	3 hours
	Topic 10 review (1st review)	1 hour
	Topic 10 review (2nd review)	1 hour
	Explainable AI code	10 hours
	Explainable AI section in report	6 hours
	Explainable AI section in presentation	2 hours
	Repository structure and maintenance	4 hours

Table 9: Work division regarding this document and additional deliverables.

Model	Chapter	Encoding	Clustering	# of classes
Passive agressive	PoC	CountVectorizer	None	2
	PoC	HashingVectorizer	None	2
	PoC	TF-IDF	None	2
	PoC	TF-IDF	None	12
Logistic Regression	PoC	CountVectorizer	None	2
	PoC	HashingVectorizer	None	2
	PoC	TF-IDF	None	2
	Main experiments	DistilBert tokenizer	None	12
	Main experiments	DistilBert tokenizer	GSSDMM	12
	Main experiments	DistilBert tokenizer	LDA	12
SVM	PoC	CountVectorizer	None	2
	PoC	HashingVectorizer	None	2
	PoC	TF-IDF	None	2
	Main experiments	DistilBert tokenizer	None	12
	Main experiments	DistilBert tokenizer	GSSDMM	12
	Main experiments	DistilBert tokenizer	LDA	12
Random Forest	PoC	CountVectorizer	None	2
	PoC	HashingVectorizer	None	2
	PoC	TF-IDF	None	2
	Main experiments	DistilBert tokenizer	None	12
	Main experiments	DistilBert tokenizer	GSSDMM	12
	Main experiments	DistilBert tokenizer	LDA	12
Naive Bayess	PoC	CountVectorizer	None	2
	PoC	HashingVectorizer	None	2
	PoC	TF-IDF	None	2
	Main experiments	DistilBert tokenizer	None	12
	Main experiments	DistilBert tokenizer	GSSDMM	12
	Main experiments	DistilBert tokenizer	LDA	12
DistilBERT	Main experiments	DistilBert tokenizer	None	12
	Main experiments	DistilBert tokenizer	GSSDMM	12
	Main experiments	DistilBert tokenizer	LDA	12

Table 10: Models trained during our research.

D Data Sheets

This section's goal is to provide a better understanding of the used dataset based on the guidelines provided in the paper Gebru et al. (2018), which advocates for creating better documentation for used data. We will comprise the proposed rules by answering the questions, divided into seven sections, provided in the aforementioned paper. To be the most precise, the answers describe the final dataset used for the modelling, available on our GitHub repository [1].

D.1 Motivation

- **For what purpose was the dataset created?** The dataset was created to test the local fake news detection methodology, which involves both topic detection and fake news detection tasks.

- **Who created the dataset and on behalf of which entity?** The dataset was created/modified by the authors of this paper, on behalf of the Warsaw University of Technology, during the Natural Language Processing course at Master's degree studies.

- **Who funded the creation of the dataset?** The creation of the dataset was not founded in any way.

D.2 Composition

- **What do the instances that comprise the dataset represent?** The instances represent a collection of online articles of moderate length (median around 400 words), which are divided into 11 fake news categories and 1 reliable.

- **How many instances are there in total?** There are 12,000 instances in total, 1000 per class.

- **Does the dataset contain all possible instances, or is it a sample of instances from a larger set?** It is a balance sample of *FakeNewsCorpus* (Pathak and Srihari, 2019), enhanced with additional statistics, and preprocessing.

- **What data does each instance consist of?** The instances are described in Table 1.

- **Is there a label or target associated with each instance?** Yes, the target is described by the column 'type', which describes which kind of text is the instance. It represents a multiclass classification problem.

- **Is any information missing from individual instances?** No.

- **Are relationships between individual instances made explicit?** No.

- **Are there recommended data splits?** No.

- **Are there any errors, sources of noise, or redundancies in the dataset?** No.

- **Does the dataset contain data that might be considered confidential?** No.

- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** Yes, the content of the articles is focused on the presidential elections in the USA, and the articles may include offensive and non-inclusive language.

D.3 Collection Process

- **How was the data associated with each instance acquired?** They were scrapped by the original authors from multiple websites.

- **What mechanisms or procedures were used to collect the data?** We do not know what the original authors did.

- **If the dataset is a sample from a larger set, what was the sampling strategy?** They were sampled at random, with the limitation that we chose only 1000 observations per class.

