Fake News Detection
Project 2 Proposal for NLP Course, Winter 2022

Marcel Affi
marcel.affi98@gmail.com

Mateusz Wójcik
mateusz.wojcik5.stud@pw.edu.pl

Arkadiusz Zalas
arkadiusz.zalas.stud@pw.edu.pl

Abstract

In this report, we describe our further research towards solving a problem of fake news detection which nowadays is a large issue. We explore already existing solutions for misinformation related problems and present datasets we will work with. Our aim is to compare different approaches by expanding basic preprocessing methods for binary and multi-class fake news detection, consider other inputs of data and testing a few selected architectures.

1 Introduction

This project aims to detect news that contain misinformation, particularly fake news, based on e.g. online article texts. Since nowadays anyone can run a blog and write whatever they please, it is crucial to be able to differentiate between so-called real and fake news. We will test and compare different methods for misinformation detection (MID) using multiple datasets.

2 Background and Goals

Before introducing the existing research and the proposed methodology, let us start with discussing the formal definition of misinformation and its types.

2.1 What Is Misinformation?

Misinformation is a false statement or set of statements which mislead other people by hiding or ”twisting” the actual facts (Islam et al., 2020). This concept causes feelings of mistrust, which is a growing problem in society due to increased popularity of the Internet – social media, news web sites, or gossip portals. It is usually a difficult task to distinguish between false and true statements, in which the mislead is very subtle. Mathematically, misinformation may be defined as a binary function, which assigns *true* or *false* to an arbitrary sentence. Nonetheless, the truth may be somewhere in between.

$$M(a) = \begin{cases} 1, & \text{if } a \text{ is true} \\ 0, & \text{if } a \text{ is false} \end{cases}$$

There are usually five different types of information specified:

- false information – broad concept of misinformation. It is intentionally used to be defined as a correct information interchangeably
- rumor – a story of doubtful truth, which is usually spread widely
- **fake news** – modified version of an original news or piece of information that is spread intentionally and usually very difficult to identify
- spam – an unwanted message with irrelevant, inappropriate, or even harmful information used to mislead users
- disinformation – false facts that are conceived to deceive a user

Type	Characteristics	Objectiveness	Severity	Integrity
Rumors	Ambiguous	Not sure	Low	Not sure
False information	Deception	Yes	High	False
Fake news	Misguided	Yes	Medium	False
Spam	Confused	Yes	Low	Not sure
Disinformation	Mislead/deceive	Yes	Medium	False

Figure 1: Different misinformation types considered in (Islam et al., 2020)

2.2 Impact of Misinformation

The phenomenon of all types of misinformation may influence various spheres of real world, for example society, economy, politics, emergency response during natural disasters, epidemic, and crisis. Public opinion is misled to confuse society and influence decisions and choices. Nowadays, due to the popularity of social network platforms like Facebook, Twitter, or Reddit, it has become easier to spread misinformation in seconds.

2.3 Our Goals

Our goal is to find potential patterns in fake news and analyse their structure so that they may be detected more accurately. We want to implement and test different approaches which originate from the existing research to propose new preprocessing pipelines, include other input data, and test original improvements to known architectures.

3 Literature review

There were many approaches to tackle the problem of fake news detection. In (Islam et al., 2020) three different categories of models were described: discriminative, generative and hybrid models.

3.1 Discriminative models

There are three main discriminative models generally used for text classification: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Recursive Neural Networks (RvNN). CNN has been successfully used in various text analyses (Jacovi et al., 2018). For the similar task of stance detection and rumor verification, word matrices consisting of word vectors were used to feed convolutional networks (Chen et al., 2017b). One research extended the usage of CNN to include not only text but also visual information such as images. This was accomplished by running two parallel CNNs for latent feature extraction, as fake news often also includes a wide variety of deceptive images (Yang et al., 2018).

RNNs are also used due to the sequential characteristics of text data. One of the approaches utilizes a combination of word2vec model architecture (Mikolov et al., 2013) with LSTM model to both encode and process rumor sentences for classification task (Alkhodair et al., 2020). Further extension use attention-based mechanisms that search for relevant parts of source sentence for predicting target label (Chen et al., 2017a).

The last considered model architecture is RvNN, which is used to capture the hierarchical structure of the input. One of the research describes the successful usage of the model for rumor detection on the Twitter dataset, which takes into consideration replies from other users' reactions, such as approving or disagreeing with the original post(Ma et al., 2018) (Zubiaga et al., 2016).

