information

Article

The PolitiFact-Oslo Corpus: A New Dataset for Fake News Analysis and Detection

Nele Põldvere [1], Zia Uddin [2,*] and Aleena Thomas [2]

[1] Department of Literature, Area Studies and European Languages, University of Oslo, 0315 Oslo, Norway; nele.poldvere@ilos.uio.no
[2] Sintef Digital, 0373 Oslo, Norway; aleena.thomas@sintef.no
* Correspondence: zia.uddin@sintef.no

Abstract: This study presents a new dataset for fake news analysis and detection, namely, the PolitiFact-Oslo Corpus. The corpus contains samples of both fake and real news in English, collected from the fact-checking website PolitiFact.com. It grew out of a need for a more controlled and effective dataset for fake news analysis and detection model development based on recent events. Three features make it uniquely placed for this: (i) the texts have been individually labelled for veracity by experts, (ii) they are complete texts that strictly correspond to the claims in question, and (iii) they are accompanied by important metadata such as text type (e.g., social media, news and blog). In relation to this, we present a pipeline for collecting quality data from major fact-checking websites, a procedure which can be replicated in future corpus building efforts. An exploratory analysis based on sentiment and part-of-speech information reveals interesting differences between fake and real news as well as between text types, thus highlighting the importance of adding contextual information to fake news corpora. Since the main application of the PolitiFact-Oslo Corpus is in automatic fake news detection, we critically examine the applicability of the corpus and another PolitiFact dataset built based on less strict criteria for various deep learning-based efficient approaches, such as Bidirectional Long Short-Term Memory (Bi-LSTM), LSTM fine-tuned transformers such as Bidirectional Encoder Representations from Transformers (BERT) and RoBERTa, and XLNet.

Keywords: corpus development; text type; sentiment; part-of-speech; Bi-LSTM; transformers

Citation: Põldvere, N.; Uddin, Z.; Thomas, A. The PolitiFact-Oslo Corpus: A New Dataset for Fake News Analysis and Detection. *Information* **2023**, *14*, 627. https://doi.org/10.3390/info14120627

Academic Editors: Vasco N. G. J. Soares, João M. L. P. Caldeira, Bruno Bogaz Zarpelão and Jaime Galán-Jiménez

Received: 14 September 2023
Revised: 16 November 2023
Accepted: 16 November 2023
Published: 23 November 2023

1. Introduction

Fake news is a societal issue that does not seem to be slowing down. If anything, the growing levels of distrust in mainstream media and government agencies [1] have added fuel to the fire of fake news and allowed for false and misleading information to spread through a variety of digital channels. From COVID-19 conspiracy theories to the bogeyman of "Nazism" in Ukraine, fact-checking websites such as PolitiFact in the US and others around the world have their work cut out for them. Automatic fake news detection can support these efforts, as shown by the large amount of research on this topic in natural language processing and machine learning, e.g., [2–9]. However, the results obtained from these studies are only as good as the data on which they are trained. Who determines what is fake and what is real? How much data are enough? How can we control for confounding variables? These are only some of the questions that developers of fake news datasets need to consider for valid and reliable outcomes.

In this study, we present a new dataset of fake and real news in English with a focus on the US, namely, the PolitiFact-Oslo Corpus. The corpus is built based on a pipeline for collecting quality data from major fact-checking websites, which we describe in detail to facilitate future corpus building efforts in this area. We also examine the linguistic characteristics of the corpus and its suitability for automatic fake news detection using a series of case studies in natural language processing and machine learning. The latter are

applied both to the PolitiFact-Oslo Corpus and the DeClarE dataset, based on data from the same fact checker (see below), in order to compare their characteristics. Moreover, we run separate analyses on the full datasets and the main text types of the PolitiFact-Oslo Corpus (social media, news and blog) to demonstrate the importance of contextual information in fake news corpora.

The PolitiFact-Oslo Corpus grew out of a need for a more controlled and effective dataset for fake news analysis and detection model development based on recent events. So far, fake news researchers have relied on three main types of data: (i) simulated data, (ii) natural data that have been classified for veracity based on source reputation, and (iii) natural data that have been individually labelled for veracity by experts. The first data source has a clear disadvantage: the training data have been artificially generated and therefore have little external validity. Natural data based on source reputation allow for large-scale data collection procedures, but since there is no one-to-one correspondence between truth content and source reputation, then such data are not very reliable. Therefore, the best approach to collecting data for automatic fake news detection is by using natural data that have been individually labelled for veracity by experts, such as journalists working for reputable fact-checking organizations.

Fact-checking websites focusing on news in the US have served as a basis for several existing fake news datasets. Early examples are PolitiFact and Channel 4 in [10] and Emergent in [11], with 221 statements and 300 rumors, respectively. In fact, PolitiFact.com has been a favorite among fake news researchers, possibly due to its good reputation and a fine-grained system of ratings: False, Mostly False and Pants on Fire to True, Mostly True and Half True. The LIAR dataset, for example, includes as many as 12,800 statements from PolitiFact [12], while MisInfoText [13] provides access to a large, but unverified, dataset based on the fact checker. PolitiFact was also used in [14]'s study of fake news around the 2016 US presidential election, which, according to the authors, is when fake news went mainstream. The dataset that we use to compare with the PolitiFact-Oslo Corpus in this study, DeClarE [15], contains a subset from PolitiFact collected before 2017.

The ready availability of fake news datasets based on PolitiFact thus begs the question: why the PolitiFact-Oslo Corpus? We argue that our corpus, a structured collection of natural texts in machine-readable form, addresses several limitations of the previous datasets. This has important implications, not only for our understanding of the best practices of language data collection and management, but also for being able to identify *real* differences between fake and real news. Automatic extraction methods have meant that many of the datasets contain news items that do not in fact correspond to the claim fact-checked by PolitiFact. Often, these are multimodal resources such as photos and videos (e.g., "An image shows gas prices on 6 January 2021"). In these cases, what is extracted is not the content of the image but the text that accompanies it, which may be irrelevant or incomplete. At other times, what is extracted is not the news item corresponding to the claim but the claim itself, which is very short and may not be enough for robust text classification. Finally, there is a problem with the lack of access to metadata about the news items. An important piece of information that is often missing and that has negative consequences for language models is text type. Since most fake news in PolitiFact seems to be from social media, and most real news seems to be from news websites, then a model trained on them is more likely to identify differences between the text types of social media and news websites than fake and real news (cf. [16]). This limitation of the previous studies calls into question the validity and effectiveness of the fake news detection models developed so far. The DeClarE dataset, for instance, provides access to the news items rather than the claims, but as far as we know, no steps were taken to ensure a direct and valid relationship between the news items and the claims or to determine their text type. As will be shown, the PolitiFact-Oslo Corpus meets all the requirements above, making it highly suited for fake news detection.

The case studies are a mix of approaches from natural language processing and machine learning, drawing on insights from, e.g., [17–23]. In the former, we carry out exploratory analyses of the sentiment and part-of-speech features of the corpus and, in

the latter, we examine the applicability of the corpus for a deep learning-based efficient approach for fake news detection by means of various state-of-the-art approaches. These are: Deep Belief Networks (DBN), Long Short-Term Memory (LSTM), Bidirectional LSTM, Bidirectional Encoder Representations from Transformers (BERT), RoBERTa, DistilBERT, and XLNet. The approaches in this study are applied both separately and in combination to achieve better machine learning performance.

2. The PolitiFact-Oslo Corpus

As the name implies, the PolitiFact-Oslo Corpus relies on the fact-checking website PolitiFact.com for its data. The data collection procedure was a collaborative effort between the University of Oslo and Sintef Digital in Oslo, Norway. In what follows, we describe the procedure in detail and then present an overview of how the key features of the corpus make it uniquely placed for practical applications in fake news analysis and detection model development.

2.1. Data Collection, Design and Metadata

The procedure for collecting data for the PolitiFact-Oslo Corpus was a combination of automatic and manual techniques to have greater control over what is included. The automatic techniques involved extracting the fact-checking articles and their metadata from PolitiFact, including all the external links. To reduce later manual work, at this stage, we also filtered out those fact-checking articles that corresponded to spoken text types such as 'ad', 'press conference', 'radio', 'TV', 'debate', etc. This is because transcripts created for general popular use come with no assurances that they are detailed enough for robust text analysis.

Then, we manually identified the one link that corresponded to the PolitiFact's claim to extract its content. We extracted the complete text, its title, the author/poster, and the date. Several important decisions were made at this stage. For example, we decided not to include content that was clearly satirical (e.g., *The Onion*), because satire is disseminated for very different reasons than fake news in its strict sense. Thus, adding satire would have negatively affected our results, in line with the functionalist idea that differences in communicative intent are reflected in linguistic structure [24]. In the case of images, the text was extracted only if it sufficiently described the content of the image. One example of an image that was excluded was of Dwayne "The Rock" Johnson, whose photo of him sitting on a beach was entitled "Epstein Island" by a Facebook user. The claim that PolitiFact then sought to verify was: "A photo shows Dwayne "The Rock" Johnson on Epstein Island" (of course, it was rated Pants on Fire). However, since the text itself did not contain any critical information about the content of the photo (not even who was in it), then it would have added little value to text classification models trained on the data. Instead, we decided to go with content that was substantial enough for robust text analysis. Finally, we came across several news items that had been corrected after being fact-checked by PolitiFact. In these cases, we accessed the original version of the news item using Wayback Machine, a digital archive, to make sure that the item indeed corresponded to the PolitiFact's claim. Thus, the manual approach used to build the PolitiFact-Oslo Corpus was crucial for ensuring an effective dataset.

To ensure a controlled dataset, we added important metadata information to the corpus. In addition to the above metadata (author/poster, date), the PolitiFact-Oslo Corpus contains information about the text type (e.g., social media, news and blog) and source (e.g., Facebook, *The Gateway Pundit*) of the news items. This information is more consistent than that provided by PolitiFact, where, for instance, an image may be tagged either as a 'viral image' or 'Instagram post' (depending on the source). Within text type, we conflated news articles and blog posts because of the fuzzy boundary between the two. This is because most of the news websites in our corpus are run by highly specialized news organizations with limited budget and resources, and therefore their function and design are very similar to blogs run by individuals. As will be shown, taking into consideration where the news

item is published has important implications for fake news analysis and detection model development, due to contextual variation between different types of texts.

Figure 1 summarizes the data collection procedure implemented in this study. The pipeline in the figure is meant to be general enough to facilitate future corpus building efforts with data from any of the major fact-checking websites (e.g., PolitiFact, Snopes, Emergent in English). It consists of four main steps and two optional steps, depending on the structure of the fact-checking website and the researchers' goals. While the pipeline itself is agnostic about the nature of the data collection procedure (automatic or manual), we strongly believe that manual techniques are key to ensuring a more controlled and effective dataset.