- **Who was involved in the data collection process?** We do not know what the original authors did, but the modifications were done by the paper authors, being students.

- **Over what time frame was the data collected?** To our best knowledge, and the timestamps provided in the dataset, all data was scrapped on 2017-11-27, however, it might not be true, as it seems incorrect.

- **Were any ethical review processes conducted?** We do not know what the original authors did, but in our case, there were none.

[1] https://github.com/adrianstando/NLP-2023W/tree/main

D.4 Preprocessing/cleaning/labelling

- **Was any preprocessing/cleaning/labelling of the data done?** Yes, we conducted the data cleaning, which is briefly described in Section 4.2.

- **Was the "raw" data saved in addition to the preprocessed/cleaned/labelled data?** No, in such cases, we advise to use the original dataset.

- **Is the software that was used to preprocess/clean/label the data available?** Yes, it is available as one of the scripts on our GitHub repository, where we used open-source Python packages.

D.5 Uses

- **Has the dataset been used for any tasks already?** No, this is the first time.

- **Is there a repository that links to any or all papers or systems that use the dataset?** There exists only the repository of this project.

- **What (other) tasks could the dataset be used for?** It could be used for regular fake news detection or topic modelling tasks.

- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labelled that might impact future uses?** We are not aware of such cases.

- **Are there tasks for which the dataset should not be used?** We are not aware of such cases.

D.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?** Yes, the dataset will be publicly available. There is a link on our GitHub repository to download the dataset from the OneDrive platform.

- **How will the dataset be distributed?** The dataset will be publicly available via the OneDrive platform, and the link is in a README file on our GitHub repository.

- **When will the dataset be distributed?** It is already available.

- **Will the dataset be distributed under a copyright or other intellectual property (IP) license and/or under applicable terms of use (ToU)?** It will be distributed under GNU GPL license.

- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.

- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.

D.7 Maintenance

- **Who will be supporting/hosting/maintaining the dataset?** We are not planning to maintain the dataset.

- **How can the owner/curator/manager of the dataset be contacted?** You can contact the owner by writing an issue on the GitHub repository.

- **Is there an erratum?** No.

- **Will the dataset be updated?** No.

- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?** It does not relate to people.

- **Will older versions of the dataset continue to be supported/hosted/maintained?** There will only be one version of the dataset.

- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** Yes, they can freely use it and publish their results.

E Model Cards

The model described in this card is used for fake news detection and labels articles into twelve classes: bias, clickbait, conspiracy, fake, hate, junk-sci, political, reliable, rumor, satire, unknown, and unreliable.

E.1 Model Details.

- Person or organization developing model: BAHM team

- Model date: 6.12.2023

- Model version: 1.0
- Model type: DistilBERT
- We used pretrained DistilBERT model implemented in PyTorch library. We fine-tuned this model for the purpose of our task for 3 epochs with batches of size 8, 500 warmup steps and decay equal to 0.01.

E.2 Intended Use

- The primary goal of the model is to detect fake news. It is designed for developers who want to build systems that flag/ filter fake news from reliable information.
- Model makes a prediction based on the first 100 tokens of the text- there is no point in including longer articles.

E.3 Factors

- Signs within the text that indicate whether the content is centered around politics, involves junk science, features clickbait, includes hate speech or includes other types of unreliable journalism practices.
- An examination of annotator bias involved assessing a sample from the training dataset during EDA to identify and address any instances of inaccurate annotations.

E.4 Metrics

- **Accuracy** - we decided to analyze accuracy as we have balanced classes and want to treat all classes equally; **Confusion Matrix** - Best way to visualize if the model learned all the classes and if it has problems with differentiation between some classes.
- We tested model performance on the validation dataset, which is a sub-sample of the entire dataset excluded from the training sample.

E.5 Evaluation Data

- To create an unbiased estimator of error, an evaluation dataset was created by taking 20% of the original sample as a validation sample.
- Data for evaluation was preprocessed exactly as data for training meaning: truncation/ padding, conversion to tensors and tokenization.

E.6 Training Data

- We chose to utilize the *FakeNewsCorpus* for our study.
- This dataset consists of articles rather than tweets, offering longer content compared to tweets, with a median length of around 400 words per text. The extended length of each text in the corpus provides the advantage of carrying ample information, contributing to a more extensive text corpus. This, in turn, facilitates easier characterization by topic or news category.
- In our study, we treat the annotation information from the *FakeNewsCorpus* as our ground truth.
- We also use the content from crawled websites to extract meaningful text features.
- Due to time constraints, we randomly sampled 12,000 articles from this dataset, distributing them evenly with 1,000 articles per class.
- For the training phase, we allocated 80% of the data, while the remaining 20% was used as a validation sample.