3.2 Generative models

The nature of generative models is well suited for a topic such as misinformation, as authors tend to intentionally sound as factual as possible to deceive the reader, the former playing the role of a generator and the latter indirectly acting as a discriminator. That said, there are many types of generative models that are applicable for this type of problem, most notably Restricted Boltzmann Machine, Deep Belief Network, Deep Boltzmann Machine, Generative Adverserial Network and Variational Autoencoder.

Starting with Deep Belief Networks which is a generative graphical model composed of multiple layers of latent variables, these types of networks can be interpreted as a composition of simple, unsupervised networks where each subnetwork's hidden layer serves as the visible layer for the next. It has been proved by Wei et al. (Wei et al., 2018) that this model achieves a better result than the traditional SVM-based approach.

Moving on to Generative Adverserial Networks, we observe that widespread rumors and misinformation usually result from the deliberate dissemination of information which is generally aimed at forming a consensus on rumor news events, Ma et al. (Ma et al., 2019) proposed a generative adversarial network model to make automated rumor detection more robust and efficient and is designed to identify powerful features related to uncertain or conflicting voice production and rumors.

Finally, we consider Variational Autoencoder models which make strong assumptions concerning the distribution of latent variables, they introduce a specific estimator for the training algorithm known as the stochastic gradient variational bayes (SGVB) estimator. Qian et al. (Qian et al., 2018) proposed this model to extract new patterns by analyzing a user's past meaningful responses on true

and false news articles, this played a vital role in detecting misinformation on social media. Additionally Wu et al. (Wu et al., 2017) explored whether the knowledge from the historical data analysis can benefit rumor detection. The result of their study was that similar rumors always produce the same behaviors.

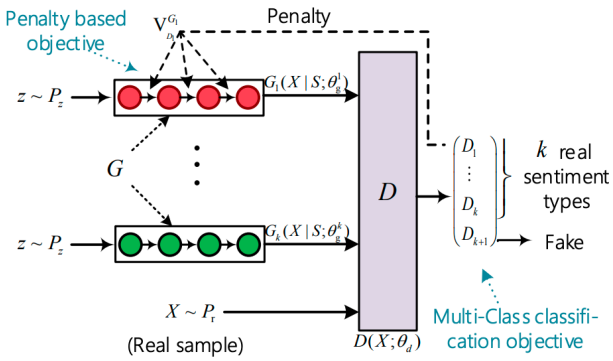

Figure 2: SentiGan architecture

We chose to implement the (Ke Wang, 2018) architecture proposed by Ke Wang and Xiaojun Wan as the model is focused more on the multi class classification objective, this will be useful especially for our LIAR dataset which comprises of smaller observations in terms of length, as generative models are computationally much more demanding when it comes to input size. The SentiGAN architecture is designed for generating text with specific sentiment types. It consists of k generators, each responsible for generating text with a specific sentiment label, and a discriminator. The generators use noise from a distribution Pz, such as a normal distribution, as input and are trained to minimize a penalty-based objective in order to produce text that can deceive the discriminator. On the other hand, the discriminator is trained to classify the text as either fake (generated by the generators) or real with one of the k sentiment types. In the experiments, SentiGAN is set up to generate text with two sentiment types (positive and negative), in our case we will have as many sentiment types as we have in the LIAR dataset.

Additionally we chose to experiment with the catGAN (Categorical Generative Adversarial Networks) which is a variant of GANs that are is in natural language processing tasks. They are used to generate text samples from a given categorical distribution, such as a specific language or topic (unlike SentiGAN which focuses more on sentiments). The generator network in a CatGAN generates text samples, while the discriminator network tries to distinguish the generated samples from real samples. The two networks are trained together in an adversarial manner, with the generator trying to produce samples that are as realistic as possible, while the discriminator tries to accurately identify the generated samples. CatGANs have been used in NLP tasks such as text generation, language transfer, and text-to-speech synthesis.

BLEU (Bilingual Evaluation Understudy) is a commonly used evaluation metric for generative models in natural language processing (NLP). It measures the similarity between a generated sentence and a reference sentence by comparing the n-grams (sets of consecutive words) of the two sentences, in our case we see some improvements each epoch, however we were not able to reach high scores due to the typical computational complexity of training GAN models.