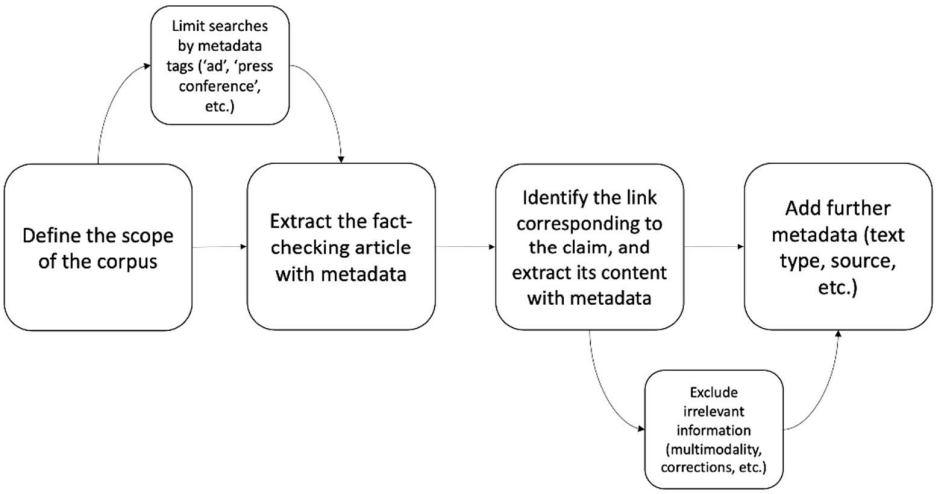


Figure 1. A pipeline for collecting quality data from PolitiFact (or any other fact-checking website).

The outcome of the data collection procedure for the PolitiFact-Oslo Corpus is given in Table 1. It should be noted that the development of the corpus is a work-in-progress and therefore the figures in the table are preliminary. However, they provide a large enough sample for the natural language processing and machine learning models developed in this study. At present, the corpus contains 428,917 words stored in 2745 texts, covering a time period between January 2019 and December 2022. The average number of words in the texts is 158, which means that the texts are relatively short. Table 1 presents the distribution of the text types in terms of the number of texts, words and average text length across the broad labels of False and True, or Fake and Real, where the former subsumes the PolitiFact ratings of False, Mostly False and Pants on Fire, and the latter subsumes True, Mostly True and Half True. We have excluded the ratings of Full Flop and No Flip due to their low occurrences. As can be seen in the table, over 83% of the texts are from social media, and 14% are from news and blog, making these two the largest text types in the corpus. In addition, the corpus contains 249 different sources, with Facebook being the most common one. The labels are also not uniformly distributed, with over 80% of them being False. This is a limitation of the PolitiFact-Oslo Corpus to which we return in Section 5.

2.2. Key Features and Applications

As previously mentioned, the PolitiFact-Oslo Corpus is characterized by three key features that make it uniquely placed for practical applications in fake news analysis and detection model development.

(i) The texts have been individually labelled for veracity by expert journalists working for a reputable fact-checking organization.
(ii) They are complete texts, rather than short claims or statements, which strictly correspond to the claims in question.
(iii) They are accompanied by important metadata about the news items.

Table 1. Distribution of text types in the PolitiFact-Oslo Corpus (as of November 2023).

Text Type	Texts		Words		Average Text Length *	
	False	True	False	True	False	True
Social media	1878	403	119,002	20,498	63	50
News and blog	351	42	215,563	33,156	614	789
Press release	27	25	16,105	14,699	778	588
Personal	8	2	951	1348	119	674
Academic	5	0	4352	0	869	0
Political	3	1	2543	700	848	700
Sub-total	2272	473	358,516	70,401	160	148
Corpus	2745		428,917		158	

* Average text length is given in words; all the decimal values have been rounded up to the nearest integer.

The first feature ensures that the news items, based on real-world data, are reliably classified for automatic detection using a fine-grained system. The second feature of having complete texts is important for building robust models, since "text classification relies mainly on the linguistic characteristics of longer text" [13]. Moreover, the texts of the news items correspond strictly to the claims in question to make sure that there is a direct and valid relationship between the two. The third feature allows us to control for well-known confounding variables in fake news research, such as text type, authorship, time, etc. Text type, in particular, is important for teasing apart the differences in variation in how news is disseminated across a variety of digital channels. In addition, the PolitiFact-Oslo Corpus gives access to new data about recent events such as COVID-19, the 2020 US presidential election and the Russian invasion of Ukraine, when fake news is believed to have reached new dimensions [25].

The corpus, therefore, has clear applications in fake news research and detection model development. Research in style-based fake news classification, for instance, has generated plenty of evidence that there are systematic differences in the language and style of fake and real news; the former is often characterized by more negative and intense emotions, exemplified by intensifiers and negative sentiment words as well as the more frequent use of verbs, adjectives and adverbs [7,8,16,26]. This information could then be used by the very same fact-checking organizations that provided the data for an initial screening of potentially false and misleading information. Due to its focus on text and the language of fake news in the strict sense, the PolitiFact-Oslo Corpus is less suited for the identification of multimodal features such as those found in deepfakes (or even emojis). Although multimodal data may have played a role in the fact-checkers' decisions, only the text corresponding to the data is represented in a reliable way in the corpus (but see the Fakeddit dataset for an alternative [27]).

3. Fake News Analysis Using Natural Language Processing

In this section, we present a couple of case studies in natural language processing, sentiment and part-of-speech (POS), to explore the specific linguistic characteristics of the PolitiFact-Oslo Corpus across the broad labels of True and False. For this, we used the two largest text types which together constitute over 97% of the dataset, namely, social media, and news and blog (a total of 2674 news items). The idea is to reveal potential similarities and differences between the text types.

3.1. Sentiment

Firstly, we conducted sentiment analysis of the news items using the VADER sentiment analysis tool [28] from the Stanford CoreNLP natural language processing toolkit [29]. We compared the number of news items with positive and negative sentiment. We considered

a news item to have positive sentiment if the VADER 'compound' score is greater than or equal to 0. A news item was considered to have negative sentiment if the 'compound' score is less than 0. We plot the differences in sentiment for the classes of True and False in Figure 2.

Figure 2. Distribution of news items with positive and negative sentiment in True and False classes across the text types of social media (**upper** plot), and news and blog (**lower** plot).

The first thing that one notices in Figure 2 is the difference between social media, on the one hand, and news and blog, on the other. Specifically, there is a higher proportion of items with negative sentiment in the news articles and blog posts than in the social media posts, regardless of whether the items are True or False. This is perhaps surprising, considering that news articles and blog posts tend to be distributed to wider audiences to shape public opinion. An important difference between fake and real news, however, is that, in both text types, the percentage of items with positive sentiment is higher in the True class than it is in the False class. The higher percentage of positive sentiment in the former could be because of factual reporting and a generally more positive and favorable depiction of the news events. By contrast, False items often contain deliberately deceptive content, i.e., disinformation, which may lead to "non-conscious leakage of anxiety" [30] about lying and, consequently, to a higher negative sentiment score. Examples (1) and (2), snippets of news articles and blog posts from the PolitiFact-Oslo Corpus, illustrate this difference. While both texts are concerned with COVID-19, they view the topic through different evaluative lenses. The news article from which (1) is extracted was labelled Mostly True by PolitiFact and received a highly positive sentiment score (0.9148).

(1) It sounds like science fiction. A fabric that smothers coronaviruses in less than a minute, its electrokinetic superpowers choking the life out of COVID-19 like Thanos squeezing the life out of Loki? It's real, says Chandan Sen, the director of the Indiana Center for Regenerative Medicine and Engineering at the Indiana University School of Medicine in Indianapolis. And, with the number of novel coronavirus infections sitting at 4 million as of 10 May, it holds tantalizing potential for the future of personal protective equipment.

A highly negative sentiment score (−0.9993) was obtained for the blog post in (2), labelled False by PolitiFact.

(2) The "novel Coronavirus" outbreak affecting China and many other countries right now, has been determined to be a military BIO-WEAPON, which was being worked on at the Wuhan Virology Laboratory by China's People's Liberation Army, Nanjiang Command. Somehow, it got out. The world is now facing a massive wipe-out of humanity as a result.

We also investigated the average of the sentiment scores in all the False and True items in the two text types. These values are reported in Table 2. In the table, we can make a similar observation as above: the average sentiment scores of True items are higher than the average sentiment scores of False items. The higher average sentiment scores could be another indication that accurate and credible news content tends to evoke more positive emotions than false and deceptive content.

Table 2. Average sentiment scores of False and True items across the text types of social media, and news and blog.

Text Type	Average Sentiment Score		News Items	
	False	**True**	**False**	**True**
Social media	−0.102	−0.03	1878	403
News and blog	−0.228	−0.115	351	42
Sub-total	−0.122	−0.038	2229	445
Total	−0.16		2674	

3.2. Part-of-Speech

Next, we looked at the distribution of POS tags in the corpus. Focusing again on the two largest text types, Figure 3 shows the top five POS tags across the False and True classes. By far the most frequent POS tag in all cases is nouns, followed by verbs. There is some variation in the rest of the tags. Pronouns, for example, only feature frequently in the False class of social media posts. This is in line with the more involved and informal style of social media posts compared to news articles and blog posts, as well as compared to the more informational and formal style of factual reporting as identified in previous research [16]. An example of a False social media post from Facebook is given in (3). Note the frequent use of pronouns, particularly personal pronouns such as *she*, *her*, *you*, *me*, *us*, *they* and *I*, which together make up a total of 14% of the complete social media post (not shown here due to copyright and privacy reasons).

(3) Queen Pelosi wasn't happy with the small USAF C-20B jet, Gulfstream III, that comes with the Speaker's job . . . OH NO! Queen Pelosi was aggravated that this little jet had to stop to refuel, so she ordered a Big Fat, 200-seat, USAF C-32, Boeing 757 jet that could get her back to California without stopping [. . .] Queen Pelosi wants you and me to conserve our carbon footprint. She wants us to buy smaller cars and Obama wants us to get a bicycle pump and air up our tires. Who do these people think they are??? Their motto is . . . Don't do as I do . . . JUST DO AS I SAY!

The higher frequency of determiners in the True class is an indication of the presence of complex noun phrases (e.g., *the new bill*, *the state's stay-at-home order*), which contribute further to the more formal style of truthful news items.

Within news and blog, there are more proper nouns in the False class than in the True class. This could be an indication of an attempt to make the false content more credible by adding recognizable names and entities (e.g., a long list of schools that are to be affected by an alleged ban by Joe Biden on school choice vouchers in Wisconsin). By contrast, the True class stands out with regard to its use of adjectives to add detail and richness to the description of the news events. Considering that previously adjectives have been found to characterize fake news rather than real news (see Section 2.2), we need to look closer into their function in the PolitiFact-Oslo Corpus in the future.

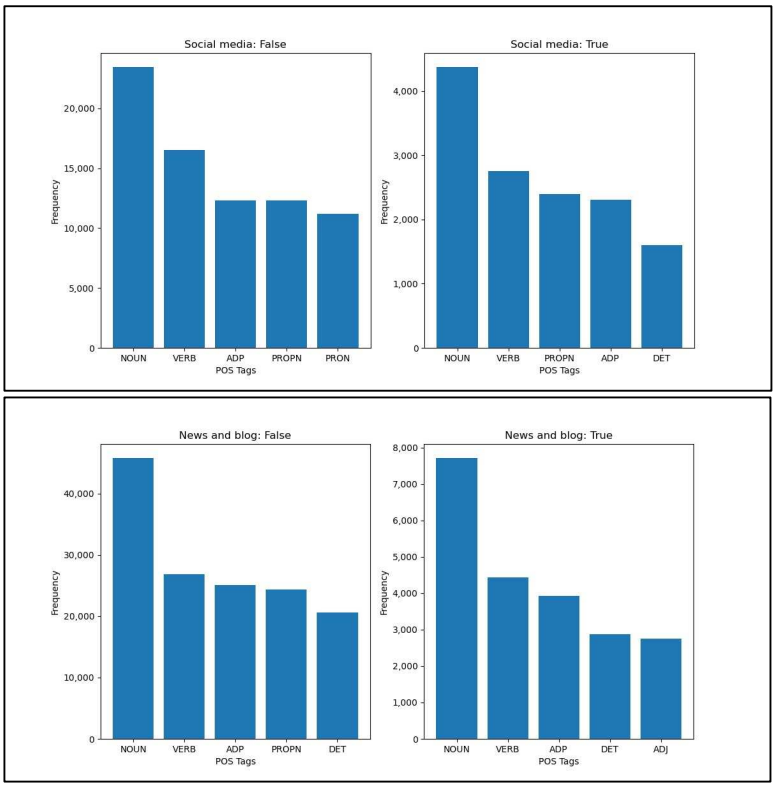


Figure 3. Distribution of POS tags in False and True classes across the text types of social media (**upper** plots), and news and blog (**lower** plots).