E.7 Quantitative Analyses

- Manual analysis of observation annotation resulted with no findings of wrong annotations.

E.8 Ethical Considerations

- Model is based on news articles and as such isn't trained based on protected attributes.

E.9 Caveats and Recommendations

- The model has no knowledge of future events and may not accurately detect fake news related to events that occurred post-training.
- Model is trained on English data sources and will not work with different languages.
- Model may not be capable of correctly detecting fake news if the news is about events not included in the training dataset.

F Resources

The full *FakeNewsCorpus* dataset can be downloaded from the following GitHub repository: https://github.com/several27/FakeNewsCorpus/releases/tag/v1.0.

The final version of the source code is available in the project's GitHub repository: https://github.com/adrianstando/NLP-2023W/tree/main/14.%20Early%20detection%20of%20fake%20news/Project1. The complete solution is included in the folder named MS3.

Our repository is divided into three directories, each representing a milestone in the project. As mentioned before, the complete solution is persisted in the MS3 folder.

Additionally, the MS3 folder contains four subdirectories - each describing a different element in the pipeline of our project. 01-EDA contains the script for Explorative Data Analysis and the dataset preprocessing. In 02-clustering, the topic detection methods were studied, whereas in 03-models, the classification models were trained. Finally, in 04-explainability, the XAI part of the project was performed.

In each part of the project, we used models and data processed earlier in the previous steps. In order to keep the repository structure clean, different symbolic links were created in the repository, which map to the used files located in different folders.

Additionally, all bigger files and datasets are available to download from the OneDrive platform. Links for the specific files are in README files in places where they should be placed.

G Reproducibility checklist

G.1 Overall results

- **MODEL DESCRIPTION – A clear description of the mathematical setting, algorithm, and/or model.** Report contains a description of all used topic detection models in the unsupervised settings and the algorithms used for the classification. For each performed experiment, there is information about the used encoding for the input data.

- **LINK TO CODE – A link to a downloadable source code with a specification of all dependencies, including external libraries.** Source Code is available at the GitHub repository for Natural Language Processing 2023 Winter course in the 14. Early detection of fake news folder: https://github.com/adrianstando/NLP-2023W/tree/main/14.%20Early%20detection%20of%20fake%20news/Project1. The repository contains a README file with all necessary information about the solution and how to navigate through the repository. Moreover, there is a requirements.txt file with all Python libraries used in the project.

- **INFRASTRUCTURE – A description of the computing infrastructure used.** All calculations were executed on single machines. For our experiments, we used different machines with respectable specifications ranging from Ryzen 5 3600 to Intel Core i7-13700k. Training of the models was not accelerated with the GPU. The calculation of the IG explanations was performed on the Google Colab platform, with the acceleration of T4 GPU (available for free).

- **RUNTIME PARAMETERS – Average runtime for each approach.** Code for the fake news detection experiments contains information about the used dataset. Training time for every model wasn't reported. Some DistilBERT models took several hours to train. The calculation of IG explanations took, with the support of GPU on the Google Colab platform, around an hour.

- **PARAMETERS – The number of parameters in each model.** Not reported.

- **VALIDATION PERFORMANCE – Corresponding validation performance for each reported test result.** Topic detection models are validated using *Calinski-Harabasz Index* and *Silhouette Coefficient*. Results of the classification task are evaluated on the validation dataset using *Accuracy* or *Weighted Accuracy*. Reports contain calculated values of those metrics for each of the models.

- **METRICS – Explanation of evaluation metrics used, with links to code** Each of the metrics used in our work has a separate section with a short description and formula on how to calculate it.