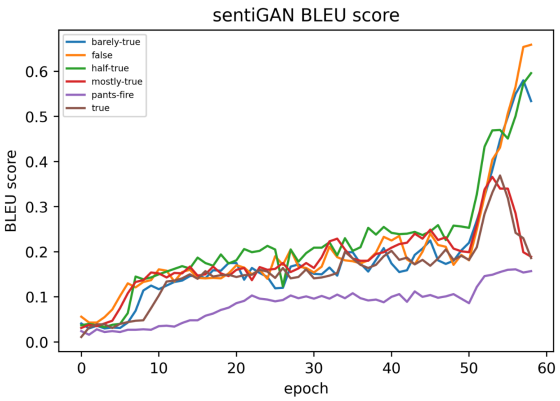

Figure 3: SentiGan BLEU scores per class on LIAR dataset

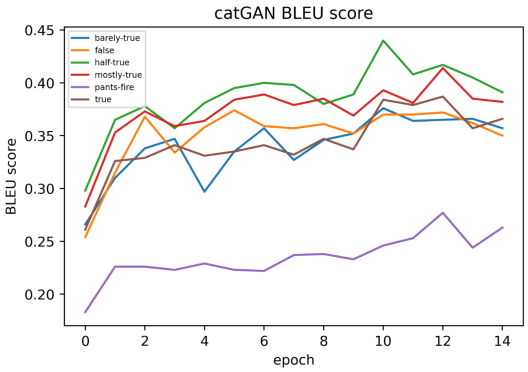

Figure 4: CatGAN BLEU scores per class on LIAR dataset

Overall, the use of both SentiGAN and CatGAN models in multiclassification NLP tasks is challenging for several reasons:

- Multi-class classification: In a multiclassification task, the distribution of classes in the training data may be imbalanced, with some classes having more examples than others. As the generator is very sensitive to the input size this can make it difficult for the SentiGAN and CatGAN discriminators to learn the features of the underrepresented classes, leading to poor performance on those classes (it is the main reason the models had bad results in the prepossessed dataset due to smaller input size).
- Multi-modal Generation: Sentiment analysis and text generation are inherently multimodal, i.e. a single sentence can express different sentiments or can belong to multiple classes. This makes the task of SentiGAN and CatGAN models more difficult.
- Evaluation Metrics: Evaluating the performance of SentiGAN and CatGAN models on multiclassification NLP tasks can be challenging due to the lack of appropriate evaluation metrics. Traditional metrics such as accuracy or F1-score may not be adequate.

All these challenges make it difficult to train and evaluate SentiGAN and CatGAN models on multiclassification NLP tasks, and further research is needed to develop models and evaluation methods that can effectively handle these challenges.

Label	barely-true	false	half-true	mostly-true	pants-fire	true	AVG
Accuracy (%) no preprocessing	79.28	12.55	2.07	0.31	24.32	15.72	22.37
Accuracy (%) with preprocessing	0	11.25	75.26	0	0	0	14.4

Figure 5: SentiGAN accuracy per class on LIAR test data

The GAN models generator outputs are present in the /project2/generator-outputs/unix-time-stamp/docs/model-type/samples/* The trained models are available in this google drive link `https://drive.google.com/drive/folders/15d_0-SY_sV2NML2dvL3ErwIF3ia3ETTH?usp=sharing`

Label	barely-true	false	half-true	mostly-true	pants-fire	true	AVG
Accuracy (no preprocessing)	8.89	21.12	35.96	18.11	0	19.11	17.19
Accuracy (with preprocessing)	0	2.86	96.63	0	0	0	16.58

Figure 6: CatGAN accuracy per class on LIAR test data

3.3 Hybrid Model for Detecting Misinformation

The variety of different models used for misinformation detection is wide. Different Deep Learning architectures were used separately. However, some researchers also decided to test hybrid models to increase the performance of individual models of one type. The example hybrid models consist of *Convolutional Recurrent Neural Networks*, *Convolutional Restricted Boltzman Machine*, *Ensemble-Based Fusion*, and *Long Short-Term Memory Density Mixture Model*.

3.3.1 Convolutional Recurrent Neural Network (CRNN)

Recently, applying CNN and RNN models in a hybrid way was proven to achieve better performance in some applications of Natural Language Processing, Speech Recognition or Time-Series Classification. It is believed that real-world data are structured sequences with spatial-temporal trends (Lin et al., 2019). These models combine a Convolutional Neural Network for visual features extraction with a Recurrent Neural Network for sequential connection between features. Such models have been effectively applied for fake news detection. For example, (Ruchansky et al., 2017) proposed the structure of a deep hybrid model with *Capture* and *Score* modules for fake news detection.

In (Zhang et al., 2018b), researchers proposed deceptive review identification by RCNNs and demonstrated that the neural network approaches outperform the conventional techniques for all datasets.