4. Machine Learning for Fake News Detection

4.1. Overview

As previously mentioned, the main application of the PolitiFact-Oslo Corpus is in automatic fake news detection. Therefore, the rest of the case studies are concerned with machine learning and a deep learning-based efficient approach for fake news detection by means of various state-of-the-art approaches. Deep learning has contributed a lot recently in many fields such as pattern analysis and artificial intelligence [31–33], with important applications in fake news detection model development [34–38]. However, deep learning has two major disadvantages: the first disadvantage is the overfitting problem most of the time and the second one is that it takes a lot of time to model the underlying data.

The pioneer deep learning algorithm was Deep Belief Networks (DBN). It consisted of Restricted Boltzmann Machines (RBMs) that made the training faster than previous learning methods. Recurrent Neural Networks (RNNs) are, however, a better choice than DBN, although they have gradient vanishing point problems [32]. To avoid that, the Long Short-Term Memory (LSTM) deep learning model was introduced. Basically, it consists of some gates over input sequences to remember the history inside the data. To be able to follow both directions to model sequential information via LSTM, Bidirectional LSTM may be used. In addition, some studies have applied pre-trained models for sequence classification such as Bidirectional Encoder Representations from Transformers (BERT), RoBERTa, DistilBERT, and XLNet [33]. Transformer models are cutting-edge natural language processing (NLP) models that have played a pivotal role in advancing the capabilities of language understanding and generation in the field of artificial intelligence. BERT, developed by Google AI, was a groundbreaking model that marked a significant shift in NLP. It introduced the concept of bidirectional context, allowing the model to understand the meaning of a word in a sentence by considering the surrounding words. BERT's pretraining on vast text corpora, followed by fine-tuning on specific tasks, has set the standard for a wide range of NLP applications. A Robustly Optimized BERT Pretraining Approach (RoBERTa), an improvement upon BERT, was developed by Facebook AI. It refines BERT's

training methodology by employing larger batch sizes and more extensive data, resulting in remarkable performance improvements on various NLP tasks. RoBERTa is celebrated for its robustness and efficiency. DistilBERT, created by Hugging Face, is a distilled version of BERT designed for efficiency. It retains BERT's performance while being smaller and faster to train and deploy. DistilBERT is particularly suitable for resource-constrained applications, making it a popular choice for real-world usage. Transformer-XL Network (XLNet), developed by Google AI and Carnegie Mellon University, extends the transformer architecture and introduces a novel permutation-based training approach. This method allows XLNet to capture complex dependencies in text by considering all possible word permutations, surpassing the performance of previous models on various NLP benchmarks. In short, BERT, RoBERTa, DistilBERT, and XLNet have significantly advanced the field of NLP, each bringing unique contributions in terms of bidirectional context understanding, robustness, efficiency, and complex dependency modeling, respectively. These models have set the foundation for a new era of language understanding in artificial intelligence.

In this study, we focus on processing text data, features, and fake news prediction as a smart application. Figure 4 shows a schematic setup of a text-based fake news prediction system in a smart application where a user provides a query text, and a server processes the text to apply feature extraction and deep learning for predicting whether the text is fake or not. Figure 5 shows the basic architecture of the proposed fake news prediction system consisting of training and testing procedures. In the training part, text data from all the users are obtained and the features are trained using machine learning models such as Bidirectional LSTM and BERT-based fine-tuned models. In the testing part, features from a sample test are applied to the trained models to make the decision whether the text contains fake news or not.

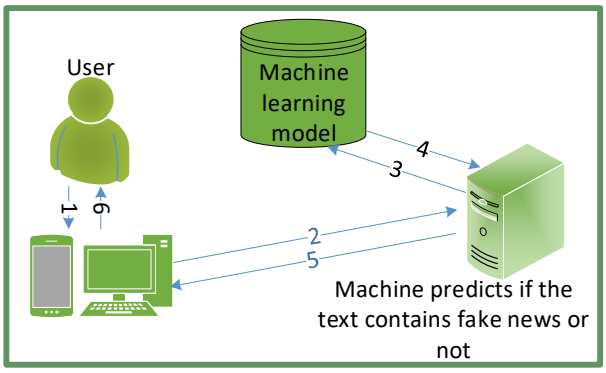


Figure 4. A schematic setup of a text-based fake news prediction system.

As previously mentioned, machine learning models capable of handling text data sequences are applied in this study. To achieve high accuracy, we propose to combine Bidirectional LSTM and BERT, because of their capabilities to model underlying events in text information. LSTM has recurrent connections between its hidden units, which connect history to the present state. LSTM was derived from typical RNNs which generally cause vanishing gradient problems during the processing of long-term information, which LSTM can overcome. Bidirectional LSTM should perform even better than typical unidirectional LSTM to model time-series information from both directions. Figure 6 shows a sample Bidirectional LSTM model.

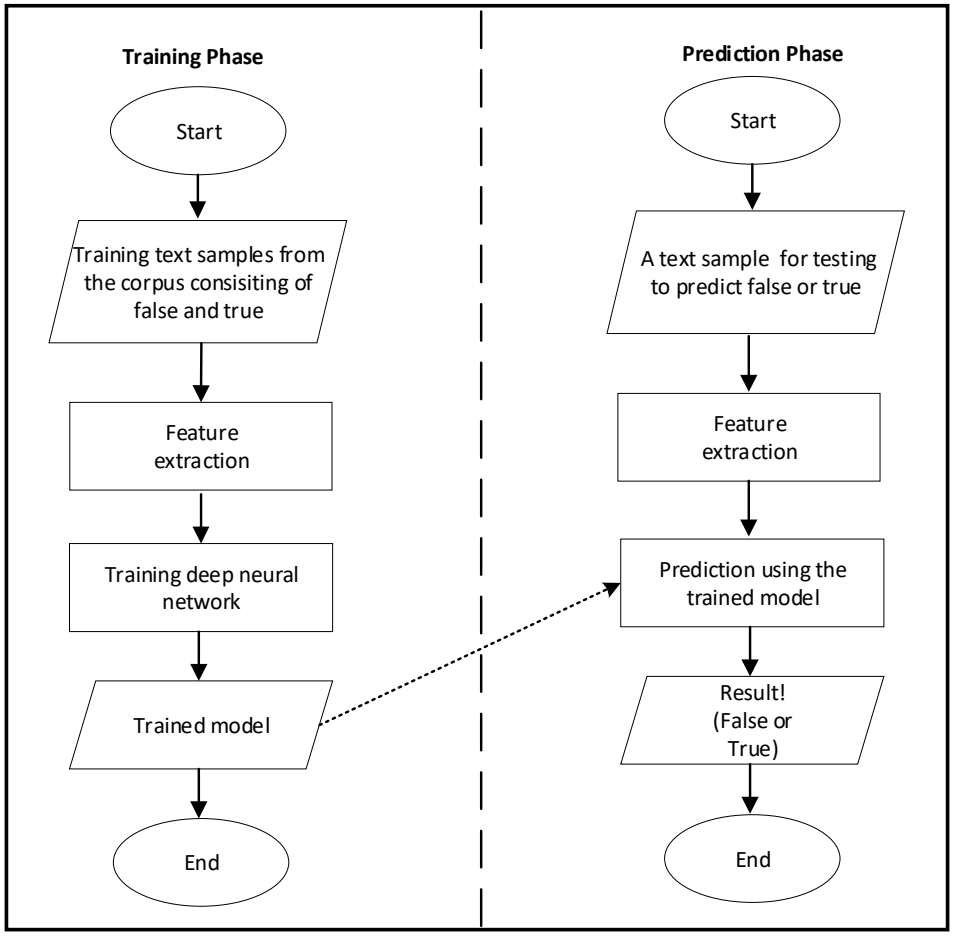


Figure 5. Flow chart of a text-based fake news prediction system.

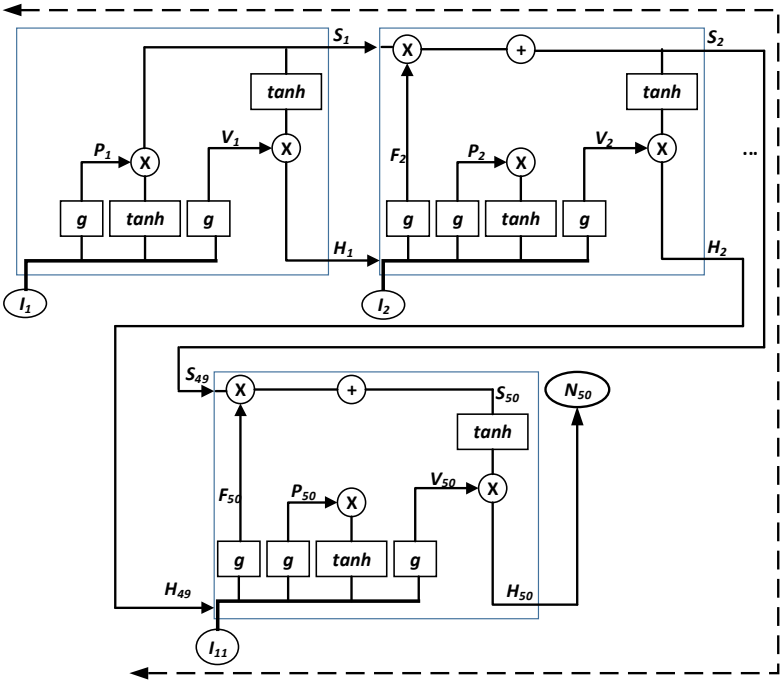


Figure 6. Bidirectional LSTM model.

Each LSTM block contains a cell state and three gates, namely, input, forget and the output gate. The input gate I_t is determined as

$$I_t = \beta(W_{PI}P_t + W_{HI}H_{t-1} + b_I) \quad (1)$$

where W is the weight matrix, b represents the bias vectors, and β is the logistic sigmoid function. The forget gate P can be expressed as

$$P_t = \beta(W_{PF}I_t + W_{HF}H_{t-1} + b_F). \quad (2)$$

The long-term memory is stored in a cell state vector S expressed as

$$S_t = P_tS_{t-1} + I_t\tanh(W_{PS}V_t + W_{HS}H_{t-1} + b_S). \quad (3)$$

The output gate N determines what is going to be an output expressed as

$$N_t = \beta(W_{PO}P_t + W_{HO}H_{t-1} + b_O). \quad (4)$$

The hidden state H is expressed as

$$H_t = O_t\tanh(S_t). \quad (5)$$

Finally, the output O can be determined as

$$O = softmax(W_UH_l + b_U) \quad (6)$$

where l indicates the last LSTM number. Figure 7 shows the fine-tuned transformer model using LSTM and dense layers applied in this study, where the *softmax* function is applied to the final layer weights to make the decision. Figure 7 shows the transfer learning using the LSTM-based fine-tuned transformer-based (BERT, RoBERTa, DistlilBERT, XLNet) basic architecture model applied in his study. We have compared the fine-tuned transformer-based model with three other typical algorithms: DBN, LSTM, and Bidirectional LSTM.