G.2 Multiple Experiments

- **NO TRAINING EVAL RUNS – The exact number of training and evaluation runs.** All performed experiments are documented in the notebooks stored in the GitHub repository of our project. Every model was trained at least 3 times with different random seeds.
- **HYPER BOUND – Bounds for each hyperparameter.** For the topic detection methods, we state the range of the examined number of clusters. Other bounds are not included in the description of our work.
- **HYPER BEST CONFIG – Hyperparameter configurations for best-performing models.** Trained classification models are serialized into the .joblib files. The serialized objects can be found on the OneDrive platform, to which a link can be found on the GitHub repository of our project.
- **HYPER SEARCH – Number of hyperparameter search trials.** The code and results for all training sessions are documented in the notebooks stored in the GitHub repository of our project
- **HYPER METHOD – The method of choosing hyperparameter values (e.g., uniform sampling, manual tuning, etc.) and the criterion used to select among them (e.g., accuracy).** In the topic detection models, the method of choosing the number of clusters was explained in the report. For the classification methods, no hyperparameter tuning was performed.
- **EXPECTED PERF – Summary statistics of the results (e.g., mean, variance, error bars, etc.).** For each classification model, we have prepared the confusion matrix between the real and predicted labels.

G.3 Datasets

- **DATA STATS – Relevant statistics, such as the number of examples.** Report contains information about the content of the original *FakeNewsCorpus* dataset and its subset, which is used for the analysis.
- **DATA SPLIT – Details of train/validation/test splits.** The code provides the random seed and the ratio of the split datasets, which can be used to replicate the training and validation split used in the experiments.
- **DATA PROCESSING – Explanation of any data that were excluded and all preprocessing steps.** Report contains extensive information about particular steps in data preprocessing. This includes text cleaning, creating new data representations and calculating statistics about data. Finally, the content of the resulting dataset is presented in the table. In the Appendix, the Data Sheets sections contain additional detailed information about the dataset.
- **DATA DOWNLOAD – A link to a downloadable version of the data.** Links to download data used in our analysis are in the Github Repository of our project: `https://github.com/adrianstando/NLP-2023W/tree/main/14.%20Early%20detection%20of%20fake%20news/Project1`.
- **NEW DATA DESCRIPTION – For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.** Not applicable in our work, as we do not collect any new data.
- **DATA LANGUAGES – For natural language data, the name of the language(s).** Used data set contains only English texts.

H Rebuttal

H.1 Rebuttal from the first part of the project

H.1.1 Team 5

We greatly appreciate the feedback provided by the linkers and are happy to see how it was mostly positive about our work. They raised some concerns connected to the methodology description of XAI techniques, as it was deemed insufficient. We agree with this assertion and plan to remedy it in future versions of our work. They also noted that we may encounter some problems with computational power as we want to base our research on a sizeable dataset. We plan to limit this dataset to 1000 observations per class, which means we will use 11,000 articles instead of 9 million.

H.1.2 Team 2

This review was submitted later on as a second thought by the authors about our work. As the major drawback of our proposal, they mention a lack of graphs or visualizations presenting the pipeline of our solutions where we directly pinpoint the function of each model/tool. We fully agree with that statement, as the general framework we will implement is a bit complex, and a visual representation of particular steps would be beneficial for the reader. We will definitely introduce such graphs to later present the final pipeline of the solution.

Another remark was made upon enabling the models to adjust to the new topics emerging in real-life datasets. The authors state that our framework should not rely solely on the static dataset and be trained only once. It is a very thoughtful idea, as we want to provide a robust solution, yet we also have to simplify the task, so we will be able to carry on the study. The proposed solution, which is a compromise for both aspects, is keeping our framework lightweight so that the clustering models will update on a regular basis, ex once a day, in the deployment environment. This way, we will be able to detect new emerging topics based on recent data, and we will not have to re-train the classifiers but only fine-tune them on recent data.

H.2 Rebuttal from the second part of the project

H.2.1 Team 7

- **In 5.2.1, it is claimed that the Passive Aggressive Classifier yielded the best results, yet only results for SVM are shown.** - We did not intend to focus on the PoC results, and that is why we decided to remove this part from the article. It must have been confusing, and that is why we have improved section 5.2 by removing numeric results, so they will be more clearly separated from one another.

- **Perhaps some columns of the final data frame could be explained in a little more detail.** With all due respect, we cannot imagine which part is not described sufficiently enough. It would be beneficial if you provided some examples of what could have been improved. However, to improve the understanding of the dataset, we included a more in-depth description in the Appendix D.

- **Maybe the final chosen dataset could be described in a little more detail at the beginning, e.g. providing some examples of misinformation classes.** - We added this information and described the dataset even more in the Appendix D.