3.3.2 Ensemble-Based Fusion (EBF)

In (Wang, 2017a), the first fake news detection benchmark dataset with speaker information was made. Wang used a hybrid model with speaker profiles data as the input. (Zhang et al., 2018a) analysed the problem of fake news detection on social media by identifying deceptive words used

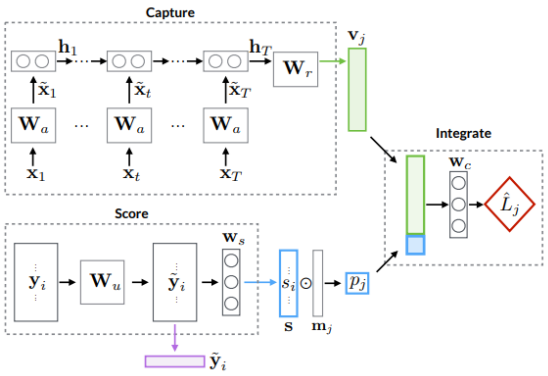

Figure 7: CSI model specification

by online fake users harming society. Also, in (Shu et al., 2018), fake news data repository FakeNewsNet was published for further analysis.

3.3.3 LSTM Density Mixture Model

Using only lexical features to detect fake news automatically was very popular methodology. Nonetheless, the hybrid deep neural network models started becoming more and more popular. Already mentioned (Ruchansky et al., 2017) analysed three different types of fake news:

- the text of an article, which our research will be focused on,
- the user response,
- the source on which users promote it.

They also analyzed how fake news affect public opinion. To do so, Ruchansky proposed a hybrid model combining all the characteristics to predict more accurately. Similarly, (Long et al., 2017) suggested a novel method with speaker profiles fed into attention-based LSTM architecture for fake news detection. LSTM-based models were proven to give better results for long sentences, whereas attention models provide weights representing importance of different words in context.

3.4 Additional Input to Existing Benchmarks

In (Alhindi et al., 2018), researchers introduced the extension of the LIAR dataset by extracting the justification from the article used to label a particular statement. It was shown that modelling the source and justifaction alongside with the initial input provides a significant improvement in models' performance in both binary and multi-class classification.

The human-provided justification – for example, a summary evidence – improves the assessment of a piece of news compared to modeling the statement alone in all types of machine learning models, both feature-based and deep learning model. The authors analysed four different scenarios:

- **S condition** – claim or statement representation alone
- **S+M condition** – claim or statement representation with metadata information
- **SJ condition** – claim or statement and the extracted justification
- **S+MJ condition** – claim or statement, metadata, and justification

3.4.1 Feature-Based Models

First, the researchers tested simple linear models, i.e. Logistic Regression and SVM (Support Vector Machines) with linear kernel. For the represantion of the statement alone, unigrams, tf-idf, and Glove word embeddings (Pennington et al., 2014) were used to extract features. Usually, the first representation gave the best results.

3.4.2 Deep Learning Models

For the base deep learning model, the authors chose to use Bidirectional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997), which proven its successful performance in NLP tasks. For the statements alone, one BiLSTM with Glove embeddings – a 100-dimensional layer followed by 32-dimensional BiLSTM layer. For the **S+MJ** condition, on the other hand, two different approaches were tested. In the first one, the justification was concatenated to the statement and passed to a single BiLSTM, whereas in the second one, a parallel architecture with two BiLSTMs was used. The outputs of these BiLSTMs were then concatenated and passed to a softmax layer. For most classes, the second approach gave better results.

It can be noticed that the performance was sometimes even tripled after adding the second input to the models.

4 Datasets

This time, we have decided to focus only on texts in which the average count of words is below 1000 and there are less complex articles.

Class	class size	S		SJ		
		LR	BiLSTM	LR	BiLSTM	P-BiLSTM
pants-fire	92	0.12	0.11	0.38	0.33	0.39
false	250	0.31	0.31	0.35	0.32	0.35
mostly-false	214	0.25	0.15	0.35	0.27	0.33
half-true	267	0.24	0.26	0.41	0.27	0.34
mostly-true	249	0.23	0.30	0.35	0.35	0.33
true	211	0.25	0.16	0.37	0.36	0.41
total/avg	1283	0.25	0.23	0.37	0.31	0.35

Figure 8: F1-Score per class on a test dataset

4.1 LIAR

In our case, the LIAR dataset (Wang, 2017b) is will be used for the purpose of both training and testing models. It was manually labeled by PolitiFact and consists of short statements published in 2017. Each sentence was assigned one of the following labels: pants-fire (*very false*), false, barely-true, half-true, mostly-true or true. There are in total 12836 cases. The dataset itself is a well-balanced set, having a range between 2000 to 2600 instances per class. Some of the attributes are:

- `label`
- `statement`
- `subject`
- `speaker`
- `speaker's job title`
- `party affiliation`

4.2 CT-FAN

The CT-FAN (Köhler et al., 2022) dataset consists of news articles multilabeled data published in 2022. The dataset was collected from 2010 to 2022, thus includes various topics related to topics like elections, COVID-19 etc. While it also has news articles in German, we will be using only data available in English. The dataset contains in total 1264 articles. The attributes of the dataset are presented as follows:

- `id`
- `title`
- `text`
- `rating`

There are 4 labels in total: False, Partially False, True and Other that relates to any unproven articles or articles in dispute.