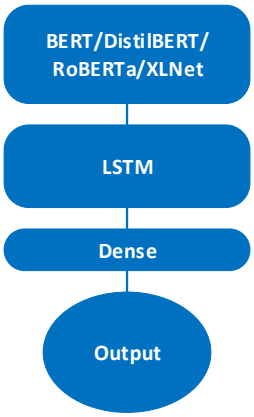


Figure 7. Transfer learning model fine-tuned LSTM and dense layers.

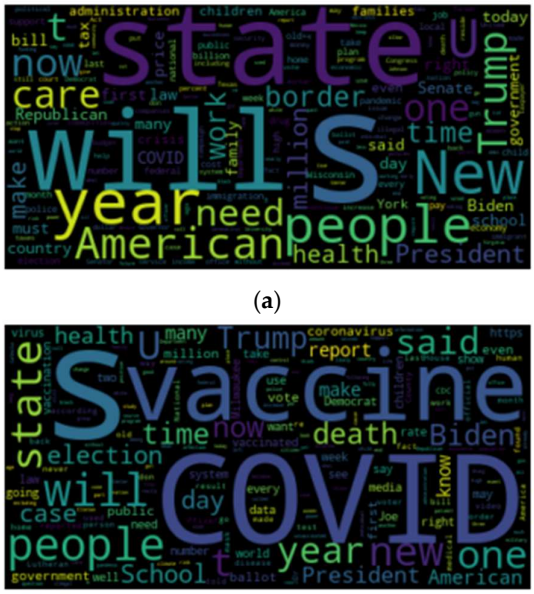
4.2. Experimental Results and Discussion

The experiments were applied both to the PolitiFact-Oslo Corpus and the DeClarE dataset in order to compare their characteristics. The classes were both fine-grained (i.e., True, Half True, Mostly True, False, Mostly False, Pants on Fire) and broad (i.e., Real or True combining True, Half True and Mostly True as well as Fake or False combining False, Mostly False and Pants on Fire); see Table 1 for the distribution of the classes. Moreover, in the PolitiFact-Oslo Corpus, we carried out separate experiments on the full dataset as well as the two largest text types: news and blog, and social media. Five-fold cross-validation was applied to evaluate the machine learning models. Twenty percent of the data were

used for testing each fold and ten percent of the data extracted from the training data were used for validation.

4.2.1. Full Dataset

We started out with the full dataset of the PolitiFact-Oslo Corpus. Figure 8 shows the word clouds of the True and False classes based on the most common words in the corpus. As can be seen in the figure, the word clouds are different in terms of lexical and topical features. False news, in particular, is clearly focused on topics that, over the last few years, have been highly vulnerable to manipulation and misinformation (e.g., COVID-19, vaccines).



(**a**)

(**b**)

Figure 8. Word clouds of (**a**) True and (**b**) False classes in the full dataset of the PolitiFact-Oslo Corpus.

Tables 3 and 4 show the results (in average recall rates (%) and their means) of traditional embedding features and various machine learning approaches to the corpus based on two and six classes, respectively. As expected, the fined-tuned transformer models (e.g., XLNet) show overall better performance. Moreover, two classes provide higher accuracy rates than six classes since the number of samples for training decreases when the number of classes increases. Figure 9 shows the loss vs. epoch plots of the Bidirectional LSTM and BERT models, which show quick convergence to lower loss when the number of epochs increases. Although transformer models are more time-consuming than Bidirectional LSTM, the former also shows convergence considering the number of epochs. It is clear, however, that in the case of both two and six classes, the models have been biased towards the False class. This is due to the considerably lower number of texts in the True class compared to the False class (see Table 1).

Table 3. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the full dataset of the PolitiFact-Oslo Corpus based on two classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	19.21	26.33	33.15	33.24	39.23	28.14	28.61
False	70.32	90.52	92.69	95.14	92.70	92.63	98.59
Mean	**44.76**	**58.42**	**62.45**	**61.88**	**62.64**	**60.38**	**63.60**

Table 4. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the full dataset of the PolitiFact-Oslo Corpus based on six classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	2.78	3.21	3.28	4.58	4.47	15.13	3.12
Half True	10.54	12.31	13.72	15.23	25.72	16.59	10.14
Mostly True	3.21	4.73	6.24	5.77	8.33	5.02	12.37
False	33.47	47.93	50.78	65.29	65.15	56.19	72.66
Mostly False	19.48	22.34	24.35	18.71	11.09	15.07	15.13
Pants on Fire	27.34	28.57	27.53	28.76	27.11	23.15	22.57
Mean	**16.14**	**19.85**	**20.98**	**23.05**	**23.65**	**21.86**	**22.66**

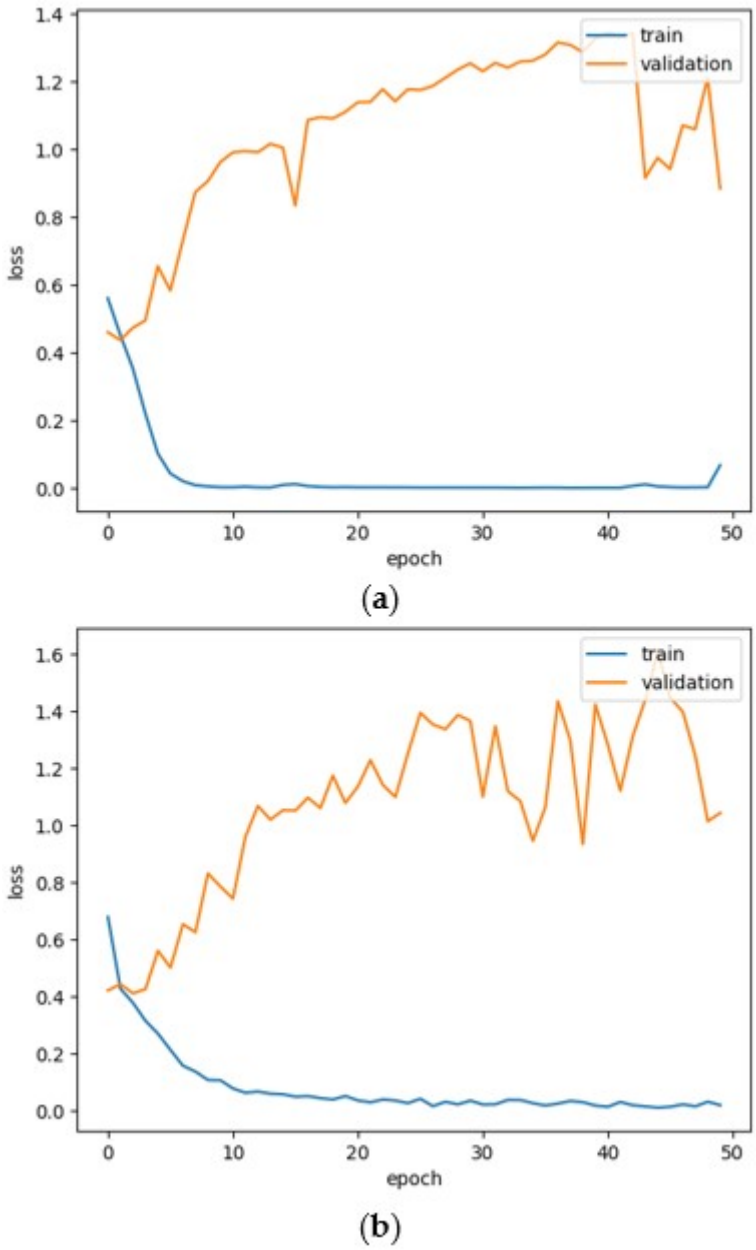


Figure 9. (**a**) Bidirectional LSTM's and (**b**) BERT model's loss vs. epoch plots based on the PolitiFact-Oslo Corpus.

Figures 10–14 show the results of the different machine learning model architectures with layers and parameters as well as example results from a fold to show the characteristics of the models on the corpus. The models are: Bidirectional LSTM, fine-tuned BERT (bert-base-uncased), RoBERTa (roberta-base), DistilBERT (distilbert-base-uncased), and XLNet (XLNet-base-cased). The results from the folds include the precision, recall, f1-score, support, and accuracy of the fold. For all the machine learning models, we used 0.001 as the learning rate.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 42, 42)            913584

 dropout (Dropout)           (None, 42, 42)            0

 bidirectional (Bidirection  (None, 200)               114400
 al)

 dropout_1 (Dropout)         (None, 200)               0

 dense (Dense)               (None, 2)                 402

=================================================================
Total params: 1028386 (3.92 MB)
Trainable params: 1028386 (3.92 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

(a)

	Precision	Recall	F1- Score	Support
True	0.45	0.35	0.39	98
False	0.86	0.91	0.89	451
Accuracy			**0.81**	549
Avg	0.66	0.63	0.64	549
Weighted Avg	0.79	0.81	0.8	549

(b)

Figure 10. (**a**) Bidirectional LSTM architecture (**b**) results using that on the PolitiFact-Oslo Corpus for a fold.

Figure 15 shows the confusion matrix of a fold during cross-fold validation experiments of two classes with the BERT-based model on the corpus. It can be observed in Figure 15 that during two-class experiments, texts from the True class were highly confused with texts from the False class due to misclassification. By contrast, most of the texts in the False class were well-classified with less confusion in the True class. Figure 16 shows the bidirectional LSTM architecture and results in detail using that on the PolitiFact-Oslo Corpus for a fold of six classes, while Figure 17 shows the LSTM-based fine-tuned BERT architecture and results using that on the same fold. We followed similar layers for RoBERTa, DistilBERT, and XLNet. Figure 18 shows the results of the same fold using RoBERTa, DistilBERT, and XLNet, respectively. Figure 19 represents the confusion matrix of a fold from cross-fold validation experiments of six classes with the BERT-based model. It can be observed in Figure 19 that when the True and False classes were further divided into sub-classes, the performance of the machine learning algorithm decreased, with increased confusion among the sub-classes. The overall accuracies from the figures confirm that the transformer-based models have better accuracies (e.g., 43% using XLNet) and recall rates (e.g., 23.65% using RoBERTa) compared to others (e.g., 26% accuracy and 20.98% recall

rates using Bidirectional LSTM). Thus, all the experimental results reported in this subsection point to the fact that the recommended machine learning models (i.e., LSTM-based fine-tuned transformers) of this study generate better performance than any of the other traditional approaches (DBN, typical LSTM, Bidirectional LSTM).

```
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_ids (InputLayer)       [(None, 42)]              0         []

attention_mask (InputLayer   [(None, 42)]              0         []
)

tf_bert_model (TFBertModel   TFBaseModelOutputWithPooli 109482240 ['input_ids[0][0]',
)                            ngAndCrossAttentions(last_           'attention_mask[0][0]']
                             hidden_state=(None, 42, 76
                             8),
                              pooler_output=(None, 768)
                             , past_key_values=None, hi
                             dden_states=None, attentio
                             ns=None, cross_attentions=
                             None)

lstm_1 (LSTM)                (None, 128)               459264    ['tf_bert_model[0][0]']

batch_normalization (Batch   (None, 128)               512       ['lstm_1[0][0]']
Normalization)

dense_1 (Dense)              (None, 64)                8256      ['batch_normalization[0][0]']

dropout_39 (Dropout)         (None, 64)                0         ['dense_1[0][0]']

outputs (Dense)              (None, 2)                 130       ['dropout_39[0][0]']

==================================================================================================
Total params: 109950402 (419.43 MB)
Trainable params: 467906 (1.78 MB)
Non-trainable params: 109482496 (417.64 MB)
```

(a)

	Precision	Recall	F1- Score	Support
True	0.56	0.3	0. 39	98
False	0.86	0.95	0.9	451
Accuracy			0.83	549
Avg	0. 71	0.62	0. 64	549
Weighted Avg	0.81	0.83	0.81	549

(b)

Figure 11. (**a**) LSTM fine-tuned BERT architecture (**b**) results using BERT on the PolitiFact-Oslo Corpus for a fold.