- **I think tables 6 and 7 have swapped clusterings, because in the text it is said that LDA produced 4 clusters and GSSDMM produced 7 clusters, but in the tables the numbers of rows are swapped.** - Thank you for this comment, indeed, we made such a mistake.

- **We were under the impression that the local approach was not sufficiently justified.** - That is, unfortunately, a true statement. In the beginning, we wanted to combine the original topics of our study somehow. After we came up with the idea of `local` approach, we wanted to test it without thinking the whole concept through, which eventually resulted in inconclusive outcomes.

H.2.2 Team 5

- **It may be not really clear whether the accuracy equal to, for example, 0.65 can be considered good or bad for the task of fake news detection. Some examples of accuracies obtained by other fake news detection projects could have provided a perspective on the framework's performance.** - Thank you for this valuable comment. Initially, we intended to include a comparison of our results to some other tasks, however, due to the fact that various tasks might differ substantially, we resigned from such an approach. Some tasks are just harder than others. Additionally, as the main goal was the evaluation of the `local` approach in comparison to the `global`, we did not need external evaluation.

I 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 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.

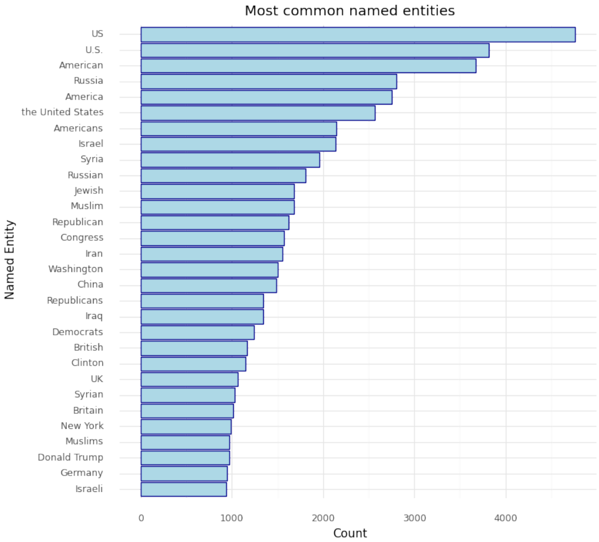

Figure 8: **Top 30 most frequently occurring named entities.**

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 voting was illegal, whereas the fake news focused mostly on Russia.

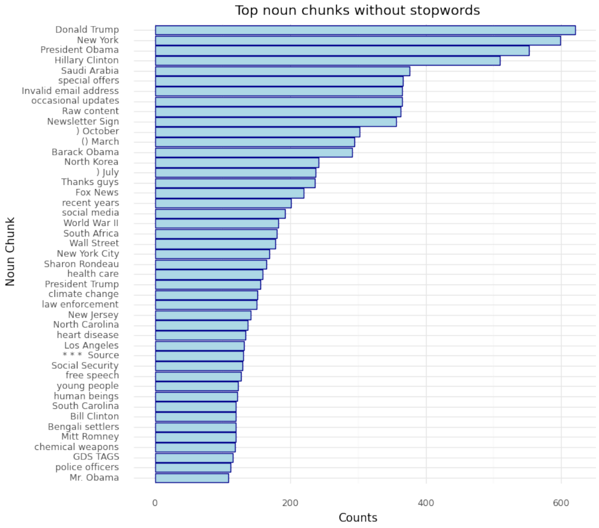

Figure 9: **Top 40 most frequently occurring noun chunks, with removed stop-words.**