4.3 LIAR-PLUS

The extended LIAR-PLUS (Alhindi et al., 2018) dataset is an improved version of the original LIAR dataset for fact-checking and fake news detection. It includes evidence sentences that have been automatically extracted from full-text verdict reports written by journalists in Politifact. The goal of this dataset is to provide a benchmark for evidence retrieval and to show that including evidence information in any automatic fake news detection method results in superior performance compared to methods that lack such information. The dataset includes 15 columns, with the added column being the extracted justification. The justification extraction method involves collecting all sentences in the 'Our Ruling' section of the report which summarises the claims, or the last five sentences if it does not exist, and removing any sentences that contain the verdict and any verdict-related words.

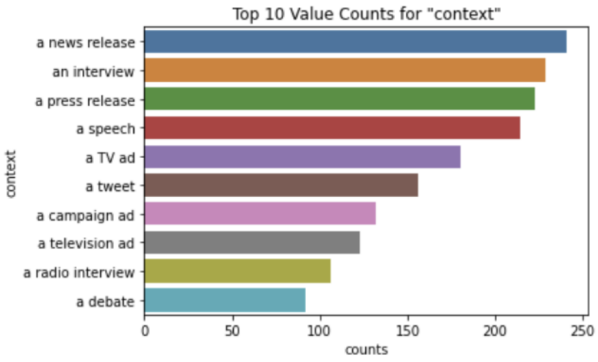

Figure 9: The most popular contexts in LIAR-PLUS dataset

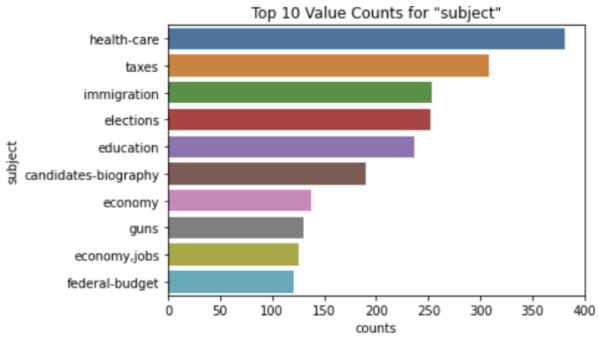

Figure 10: The most popular subjects in LIAR-PLUS dataset

Let us present some random example justification provided for the statement about Oregon state.

- **Statement:** Oregon is the only state out of

the 50 states in the USA that continues to pay 100% of the medical benefits for its employees and their families.

- **Label:** false
- **Justification:** Oregon is one of two states that covers a full range of benefits – health, dental and vision – but it's one of at least four that covers the premiums for its lowest-cost health plan. Richardson's larger point, that Oregon is in a shrinking group of states that do so is certainly a strong and valid one. However, he undercuts his argument by resorting to hyperbole.

5 Second Project Proposal

We plan to continue working on fake news classification task with the following extensions.

- multi-class classification with the degree of certainty that a piece of news is fake (for example, 0-5 scale)
- datasets with extended input, for example authors' data, context, justification
- providing analysis how the choice of inputs influence the models' performance
- different methods used in state-of-the-art approaches that can be improved
- application of generative models: SentiGAN and CatGAN

6 Methodology

In our case, we define a multi-class classification problem in which we will predict to what extent a text can be classified as fake news. We will compare performance of a few different methods on this specific task.

6.1 Project Contributions

After further research of existing papers and taking into account your reviews, we have planned to work on a few unanswered questions in fake news detection tasks.

- How different preprocessing techniques influence the results of different architectures?
- Does keeping some stop words like yes or no influence the model's performance?
- Does a particular model trained on one political fake news dataset generalise to other datasets of a similar topic? Do models usually overfit, or are able to recognise patterns in fake news data?

6.2 Preprocessing

In order to prepare the data for our model we need to apply some preprocessing techniques such as:

- Using regular expressions to remove: punctuations, special symbols ($, #, &), digits, URLs, wide spaces, single characters
- Removing stop words
- Lower casing
- Lemmatization using two different libraries (SpaCy, NLTK)
- Tokenization suited for a particular architecture
- Article padding

We might have to also consider additional preprocessing steps depending on which type of model we will decide to use.