4.2.2. Full DeClarE Dataset

Next, we turned to the DeClarE dataset. DeClarE contains 15,018 True and 14,537 False texts, so the dataset is larger and more balanced than the PolitiFact-Oslo Corpus in its current form. The fine-grained labels are also distributed more evenly: True (4393), Half True (5466), Mostly True (5159), False (6052), Mostly False (5024), and Pants on Fire (3461). Moreover, DeClarE does not provide access to the complete texts but rather to the snippets of the complete texts around the claim in question. The snippets are clearly longer than the claims themselves and therefore are more suited for text classification based on linguistic characteristics. Access to the snippets only means that the texts do not contain any irrelevant

information in the rest of the news item that might negatively affect the experimental results; this is a concern when complete texts are used.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 42)]	0	[]
attention_mask (InputLayer)	[(None, 42)]	0	[]
tf_roberta_model (TFRobertaModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 42, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	124645632	['input_ids[0][0]', 'attention_mask[0][0]']
lstm_2 (LSTM)	(None, 128)	459264	['tf_roberta_model[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 128)	512	['lstm_2[0][0]']
dense_2 (Dense)	(None, 64)	8256	['batch_normalization_1[0][0]']
dropout_77 (Dropout)	(None, 64)	0	['dense_2[0][0]']
outputs (Dense)	(None, 2)	130	['dropout_77[0][0]']

Total params: 125113794 (477.27 MB)
Trainable params: 467906 (1.78 MB)
Non-trainable params: 124645888 (475.49 MB)

(a)

	Precision	Recall	F1- Score	Support
True	0.49	0.39	0.43	98
False	0. 87	0.91	0.89	451
Accuracy			0.82	549
Avg	0.68	0.65	0.66	549
Weighted Avg	0.81	0.82	6.81	549

(b)

Figure 12. (**a**) LSTM fine-tuned RoBERTa architecture (**b**) results using RoBERTa on the PolitiFact-Oslo Corpus for a fold.

We applied the same cross-fold validation and machine learning approaches to DeClarE as to the PolitiFact-Oslo Corpus. Figure 20 shows the word clouds of the True and False classes in the DeClarE dataset. As can be seen in the figure, the lexical and topical features in DeClarE are very different from those in the PolitiFact-Oslo Corpus due to the different time period covered by the former dataset (pre-2017). (Barack) Obama figures in both types of news, while the fake news sample also includes (Hilary) Clinton and health as major topics. Presumably, the former is due to the various disinformation campaigns launched against Clinton during the 2016 US presidential election [14].

```
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_ids (InputLayer)       [(None, 42)]              0         []

attention_mask (InputLayer   [(None, 42)]              0         []
)

tf_distil_bert_model (TFDi   TFBaseModelOutput(last_hid  6636288   ['input_ids[0][0]',
stilBertModel)               den_state=(None, 42, 768),  0          'attention_mask[0][0]']
                              hidden_states=None, atten
                             tions=None)

lstm_3 (LSTM)                (None, 128)               459264    ['tf_distil_bert_model[0][0]']

batch_normalization_2 (Bat   (None, 128)               512       ['lstm_3[0][0]']
chNormalization)

dense_3 (Dense)              (None, 64)                8256      ['batch_normalization_2[0][0]'
                                                                 ]

dropout_97 (Dropout)         (None, 64)                0         ['dense_3[0][0]']

outputs (Dense)              (None, 2)                 130       ['dropout_97[0][0]']

==================================================================================================
Total params: 66831042 (254.94 MB)
Trainable params: 467906 (1.78 MB)
Non-trainable params: 66363136 (253.16 MB)
```

(a)

	Precision	Recall	F1- Score	Support
True	0.42	0.37	0. 39	98
False	6.87	0. 89	0.88	451
Accuracy			**0.80**	549
Avg	0.64	0.63	0.63	549
Weighted Avg	0. 79	0.8e	0. 79	549

(b)

Figure 13. (**a**) LSTM fine-tuned distilBERT architecture (**b**) results using distilBERT on the PolitiFact-Oslo Corpus for a fold.

Tables 5 and 6 show the results (in average recall rates (%) and their means) of traditional embedding features and various machine learning approaches to the dataset based on two and six classes, respectively. The same architectures as in Section 4.2.1 were applied. In the case of both classes, the fined-tuned transformer models show better performance than the others. Similar to the PolitiFact-Oslo Corpus, two classes provide higher accuracy rates than six classes. Figures 21 and 22 show the results of a single fold applying different machine learning models. Each figure includes the precision, recall, f1-score, support, and accuracy of the fold. Figure 23 shows the loss vs. epoch plots of the Bi-LSTM and BERT models. The plots indicate that the models take more epochs to converge towards lower error rates compared to the models applied to the PolitiFact-Oslo Corpus. The main reason is that the DeClarE dataset contains more texts.

```
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_ids (InputLayer)       [(None, 42)]              0         []

attention_mask (InputLayer   [(None, 42)]              0         []
)

tfxl_net_model (TFXLNetMod   TFXLNetModelOutput(last_hi   1167183   ['input_ids[0][0]',
el)                          dden_state=(None, 42, 768)   36         'attention_mask[0][0]']
                             , mems=((42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768),
                              (42, None, 768)),
                             hidden_states=None, atten
                             tions=None)

lstm_4 (LSTM)                (None, 128)               459264    ['tfxl_net_model[0][0]']

batch_normalization_3 (Bat   (None, 128)               512       ['lstm_4[0][0]']
chNormalization)

dense_4 (Dense)              (None, 64)                8256      ['batch_normalization_3[0][0]'
                                                                 ]

dropout_135 (Dropout)        (None, 64)                0         ['dense_4[0][0]']

outputs (Dense)              (None, 2)                 130       ['dropout_135[0][0]']

==================================================================================================
Total params: 117186498 (447.03 MB)
Trainable params: 467906 (1.78 MB)
Non-trainable params: 116718592 (445.25 MB)
```

(a)

	Precision	Recall	F1- Score	Support
True	0.69	0.24	0.36	98
False	0.86	0.98	0.91	451
Accuracy			**0.85**	549
Avg	0.77	0.61	0. 64	549
Weighted Avg	0.83	0.85	0.81	549

(b)

Figure 14. (**a**) LSTM fine-tuned XLNet architecture (**b**) results using XLNet on the PolitiFact-Oslo Corpus for a fold.

Table 5. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the DeClarE dataset based on two classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	51.29	72.17	76.54	75.23	73.89	76.92	74.15
False	14.27	60.18	70.13	73.19	76.18	73.90	70.57
Mean	**32.78**	**66.17**	**73.35**	**74.21**	**75.06**	**75.41**	**72.36**

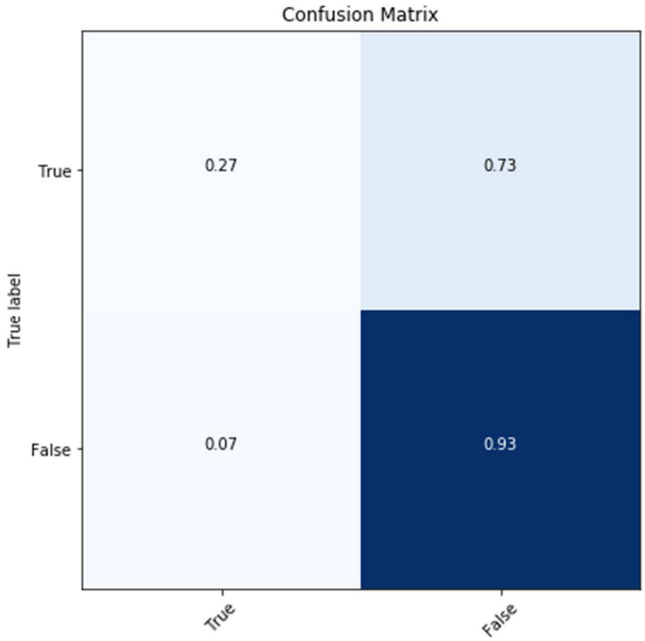


Figure 15. Confusion matrix of test results (BERT) of two classes from a fold.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 42, 42)	913584
dropout (Dropout)	(None, 42, 42)	0
bidirectional (Bidirectional)	(None, 200)	114400
dropout_1 (Dropout)	(None, 200)	0
dense (Dense)	(None, 6)	1206

Total params: 1029190 (3.93 MB)
Trainable params: 1029190 (3.93 MB)
Non-trainable params: 0 (0.00 Byte)

(a)

	Precision	Recall	F1- Score	Support
True	0. 00	0. 00	0. 00	25
Half True	0.09	0.06	0.07	51
Mostly True	0. 00	0. 00	0. 00	25
False	0.51	0.26	0.34	277
Mostly False	0.15	0.21	0.17	71
Pants on Fire	0.21	0. 55	0.3	100
Accuracy			**0.26**	549
Avg	0.16	0. 18	0.15	549
Weighted Avg	0.32	0. 26	0.26	549

(b)

Figure 16. (**a**) Bidirectional LSTM architecture (**b**) results using Bidirectional-LSTM on the PolitiFact-Oslo Corpus for a fold.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 42)]	0	[]
attention_mask (InputLayer)	[(None, 42)]	0	[]
tf_bert_model (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 42, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	109482240	['input_ids[0][0]', 'attention_mask[0][0]']
lstm_1 (LSTM)	(None, 128)	459264	['tf_bert_model[0][0]']
batch_normalization (BatchNormalization)	(None, 128)	512	['lstm_1[0][0]']
dense_1 (Dense)	(None, 64)	8256	['batch_normalization[0][0]']
dropout_39 (Dropout)	(None, 64)	0	['dense_1[0][0]']
outputs (Dense)	(None, 6)	390	['dropout_39[0][0]']

Total params: 109950662 (419.43 MB)
Trainable params: 468166 (1.79 MB)
Non-trainable params: 109482496 (417.64 MB)

(a)

	Precision	Recall	F1- Score	Support
True	0. 00	0. 00	0. 00	25
Half True	0.15	0.12	0.13	51
Mostly True	0.08	0. 04	0.05	25
False	0. 53	0. 66	0.59	277
Mostly False	0. 24	0. 14	0.18	71
Pants on Fire	0.26	0.27	0.27	100
Accuracy			**0.42**	549
Avg	0.21	0.21	0.2	549
Weighted Avg	0.37	0. 42	0.39	549

(b)

Figure 17. (**a**) LSTM fine-tuned BERT (**b**) results using BERT on the PolitiFact-Oslo Corpus for a fold.