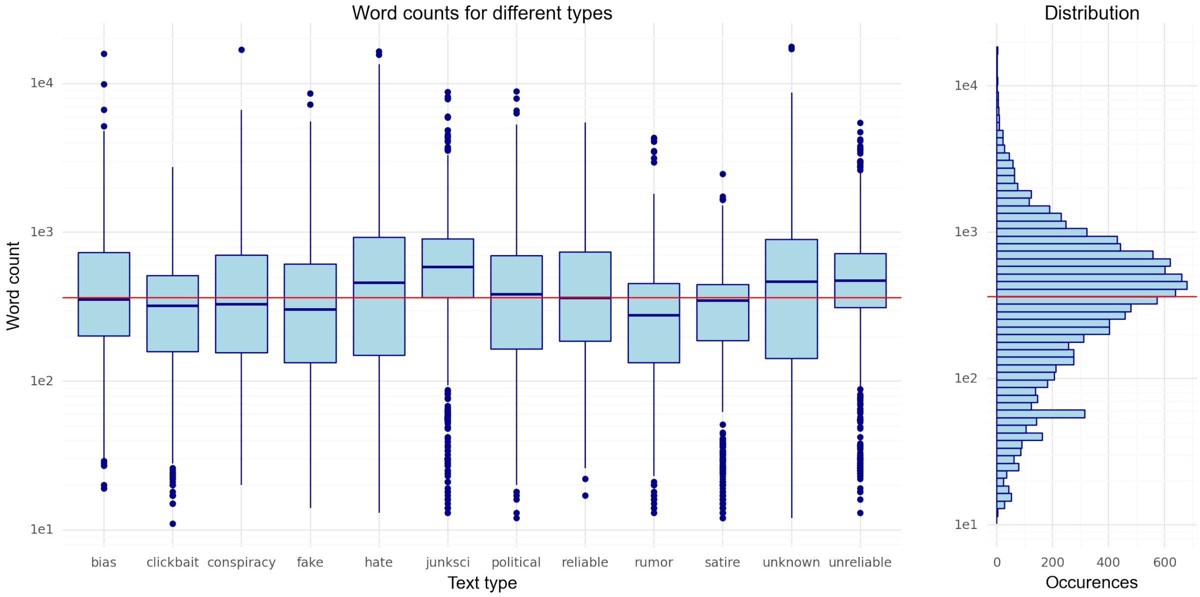

Figure 10: **Word Count analysis** The distribution of text word counts for different article types compared to the `global` distribution of all news on the right. The red line indicates the median for the reliable class.

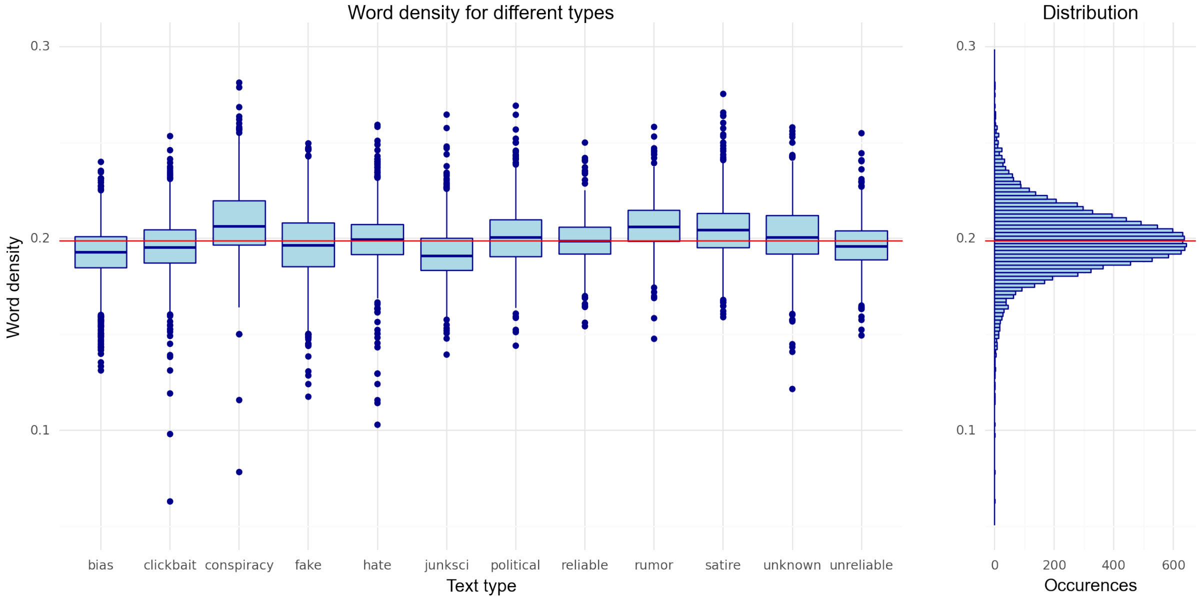

Figure 11: **Word Density analysis** The distribution of text word density for different article types compared to the `global` distribution of all news on the right. The red line indicates the median for the reliable class.