6.3 Proposed Architectures

First method will involve combination of word embedding using word2vec and LSTM for classification task. Further customizations include different loss functions like cross-entropy loss or log-likelihood and other hyperparameters.

The next architectures that we want to apply for the given problem are hybrid models, which were proven to result in accurate classification. We will start with simple CNN+RNN models that extract both shape-related and sequential features of the articles' texts. At this point, ensemble models with lighter architectures may also be introduced to provide variety of approaches for the given task.

Lastly, we are planning to test state-of-the-art attention-based architectures with pre-trained text representation model like BERT to extract text features, (Matsumoto et al., 2021).

7 Results

7.1 Word2vec with LSTM

We have tested a mix of word2vec model for word embeddings with LSTM neural network. We applied all preprocessing techniques mentioned in

section 6.2. The optimizer used for training is Adam. The model consists of:

1. embedding layer obtained with word2vec
2. 32 LSTM units
3. dropout layers

The tests were conducted for both LIAR-PLUS and CT-FAN datasets consisting of 6 and 4 classes, respectively. We remind that results with S character denote including statements into the training of the model, and S+J denotes including both statements and justifications for the model training. The balanced accuracy is our main indicator of results since datasets were significantly imbalanced.

Dataset	Accuracy	Balanced accuracy
LIAR-PLUS (6 classes): S	0.19079	0.17397
LIAR-PLUS (6 classes): S+J	0.19205	0.18438
CT-FAN (4 classes)	0.45810	0.31610

Table 1: Results for word2vec with LSTM model

As seen in the table, while the model has managed to learn and get high accuracy on the training dataset, it was not reflected on the test dataset, obtaining accuracy similar to that of a random classifier.

7.2 BERT

Similar to the first project, we used pretrained *BertForSequenceClassification* with *BertTokenizer* implemented in Transformers library in Python. The architectures consists of 124 layers, 1024 hidden layers, and 16 heads. The weight of the model is about 450 MB. This model was pretrained on uncased vocabulary. We trained the model on GPU – Nvidia RTX 3060 with 12 GB of VRAM. The memory allowed to use batch size of 4 observations.

Maximum lengths of tokens is 512, which is popular for pretrained transformers. It caused some complications for longer statements, articles, and justifications used for model training. Some observations were padded to fit the model. Other interesting approach would be to use multiple models for different inputs.

For reproducability, the seed of both the dataloader and the model was fixed before the training.

For both of the datasets, training-validation-test split proposed by the researchers was used. We also used AdamW optimiser with 10^{-4} learning rate and StepLR scheduler with the step size of 3 and factor 0.5 or 0.1. We trained each model for between 10 to 20 epochs.

As a point of reference, we also trained our BERT model on LIAR-PLUS dataset with using all preprocessing techniques. After 20 epochs, we obtained the following results.

- **Accuracy:** $0.61510 \rightarrow 0.62344$
- **F1-Score:** $0.62789 \rightarrow 0.61682$

It occured that there was no significant improvement in the results. It may be caused by the fact that the added justification was too long for the BERT model.

Because the classes present in the LIAR-PLUS dataset represent the degree of uncertainty, we decided to test two more metrics – MAE and MSE.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i|$$

$$\text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2}$$

The results of the multiclass classification on LIAR-PLUS dataset for the random 80-20 split for the training, which were tested on the proposed by the researchers test data, are as follows.

Scenario	Balanced Accuracy	Weighted F1-Score	MAE	MSE
80-20 training on Valid: S+J	0.78586	0.79560	0.43828	1.29297
80-20 training on Test: S+J	0.84140	0.83385	0.36709	1.11076

Figure 11: The model trained on 80-20 split tested on the test dataset proposed in (Alhindi et al., 2018)

The results were quite promising, but some of the observations may have been used for training. The ability for the architecture to generalise on unseen data was much more difficult task. Table 12 presents the results for train, validation, and test dataset in two scenarios – S (only statements) and $S + J$ (statements and justification.

The results on both the validation and test datasets are significantly worse than the one on training datasets. The BERT model overfits after a few epochs and the validation accuracy stops

Scenario	Balanced Accuracy	Weighted F1-Score	MAE	MSE
Train: S	0.93465	0.93677	0.11748	0.28736
Train: S+J	0.91823	0.92099	0.14016	0.34311
Validation: S	0.23441	0.24383	1.68047	4.84922
Validation: S+J	**0.24940**	**0.25722**	1.61875	4.72031
Test: S	**0.23825**	**0.23796**	1.71851	5.05680
Test: S+J	0.20855	0.21413	1.79763	5.44256

Figure 12: The results for the training, validation, and test dataset on LIAR-PLUS for S and $S + J$ scenarios

to improve. Finally, the results for the CT-FAN dataset are presented in Table 13.