	Precision	Recall	F1- Score	Support
True	0. 50	0.04	0.07	25
Half True	0. 20	0. 25	0.22	51
Mostly True	0.33	0.08	0.13	25
False	0. 54	0.65	0.59	277
Mostly False	0.26	0. 11	0.16	71
Pants on Fire	0.22	0.25	0.24	100
Accuracy			**0.42**	549
Avg	0. 34	0.23	0.23	549
Weighted Avg	0. 40	0.42	0.39	549

(a)

	Precision	Recall	F1- Score	Support
True	0. 50	0.04	0.07	25
Half True	0. 20	0. 25	0.22	51
Mostly True	0.33	0.08	0.13	25
False	0. 54	0.65	0.59	277
Mostly False	0.26	0. 11	0.16	71
Pants on Fire	0.22	0.25	0.24	100
Accuracy			**0.42**	549
Avg	0. 34	0.23	0.23	549
Weighted Avg	0. 40	0.42	0.39	549

(b)

	Precision	Recall	F1- Score	Support
True	0. 00	0. 00	0. 00	25
Half True	0.21	0.06	0.09	51
Mostly True	0.09	0.12	0.1	25
False	0.53	0.72	0.61	277
Mostly False	0.29	0.13	0.18	71
Pants on Fire	0.26	0.23	0. 24	100
Accuracy			**0.43**	549
Avg	0.23	0. 21	0.2	549
Weighted Avg	0.38	0.43	0.39	549

(c)

Figure 18. Results using LSTM fine-tuned (**a**) RoBERTa, (**b**) DistilBERT, and (**c**) XLNet on the PolitiFact-Oslo Corpus for a fold.

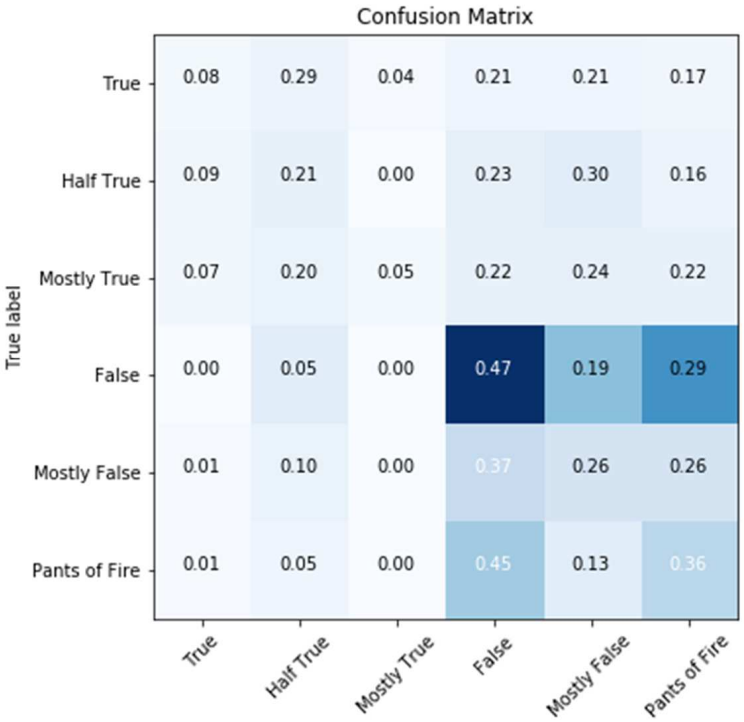
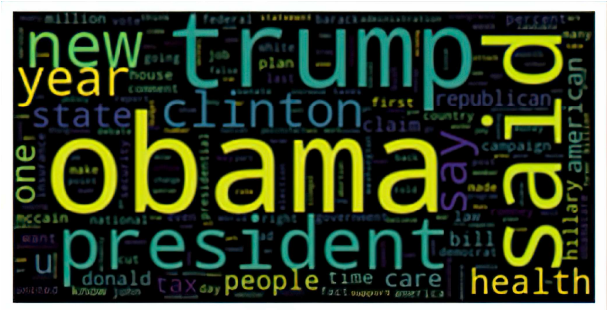


Figure 19. Confusion matrix of test results (BERT) of six classes from a fold.

(**a**)



(**b**)

Figure 20. Word clouds for (**a**) True and (**b**) False classes in the DeClarE dataset.

	Precision	Recall	F1- Score	Support
True	0.73	0.76	0.75	6043
False	0. 74	0.7	0.72	5779
Accuracy			**0.73**	11822
Avg	0.73	0.73	0.73	11822
Weighted Avg	0.73	0.73	0.73	11822

(a)

	Precision	Recall	F1- Score	Support
True	0.73	0.74	0.74	6043
False	0.73	0.72	0.72	5779
Accuracy			**0.73**	11822
Avg	0.73	0.73	0.73	11822
Weighted Avg	0.73	0.73	0.73	11822

(b)

	Precision	Recall	F1- Score	Support
True	0.75	0.72	0.73	6043
False	0.72	0.75	0.73	5779
Accuracy			**0.73**	11822
Avg	0.73	0.73	0.73	11822
Weighted Avg	0.73	0.73	0.73	11822

(c)

	Precision	Recall	F1- Score	Support
True	0.74	0.75	0.74	6043
False	0.73	0.73	0.73	5779
Accuracy			**0.74**	11822
Avg	0.74	0.74	0.74	11822
Weighted Avg	0.74	0.74	0.74	11822

(d)

	Precision	Recall	F1- Score	Support
True	0.72	0.74	0.73	6043
False	0.73	0.71	0.72	5779
Accuracy			**0.73**	11822
Avg	0.73	0.73	0.73	11822
Weighted Avg	0.73	0.73	0.73	11822

(e)

Figure 21. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the DeClarE dataset for a fold for two classes.

	Precision	Recall	F1- Score	Support
True	0.53	0. 52	0. 52	1777
Half True	0.53	0. 52	0. 53	2196
Mostly True	0.52	0.51	0.51	2052
False	0. 56	0. 55	0.55	2369
Mostly False	0.55	0.54	0.54	2040
Pants on Fire	0. 52	0.61	0. 56	1388
Accuracy			**0. 54**	11822
Avg	0.54	0.54	0.54	11822
Weighted Avg	0. 54	0. 54	0.54	11822

(a)

	Precision	Recall	F1- Score	Support
True	0.57	0.47	0.51	1777
Half True	0.54	0.51	0.52	2196
Mostly True	0.45	0.58	0.51	2052
False	0.58	0.54	0.56	2369
Mostly False	0.54	0.54	0.54	2040
Pants on Fire	0.59	0.58	0.59	1388
Accuracy			0.54	11822
Avg	0.54	0.54	0.54	11822
Weighted Avg	0.54	0.54	0.54	11822

(b)

	Precision	Recall	F1- Score	Support
True	0.57	0.47	0.52	1777
Half True	0.55	0.49	0.51	2196
Mostly True	0.47	0.55	0.51	2052
False	0.53	0.6	0.56	2369
Mostly False	0.53	0.53	0.53	2040
Pants on Fire	0.58	0.56	0.57	1388
Accuracy			0.53	11822
Avg	0.54	0.53	0.53	11822
Weighted Avg	0.54	0.53	0.53	11822

(c)

	Precision	Recall	F1- Score	Support
True	0.5	0.55	0.52	1777
Half True	0.49	0.58	0.53	2196
Mostly True	0.52	0.46	0.49	2052
False	0.54	0.58	0.56	2369
Mostly False	0.56	0.48	0.52	2040
Pants on Fire	0.63	0.52	0.57	1388
Accuracy			0.53	11822
Avg	0.54	0.53	0.53	11822
Weighted Avg	0.54	0.53	0.53	11822

(d)

	Precision	Recall	F1- Score	Support
True	0.52	0.5	0.51	1777
Half True	0.53	0.53	0.53	2196
Mostly True	0.47	0.56	0.51	2052
False	0.55	0.56	0.55	2369
Mostly False	0.61	0.48	0.54	2040
Pants on Fire	0.56	0.57	0.56	1388
Accuracy			0.53	11822
Avg	0.54	0. 53	0.53	11822
Weighted Avg	0.54	0. 53	0.53	11822

(e)

Figure 22. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the DeClarE dataset for a fold for six classes.

Table 6. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the DeClarE dataset based on six classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	23.61	40.181	54.14	57.12	47.93	55.18	51.02
Half True	30.23	50.02	52.71	54.24	50.09	57.97	54.83
Mostly True	24.12	45.71	51.74	45.91	55.18	47.18	56.18
False	50.18	51.17	55.18	55.08	61.18	59.91	57.15
Mostly False	40.23	45.30	55.48	54.94	54.11	48.14	49.61
Pants on Fire	33.12	48.58	52.17	60.03	57.18	52.13	53.02
Mean	**33.58**	**48.82**	**53.57**	**54.55**	**54.27**	**53.41**	**53.64**

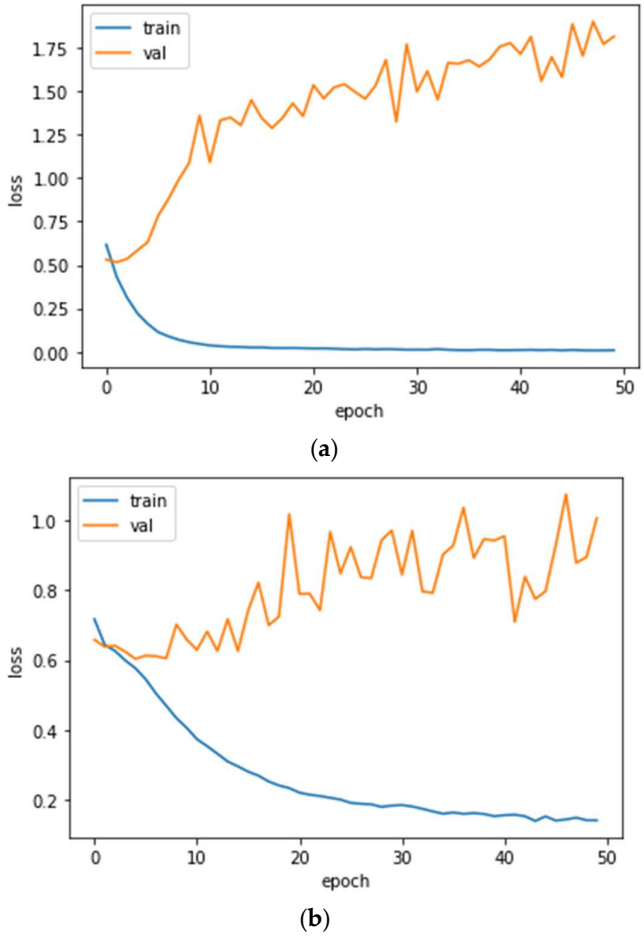


Figure 23. (**a**) Bidirectional LSTM's and (**b**) BERT model's loss vs. epoch plot based on the DeClarE dataset.

Overall, the accuracy rates from the machine learning models are comparable across the PolitiFact-Oslo Corpus and the DeClarE dataset, with the latter showing slightly better performance. However, we are hesitant to use accuracy rates as a point of comparison between the datasets, due to differences in the sizes of the datasets as well as the heavy class imbalances in the PolitiFact-Oslo Corpus (but see Section 5 for a possible solution). Therefore, we now turn to a feature of our corpus that is not present in DeClarE and that provides a more appropriate point of comparison, namely, access to information about where the news items come from, i.e., their text type.