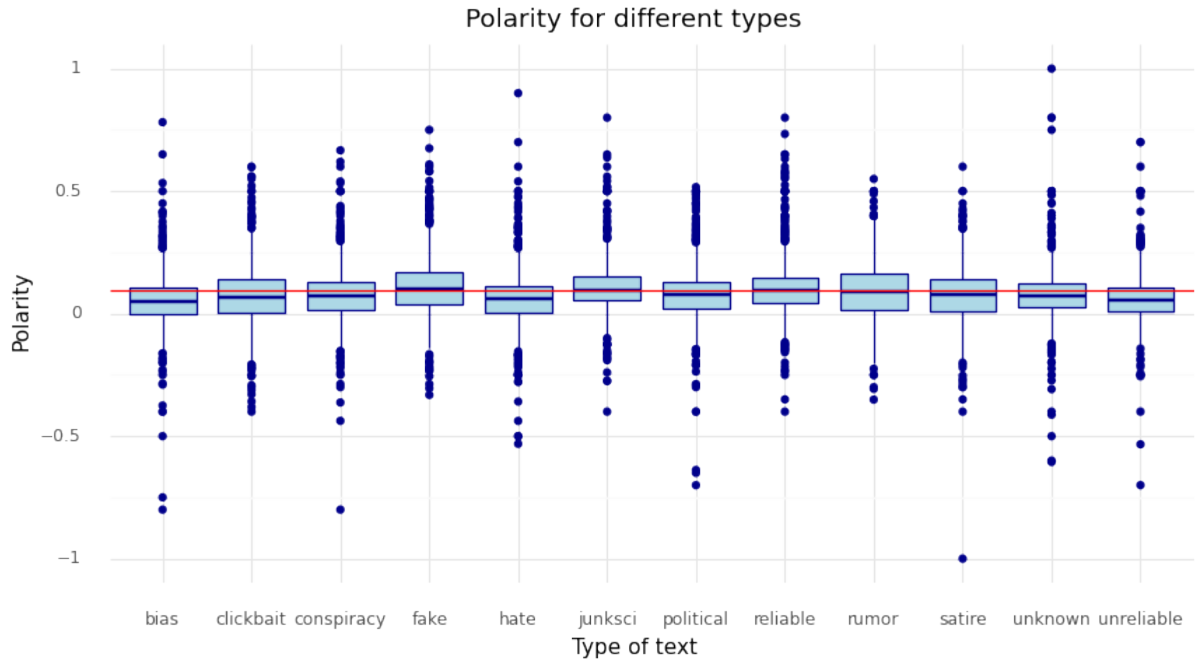

Figure 12: **Sentiment Polarity analysis** The distribution of text polarity for different article types. The red line indicates the median for the reliable class.