Scenario	Balanced Accuracy	Weighted F1-Score	MAE	MSE
Train Dataset	0.83937	0.88907	0.15348	0.26899
Test Dataset	**0.40765**	**0.53178**	0.83333	1.64103

Figure 13: The results for the training and test dataset on CT-FAN

The obtained results are better than for the LIAR-PLUS dataset. However, it should be noticed that there are fewer classes in CT-FAN dataset. Table 14 represents the metrics obtained by teams taking part in the competition organised by the researchers (Nakov et al., 2022).

	Team	True	False	Partially False	Other	Accuracy	Macro-F1
1	iCompass [37]	0.383	0.721	0.173	0.080	0.547	0.339
2	NLP&IR@UNED [38]	0.446	0.729	0.097	0.057	0.541	0.332
3	Awakened [41]	0.328	0.744	0.185	0.035	0.531	0.323
4	UNED	0.346	0.725	0.191	0.000	0.544	0.315

Figure 14: The results for the training and test dataset on CT-FAN

Unfortunately, the metrics cannot be compared since teams evaluated models using accuracy instead of balanced accuracy and macro F1-score instead of weighted F1-score.

8 Potential Improvements and Further Research

- Making the evaluation and testing consistent for all of the architectures used
- Applying ensemble learning with models of different complexity and word embeddings
- Using multiple architectures for each of the inputs (for example metadata, statement, justification in LIAR-PLUS)
- Training GANs on other datasets, with more epochs
- Training on more powerful hardware with more epochs and training optimisation (early stopping, different schedulers, and hyperparameters)