4.2.3. News and Blog

As previously mentioned, the two largest text types in the PolitiFact-Oslo Corpus are news and blog, and social media. The two text types differ from each other in important ways pertaining mainly to conventional features such as specialized expressions, rhetorical organization and formatting, as well as aspects related to variation in the use of linguistic features [16]. To gauge these differences, we have carried out separate analyses based on the text types. Figures 24 and 25 report the results of a fold applying different machine learning models. Each figure includes the precision, recall, f1-score, support, and accuracy of the fold. Starting with news and blog, Tables 7 and 8 show the results of traditional embedding features and various machine learning approaches to the text type based on two and six classes, respectively. Again, it is the fine-tuned transformer models that show better performance, most clearly when applied to two classes compared to six classes. In both cases, however, the accuracy rates are slightly lower than for the full dataset.

	Precision	Recall	F1- Score	Support
True	0	0	0	10
False	0.89	0. 98	0.93	83
Accuracy			**0.87**	93
Avg	0.45	0. 49	0.47	93
Weighted Avg	0.79	0.87	0.83	93

(a)

	Precision	Recall	F1- Score	Support
True	0.12	0.1	0.11	10
False	6.89	0.92	0.9	83
Accuracy			**0.83**	93
Avg	0.51	0.51	0.51	93
Weighted Avg	0.81	0.83	0.82	93

(b)

	Precision	Recall	F1- Score	Support
True	0.29	0.2	0.24	10
False	0.91	0.94	0.92	83
Accuracy			**0.86**	93
Avg	0.6	0.57	0.58	93
Weighted Avg	0.84	0.86	0.85	93

(c)

	Precision	Recall	F1- Score	Support
True	0.2	0.1	0.13	10
False	0.9	0.95	0.92	83
Accuracy			**0.86**	93
Avg	0.55	0.53	0.53	93
Weighted Avg	0.82	0.86	0.84	93

(d)

	Precision	Recall	F1- Score	Support
True	0.17	0.4	0.24	10
False	0.91	0.77	0.84	83
Accuracy			**0.73**	93
Avg	0.54	0.59	0.54	93
Weighted Avg	0.83	0.73	0.77	93

(e)

Figure 24. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the PolitiFact-Oslo Corpus (news and blog) for a fold for two classes.

	Precision	Recall	F1- Score	Support
True	0	0	0	4
Half True	0	0	0	5
Mostly True	0	0	0	3
False	0.2	0.05	0.08	42
Mostly False	0.12	0.08	0.1	12
Pants on Fire	0.28	0.74	0.41	27
Accuracy			**0.25**	93
Avg	0.1	0.15	0.1	93
Weighted Avg	0.19	0.25	0.17	93

(a)

	Precision	Recall	F1- Score	Support
True	0	0	0	4
Half True	0	0	0	5
Mostly True	0	0	0	3
False	0.3	0.04	0.08	42
Mostly False	0.12	0.08	0.1	12
Pants on Fire	0.28	0.74	0.41	27
Accuracy			**0.25**	93
Avg	0.1	0.15	0.1	93
Weighted Avg	0.19	0.25	0.17	93

(b)

	Precision	Recall	F1- Score	Support
True	0	0	0	4
Half True	0.2	0.2	0.2	5
Mostly True	0	0	0	3
False	0	0.74	0.56	42
Mostly False	0.2	0.08	0.12	12
Pants on Fire	0.31	0.15	0.2	27
Accuracy			**0.4**	93
Avg	0.19	0.19	0.18	93
Weighted Avg	0.33	0.4	0.34	93

(c)

	Precision	Recall	F1- Score	Support
True	0	0	0	4
Half True	0	0	0	5
Mostly True	0	0	0	3
False	0.47	0.83	0.6	42
Mostly False	0	0	0	12
Pants on Fire	0.29	0.15	0.2	27
Accuracy			**0.42**	93
Avg	0.13	0.16	0.13	93
Weighted Avg	0.29	0.42	0.33	93

(d)

	Precision	Recall	F1- Score	Support
True	0	0	0	4
Half True	0	0	0	5
Mostly True	0.5	0.33	0.4	3
False	0.5	0.45	0.48	42
Mostly False	0.12	0.08	0.1	12
Pants on Fire	0.36	0.59	0.44	27
Accuracy			**0.4**	93
Avg	0.25	0.24	0.24	93
Weighted Avg	0.36	0.4	0.37	93

(e)

Figure 25. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the PolitiFact-Oslo Corpus (news and blog) for a fold for six classes.

Table 7. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the PolitiFact-Oslo Corpus (news and blog) based on two classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	19.03	22.19	13.13	18.58	25.51	14.13	25.61
False	60.15	84.18	95.13	94.17	95.62	95.61	86.57
Mean	**39.60**	**53.19**	**54.14**	**56.37**	**60.56**	**54.87**	**56.09**

Table 8. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the PolitiFact-Oslo Corpus (news and blog) based on six classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	0.02	0.76	1.21	0.87	2.15	1.69	0.98
Half True	0.74	2.37	2.91	2.26	21.25	2.13	1.56
Mostly True	0.25	1.18	3.58	5.47	3.87	2.15	34.15
False	12.71	11.35	10.15	8.13	74.76	83.97	46.12
Mostly False	8.77	6.02	09.22	10.09	10.72	3.61	9.16
Pants on Fire	25.63	27.28	73.97	73.91	17.21	16.15	59.69
Mean	**8.02**	**8.16**	**16.84**	**16.78**	**21.66**	**18.28**	**25.27**

4.2.4. Social Media

When applied to social media, we can observe some improvement in terms of the performance of the experiments. Figures 26 and 27 report the results of a fold applying different machine learning models. Each figure includes the precision, recall, f1-score, support, and accuracy of the fold. Tables 9 and 10 show the results of traditional embedding features and various machine learning approaches to the text type based on two and six classes, respectively. Here, almost all the models show better performance than for the full dataset or for news and blog separately. In fact, the combined model shows the highest accuracy obtained for the PolitiFact-Oslo Corpus so far. This might be explained by the greater homogeneity of the social media posts, most of which are from a handful of platforms (e.g., Facebook, Twitter, and Instagram). News articles and blog posts, by contrast, are more diverse in terms of publishers and therefore their lexical and topical features may be more difficult to capture.

	Precision	Recall	F1- Score	Support
True	0.38	0.22	0.28	81
False	0.85	0.92	0.88	347
Accuracy			**0.8**	456
Avg	0.61	0.57	0.58	456
Weighted Avg	0.76	0.8	0.77	456

(a)

	Precision	Recall	F1- Score	Support
True	0.5	0.33	0.4	81
False	0.87	0.93	0.9	347
Accuracy			**0.82**	456
Avg	0.68	0.63	0.65	456
Weighted Avg	0.8	0.82	0.81	456

(b)

	Precision	Recall	F1- Score	Support
True	0.43	0.4	0.41	81
False	0.87	0.89	0.88	347
Accuracy			**0.8**	456
Avg	0.65	0.64	0.64	456
Weighted Avg	0.79	0.8	0.8	456

(c)

	Precision	Recall	F1- Score	Support
True	0.51	0.36	0.42	81
False	0.87	0.93	0.9	347
Accuracy			**0.82**	456
Avg	0.69	0.64	0.66	456
Weighted Avg	0.81	0.82	0.81	456

(d)

	Precision	Recall	F1- Score	Support
True	0.36	0.41	0.38	81
False	0.87	0.85	0.86	347
Accuracy			**0.77**	456
Avg	0.62	0.63	0.62	456
Weighted Avg	0.78	0.77	0.77	456

(e)

Figure 26. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the PolitiFact-Oslo Corpus (social media) for a fold for two classes.

	Precision	Recall	F1- Score	Support
True	0	0	0	13
Half True	0.2	0.03	0.05	39
Mostly True	0.33	0.03	0.05	35
False	0.54	0.66	0.59	229
Mostly False	0.18	0.35	0.23	54
Pants on Fire	6.28	0.19	0.22	86
Accuracy			**0.41**	456
Avg	0.25	0.21	0.19	456
Weighted Avg	0.38	0.41	0.37	456

(a)

	Precision	Recall	F1- Score	Support
True	0.25	0.23	0.24	13
Half True	0.16	0.1	0.12	39
Mostly True	0.14	0.03	0.05	35
False	0.53	0.72	0.61	229
Mostly False	0.12	0.09	0.1	54
Pants on Fire	0.35	0.26	0.3	86
Accuracy			**0.44**	456
Avg	0.26	0.24	0.24	456
Weighted Avg	0.38	0.44	0.4	456

(b)

	Precision	Recall	F1- Score	Support
True	0	0	0	13
Half True	0.11	0.05	0.07	39
Mostly True	0.17	0.14	0.15	35
False	0.54	0.65	0.59	229
Mostly False	0.08	0.11	0.09	54
Pants on Fire	0.19	0.12	0.14	86
Accuracy			**0.38**	456
Avg	0.18	0.18	0.17	456
Weighted Avg	0.34	0.38	0.35	456

(c)

	Precision	Recall	F1- Score	Support
True	0.08	0	0	13
Half True	0.04	0.03	0.03	39
Mostly True	0.14	0.03	0.05	35
False	0.55	0.85	0.66	229
Mostly False	0.35	0.2	0.26	54
Pants on Fire	0.22	0.08	0.12	86
Accuracy			**0.47**	456
Avg	0.22	0.2	0.19	456
Weighted Avg	0.37	0.47	0.39	456

(d)

	Precision	Recall	F1- Score	Support
True	0.12	0.23	0.15	13
Half True	0.21	0.18	0.19	39
Mostly True	0.2	0.03	0.05	35
False	0.55	0.71	0.62	229
Mostly False	0.19	0.13	0.16	54
Pants on Fire	0.24	0.16	0.19	86
Accuracy			**0.43**	456
Avg	0.25	0.24	0.23	456
Weighted Avg	0.38	0.43	0.39	456

(e)

Figure 27. Results using (**a**) Bidirectional LSTM, (**b**) LSTM fine-tuned BERT, (**c**) RoBERTa, (**d**) DistilBERT, and (**e**) XLNet on the PolitiFact-Oslo Corpus (social media) for a fold for six classes.

Table 9. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the PolitiFact-Oslo Corpus (social media) based on two classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	18.23	25.16	26.27	33.81	41.71	36.47	40.15
False	77.14	90.06	91.37	93.96	90.37	93.19	85.91
Mean	**47.69**	**57.61**	**58.82**	**63.88**	**66.04**	**64.83**	**63.03**

Table 10. Experimental results (recall rates in %) of traditional embedding features and various machine learning approaches to the PolitiFact-Oslo Corpus (social media) based on six classes.