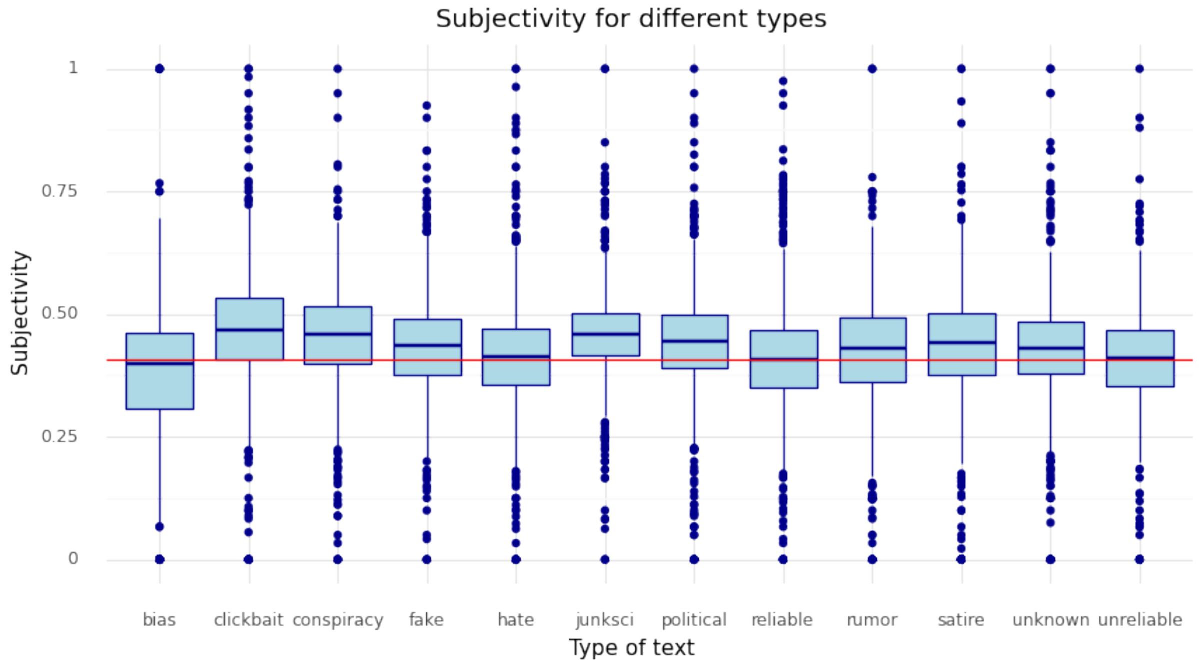

Figure 13: **Sentiment Subjectivity analysis** The distribution of text subjectivity for different article types. The red line indicates the median for the reliable class.

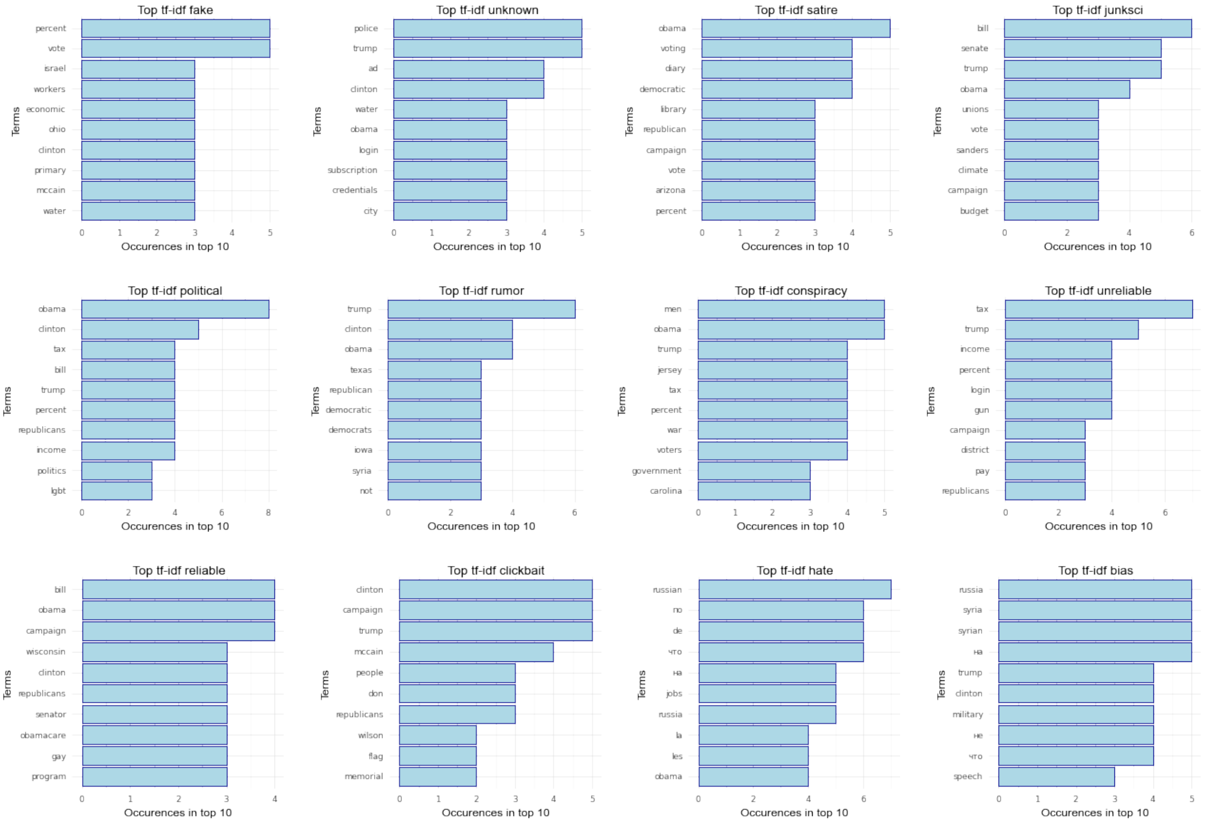

Figure 14: **Conditional TF-IDF analysis** The set of plots representing the top 10 terms according to TF-IDF for different article types.