Appendix

Figure 15: Word cloud for *statement* variable in the LIAR-PLUS dataset

Figure 16: Word cloud for *false* statements in the LIAR-PLUS dataset

Figure 17: Word cloud for CT-FAN Dataset

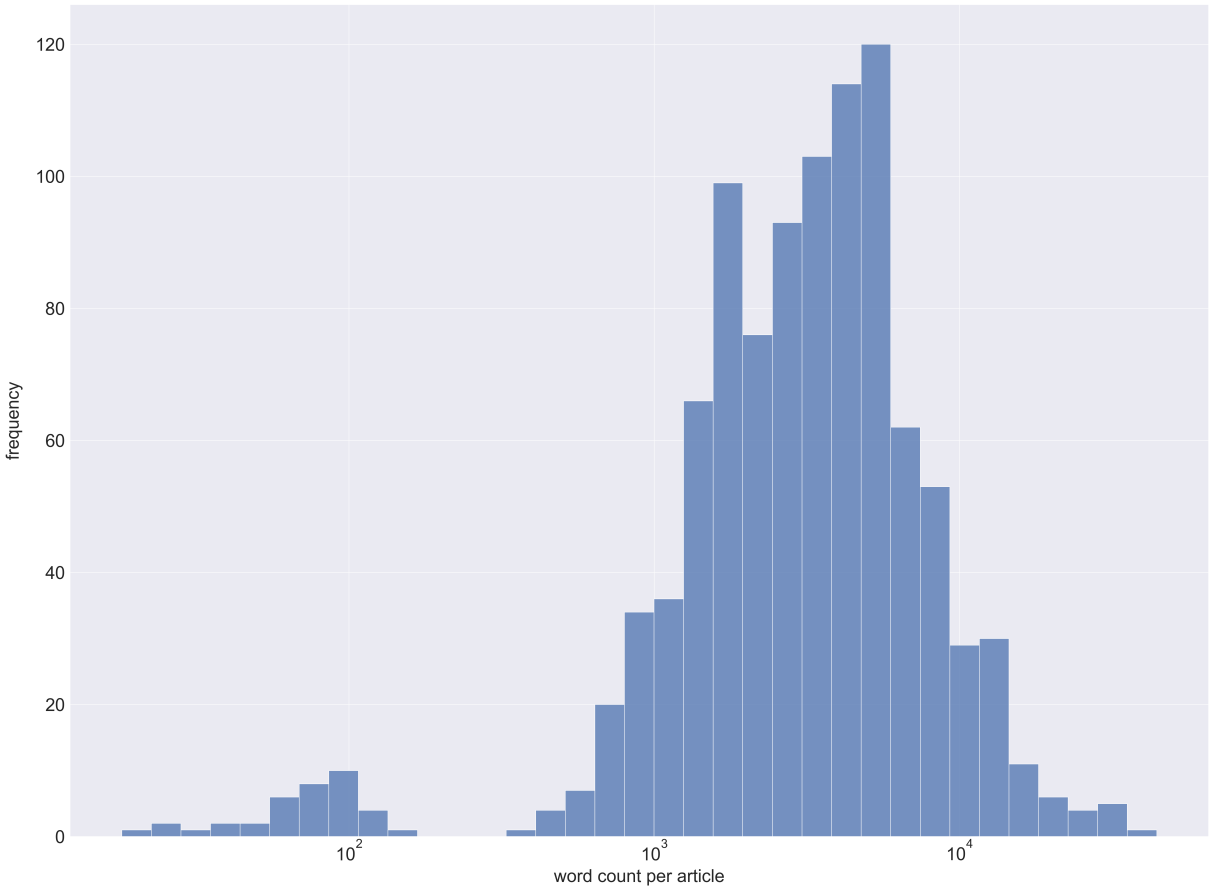

Figure 18: Word count of articles histogram for CT-FAN Dataset

References

[Alhindi et al.2018] Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium, November. Association for Computational Linguistics.

[Alkhodair et al.2020] Sarah A. Alkhodair, Steven H.H. Ding, Benjamin C.M. Fung, and Junqiang Liu. 2020. Detecting breaking news rumors of emerging topics in social media. *Information Processing Management*, 57(2):102018.

[Chen et al.2017a] Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. 2017a. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection.

[Chen et al.2017b] Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017b. IKM at SemEval-2017 task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 465–469, Vancouver, Canada, August. Association for Computational Linguistics.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11.

[Islam et al.2020] Md. Rafiqul Islam, S. Liu, Xianzhi Wang, and Guandong Xu. 2020. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, 10.

[Jacovi et al.2018] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification.

[Ke Wang2018] Xiaojun Wan Ke Wang. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. July.

[Köhler et al.2022] Juliane Köhler, Gautam Kishore Shahi, Julia Maria Struß, Michael Wiegand, Melanie Siegel, and Thomas Mandl. 2022. Overview of the CLEF-2022 CheckThat! lab task 3 on fake news detection. In *Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum*, CLEF '2022, Bologna, Italy.

[Lin et al.2019] Xiang Lin, Xiangwen Liao, Tong Xu, Wenjing Pian, and Kam-Fai Wong. 2019. Rumor detection with hierarchical recurrent convolutional neural network. In *NLPCC*.

[Long et al.2017] Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *IJCNLP*.

[Ma et al.2018] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on Twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1980–1989, Melbourne, Australia, July. Association for Computational Linguistics.

[Ma et al.2019] Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The World Wide Web Conference*, WWW '19, page 3049–3055, New York, NY, USA. Association for Computing Machinery.

[Matsumoto et al.2021] Hayato Matsumoto, Soh Yoshida, and Mitsuji Muneyasu. 2021. Propagation-based fake news detection using graph neural networks with transformer. pages 19–20, 10.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

[Nakov et al.2022] Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Martino, Firoj Alam, Julia Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Gautam Kishore Shahi, Hamdy Mubarak, Alex Nikolov, Nikolay Babulkov, Yavuz Kartal, and Javier Beltrán, 2022. *The CLEF-2022 CheckThat! Lab on Fighting the COVID-19 Infodemic and Fake News Detection*, pages 416–428. 01.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

[Qian et al.2018] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. 2018. Neural user response generator: Fake news detection with collective user intelligence. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3834–3840. AAAI Press.

[Ruchansky et al.2017] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news. *CoRR*, abs/1703.06959.

[Shu et al.2018] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286.

[Wang2017a] William Yang Wang. 2017a. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648.

[Wang2017b] William Yang Wang. 2017b. "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 (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, July. Association for Computational Linguistics.

[Wei et al.2018] Lei Wei, Donghuai Gao, and Cheng Luo. 2018. False data injection attacks detection with deep belief networks in smart grid. In *2018 Chinese Automation Congress (CAC)*, pages 2621–2625.

[Wu et al.2017] Liang Wu, Jundong Li, Xia Hu, and Huan Liu, 2017. *Gleaning Wisdom from the Past: Early Detection of Emerging Rumors in Social Media*, pages 99–107. 06.

[Yang et al.2018] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. 2018. Ti-cnn: Convolutional neural networks for fake news detection.

[Zhang et al.2018a] Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B. Gouza. 2018a. Fake news detection with deep diffusive network model. *CoRR*, abs/1805.08751.

[Zhang et al.2018b] Wen Zhang, Yuhang Du, Taketoshi Yoshida, and Qing Wang. 2018b. Dri-rcnn: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing Management*, 54(4):576–592.

[Zubiaga et al.2016] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2438–2448, Osaka, Japan, December. The COLING 2016 Organizing Committee.