	DBN	LSTM	Bidirectional LSTM	BERT	RoBERTa	DistilBERT	XLNet
True	10.92	11.47	15.44	23.18	5.14	4.61	24.47
Half True	12.17	15.79	17.29	14.61	8.61	3.17	19.28
Mostly True	2.19	3.23	10.01	7.13	15.17	3.83	4.15
False	38.14	41.16	49.01	72.15	65.37	86.19	72.32
Mostly False	18.17	21.27	25.09	10.31	11.15	21.52	15.81
Pants on Fire	17.28	18.84	20.14	27.51	12.57	10.15	16.27
Mean	**16.47**	**18.63**	**22.83**	**25.81**	**19.66**	**21.58**	**25.38**

In summary, the fact that we obtain different results for the full dataset vs. the different text types is an indication that the language of fake news is not the same across all situations and contexts of use. Also, the lack of access to information about text type in other fake news datasets such as DeClarE means that what is being captured by the machine learning algorithms may not be the difference between fake and real news, but rather a difference between the text types that dominate each of the news samples. Therefore, for the best result possible, text type variation needs to be built into future machine learning models to ensure that we compare like with like: real social media posts with fake social media posts, and so on. With this study, we have taken an important step forward in this regard. In addition to featuring important metadata information, the PolitiFact-Oslo Corpus shows improvement in terms of several other aspects of corpus design, in line with the pipeline developed in this study (see Figure 1). For example, thanks to manual work, the corpus does not include any texts that do not strictly correspond to the claim in question, whereas the automatic procedures used in DeClarE may have let such texts seep in. Finally, a lot has changed in the digital news media landscape since many of the PolitiFact datasets were collected, as suggested by the lexical and topical differences between DeClarE and the PolitiFact-Oslo Corpus in Sections 4.2.1 and 4.2.2. Malign actors, both human and artificial, have become increasingly good at simulating the speech and style of real news, potentially narrowing the window of opportunity for finding any differences at all. The more recent the data, the more likely we are to keep up with developments in this area.

5. Conclusions

This study has presented the PolitiFact-Oslo Corpus, a new dataset of fake and real news in English based on recent events, and critically examined its suitability for fake news analysis and detection model development by means of a series of case studies in natural language processing and machine learning. The case studies in machine learning were applied both to the PolitiFact-Oslo Corpus and to an existing PolitiFact dataset, namely, DeClarE. While DeClarE shows several advantages over our own corpus, it has nevertheless not been built based on strict criteria. These criteria were presented in this study in the form of a pipeline for collecting quality data from major fact-checking websites such as PolitiFact, which may be used in future corpus building efforts. Besides being individually labelled for veracity by experts, in the PolitiFact-Oslo Corpus the complete texts correspond strictly to the claims in question and are accompanied by important metadata information, thus ensuring a more controlled and effective dataset. Access to information about text type is particularly crucial for valid and reliable outcomes since this kind of variation is pervasive in natural language.

In the case studies, we identified interesting differences between fake and real news in terms of sentiment and POS information. Fake news was characterized by more negative sentiment and a greater use of pronouns, which contribute to a more emotional and informal style of this type of news. However, these features varied across the text types of social media, and news and blog, thus highlighting the importance of adding contextual information to fake news corpora. Access to information about text type also turned out to be an important feature of the PolitiFact-Oslo Corpus in the machine learning models applied in this study; the accuracy rates were different in the case of the full corpus and when the text types were considered separately. The lack of ready access to information about where the news items came from in other fake news datasets, including DeClarE, is a potential limitation that may negatively affect the automatic fake news detection models trained on the data. The PolitiFact-Oslo Corpus meets this requirement. As for the machine learning models that we applied, the LSTM fined-tuned transformer models generally showed better performance over non-transformer-based models by achieving higher accuracy rates. However, at this point, the use of the models only pays off in relation to the broad distinction between fake and real news, rather than to finer-grained distinctions. Therefore, a more powerful combination of various ensemble models based on larger sample sizes will have to be explored in the future.

This study has brought to light several limitations and directions for the future development of the PolitiFact-Oslo Corpus. Firstly, it has drawn attention to the relatively uneven distribution of fake and real news in the corpus due to the preference for PolitiFact and other fact-checkers to debunk false information rather than to find support for true information. One solution that we are currently implementing is to extend the scope of fact-checkers (as found in Google's Fact Check Explorer) to find more instances of True news without having to stretch out the timeline. Secondly, the case studies are just a taster of what can be performed with the PolitiFact-Oslo Corpus. With a focus on text and the language of fake news in the strict sense, the corpus provides an excellent resource for the analysis of specific linguistic features of fake news. Early models based on grammatical features and relations from linguistic theory are showing promising results. Finally, due to ethical and legal restrictions, the corpus texts are currently available upon request only. Inspired by the DeClarE dataset, however, we are exploring opportunities to release the text snippets via an online interface.

Author Contributions: All authors worked on the concept of the work. More specifically N.P. and A.T. worked on the dataset. Z.U. worked on machine learning experiments and reporting. Finally, all authors revised the work. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by The Research Council of Norway under the project-ID 302573.

Data Availability Statement: Dataset used in this work is available based on the request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Newman, N.; Fletcher, R.; Schulz, A.; Andi, S.; Robertson, C.T.; Nielsen, R.K. *Reuters Institute Digital News Report 2021*; Reuters Institute: Oxford, UK, 2021.
2. Capuano, N.; Fenza, G.; Loia, V.; Nota, F.D. Content-based fake news detection with machine and deep learning: A systematic review. *Neurocomputing* **2023**, *530*, 91–103. [CrossRef]
3. Conroy, N.K.; Rubin, V.L.; Chen, Y. Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting, St. Louis, MO, USA, 6–10 November 2015; Association for Information Science and Technology: St. Louis, MO, USA, 2015; pp. 1–4.
4. Ibrishimova, M.D.; Li, K. A machine learning approach to fake news detection using knowledge verification and natural language processing. In *Advances in Intelligent Networking and Collaborative Systems, INCoS 2019*; Barolli, L., Nishino, H., Miwa, H., Eds.; Springer: Cham, Switzerland, 2020; Volume 1035, pp. 223–234.
5. Oshikawa, R.; Qian, J.; Wang, W.Y. A survey of natural language processing for fake news detection. In Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), Marseille, France, 11–16 May 2018; European Language Resources Association: Marseille, France, 2018; pp. 6086–6093.
6. Villela, H.F.; Correa, F.; Ribeiro, J.S.A.N.; Rabelo, A.; Carvalho, D.B.F. Fake news detection: A systematic literature review of machine learning algorithms and datasets. *J. Interact. Syst.* **2023**, *14*, 47–58. [CrossRef]
7. Rashkin, H.; Choi, E.; Jang, J.Y.; Volkova, S.; Choi, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 2931–2937.
8. Volkova, S.; Shaffer, K.; Jang, J.Y.; Hodas, N. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 647–653.
9. Siino, M.; Di Nuovo, E.; Tinnirello, I.; La Cascia, M. Fake News Spreaders Detection: Sometimes Attention Is Not All You Need. *Information* **2022**, *13*, 426. [CrossRef]
10. Vlachos, A.; Riedel, S. Fact checking: Task definition and dataset construction. In Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, Baltimore, MD, USA, 26 June 2014; Association for Computational Linguistics: Baltimore, MD, USA, 2014; pp. 18–22.
11. Ferreira, W.; Vlachos, A. Emergent: A novel data-set for stance classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: San Diego, CA, USA, 2016; pp. 1163–1168.
12. Wang, W.Y. "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 422–426.

13. Asr, F.T.; Taboada, M. Big Data and quality data for fake news and misinformation detection. *Big Data Soc.* **2019**, *6*, 3310. [CrossRef]
14. Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. *J. Econ. Perspect.* **2017**, *31*, 211–236. [CrossRef]
15. Popat, K.; Mukherjee, S.; Yates, A.; Weikum, G. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 22–32.
16. Grieve, J.; Woodfield, H. *The Language of Fake News*; Cambridge University Press: Cambridge, UK, 2023.
17. Subba, B.; Kumari, S. A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Comput. Intell.* **2021**, *38*, 530–559. [CrossRef]
18. Rodriguez, P.L.; Spirling, A. Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Applied Research. *J. Polit.* **2022**, *84*, 101–115. [CrossRef]
19. Mangione, S.; Siino, M.; Garbo, G. Improving Irony and Stereotype Spreaders Detection using Data Augmentation and Convolutional Neural Network. In *CEUR Workshop Proceedings*; Università degli Studi di Palermo, Dipartimento di Ingegneria: Palermo, Italy, 2022; Volume 3180, pp. 2585–2593.
20. Saleh, H.; Alhothali, A.; Moria, K. Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model. *Appl. Artif. Intell.* **2023**, *37*, 2166719. [CrossRef]
21. Daniele, C.; Garlisi, D.; Siino, M. An SVM Ensamble Approach to Detect Irony and Stereotype Spreaders on Twitter. In *CEUR Workshop Proceedings*; Sun SITE Central Europe: Aachen, Germany, 2022; Volume 3180.
22. Incitti, F.; Urli, F.; Snidaro, L. Beyond word embeddings: A survey. *Inf. Fusion* **2023**, *89*, 418–436. [CrossRef]
23. Espinosa, D.; Sidorov, G. Using BERT to profiling cryptocurrency influencers. In *Working Notes of CLEF*; Sun SITE Central Europe: Aachen, Germany, 2023.
24. Biber, D. *Variation across Speech and Writing*; Cambridge University Press: Cambridge, UK, 1988.
25. Association for Progressive Communications: Disinformation and Freedom of Expression. 2021. Available online: https://www.apc.org/sites/default/files/APCSubmissionDisinformationFebruary2021.pdf (accessed on 2 November 2023).
26. Sousa-Silva, R. Fighting the fake: A forensic linguistic analysis to fake news detection. *Int. J. Semiot. Law* **2022**, *35*, 2409–2433. [CrossRef] [PubMed]
27. Nakamura, K.; Levy, S.; Wang, W.Y. r/Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. In Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), Marseille, France, 11–16 May 2018; European Language Resources Association: Marseille, France, 2020; pp. 6149–6157.
28. Hutto, C.; Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the International AAAI Conference on Web and Social Media, Ann Arbor, MI, USA, 1–4 June 2014; Volume 8.
29. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
30. Markowitz, D.M.; Hancock, J.T. Linguistic traces of a scientific fraud: The case of Diederik Stapel. *PLoS ONE* **2014**, *9*, e105937. [CrossRef] [PubMed]
31. Uddin, Z. *Applied Machine Learning for Assisted Living*; Springer Science and Business Media LLC: Dordrecht, The Netherlands, 2022. [CrossRef]
32. Uddin, Z.; Hassan, M.M.; Alsanad, A.; Savaglio, C. A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare. *Inf. Fusion* **2020**, *55*, 105–115. [CrossRef]
33. Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the Real World: A Survey on NLP Applications. *Information* **2023**, *14*, 242. [CrossRef]
34. Masciari, E.; Moscato, V.; Picariello, A.; Sperli, G. A deep learning approach to fake news detection. In Proceedings of the Foundations of Intelligent Systems: 25th International Symposium, ISMIS 2020, Graz, Austria, 23–25 September 2020; Springer International Publishing: Berlin/Heidelberg, Germany, 2020.
35. Konkobo, P.M.; Zhang, R.; Huang, S.; Minoungou, T.T.; Ouedraogo, J.A.; Li, L. A deep learning model for early detection of fake news on social media. In Proceedings of the 2020 7th International Conference on Behavioural and Social Computing (BESC), Bournemouth, UK, 5–7 November 2020.
36. Alghamdi, J.; Lin, Y.; Luo, S. A Comparative Study of Machine Learning and Deep Learning Techniques for Fake News Detection. *Information* **2022**, *13*, 576. [CrossRef]
37. Palani, B.; Elango, S.; Viswanathan, K.V. CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT. *Multimed. Tools Appl.* **2022**, *81*, 5587–5620. [CrossRef] [PubMed]
38. Ali, A.M.; Ghaleb, F.A.; Al-Rimy, B.A.S.; Alsolami, F.J.; Khan, A.I. Deep Ensemble Fake News Detection Model Using Sequential Deep Learning Technique. *Sensors* **2022**, *22*, 6970. [CrossRef] [PubMed]