Performance Analysis of Different Word Embeddings and Transformers on Fake News Detection

Patupat, Albert John Lalim
20544416
ajlpatupat@connect.ust.hk

Cheng, I-Tsun
20576079
ichengaa@connect.ust.hk

2020 May 22

1 Introduction

In the modern digital age, the rise in popularity of web and social media has made the spread of information faster and easier than ever before. However, as more information becomes increasingly accessible, the risk of harmful and dangerous disinformation increases concurrently. In particular, fake news contains deliberate disinformation that is often published with malicious intent, such as harming a targeted group or attracting attention solely for earning advertising revenue.

Due to the proliferation of fake news across the Internet, it is essential to design accurate and robust methods to detect fake news. In natural language processing, recent advances proposed numerous neural-network-based models for analyzing text data, including models with word embedding layers and models based on transformers. Hence, in this study, we explored the performance of different word embeddings and transformer models for fake news detection.

2 Machine Learning Task

This study focused on detecting fake news, which can be interpreted as Binary Text Classification. Specifically, a news article is represented as a tuple (x_{i1}, x_{i2}, y_i) of its headline or title x_{i1}, its content or text x_{i2}, and its label $y_i \in \{0, 1\}$. The machine learning task is to predict y_i given x_{i1} and x_{i2}.

3 Datasets and Preprocessing

3.1 Datasets

In this study, we used three datasets: Real and Fake News Kaggle Dataset, ISOT Fake News Dataset, and LIAR Dataset.

The Real and Fake News Kaggle Dataset consists of 6335 (3174 real and 3171 fake) news articles compiled from various news sources. The majority of these articles are related to politics, specifically, the 2016 US presidential elections. Each news example consists of the numerical ID, the title, the text, and the label indicating whether the news is "REAL" or "FAKE".

The ISOT Fake News Dataset, created by the University of Victoria, contains real and fake news articles collected from online real-world sources. The real news examples were crawled from Reuters, while the fake ones were obtained from unreliable websites that are flagged by PolitiFact and Wikipedia. The dataset contains 21417 real news on politics and world news, and 23481 fake news roughly categorized into the following topics: Government, Middle-East,

US, Left-wing, Politics, and General. The examples are separated into two files according to the label, and each example consists of the title, the text, the subject, and the publication date.

The LIAR Dataset from University of California Santa Barbara, contains 12791 short statements from a wide variety of sources: political debates, TV ads, Facebook posts, tweets, interviews, news releases, etc. Each example consists of the ID of the statement, the label, the statement, and contextual information, such as the subject, the speaker, and the speaker's job title. Unlike in the previous datasets, the labels in the LIAR Dataset are on a discrete scale: "pants-fire", "false", "barely-true", "half-true", "mostly-true", and "true". The distribution of the labels are 1047, 2507, 2103, 2627, 2454, and 2053, respectively.

3.2 Preprocessing

Preprocessing started with column preparation wherein only necessary data is extracted or constructed from the initial dataset. Specifically, for the Real and Fake News Kaggle Dataset, the label, the title, and the text are extracted, while an extra column named "titletext" was constructed by concatenating the title and the text. For the ISOT Fake News Dataset, the text was extracted, and the label was constructed according to which file the example originated. For the LIAR Dataset, the text was extracted from the statement, and the label was constructed according to the following rule: "pants-fire" and "false" statements are considered fake, "mostly-true" and "true" are considered real, while "barely-true" and "half-true" statements were dropped.

After column preparation, preprocessing proceeded with data filtering. For all three datasets, examples with empty or null data were removed. Furthermore, in all textual data, all whitespaces were replaced with the space character, and the strings were trimmed to only the first 200 words in order to reduce data size. Due to the exclusive presence of Reuters datelines in the real news of the ISOT Fake News Dataset, these datelines were trimmed before extracting the first 200 words. Indeed, as shown in the proceeding sections, the first 200 words of a news article appears to be sufficient in verifying authenticity.

Finally, after data filtering, the datasets were split into training, validation, and testing sets according to the desired train-valid-test ratio, while maintaining the balanced nature of the datasets, i.e. stratified splitting.

4 Machine Learning Methods

4.1 Models

Our investigation used two baseline models (BiLSTM with Word2Vec; BiLSTM with Word2Vec_2), studied two intermediate models (BiLSTM with GloVe; BiLSTM with ELMo), and analyzed three advanced models (BERT; ALBERT; DistilBERT).

The BiLSTM was selected to be the common encoder of the different word embedding layers due to its advantage in dealing with long-term dependencies in long sequence data and because bidirectional language representations generally perform better than unidirectional ones.

For our baseline models, we adopted Word2Vec as the embedding method. Word2Vec is an algorithm that converts words into distributed representations, which are fixed-length vectors learned in training. The Word2Vec layers in our baseline models are randomly initialized and not pretrained. For specification, let Word2Vec denote the layer initialized with 300 dimensions and Word2Vec_2 denote the layer initialized with 256 dimensions.

For our intermediate models, we utilized GloVe and ELMo. GloVe (Global Vectors) are pretrained word representations trained on the global word-word co-occurrence matrices from different corpora. The version used in this study is pretrained on the Wikipedia 2014 + Gigaword 5 corpora consisting of 6B tokens, and outputs word vectors in 300 dimensions. On the other hand, ELMo (Embeddings from Language Models) is a deep contextualized word representation that models complex characteristics of words. Notably, it is character-based, making

it more robust in learning out-of-vocabulary tokens. The version used in this study is pretrained on the 1 Billion Word Language Model Benchmark consisting of approximately 800M tokens of news crawl data from WMT 2011. Additionally, the ELMo layer of the selected size outputs word vectors in 256 dimensions.

For our advanced models, we utilized the recent state-of-the-art transformer architecture. BERT (Bidirectional Encoder Representations from Transformers) is deeply bidirectional and pretrained on the BooksCorpus (800M) + English Wikipedia (2500M words) corpora. We further picked two BERT-based models to compare their performance: ALBERT and DistilBERT. ALBERT presents parameter-reduction techniques to lower memory consumption and improve training speed of BERT, while DistilBERT's main advantage is being 40% smaller yet 60% faster than BERT while retaining 97% of BERT's language capabilities. For fair comparison, the base versions with the least parameters from each model were selected and implemented.

4.2 Hardware and Software

All experiments in this study were conducted on Google Colaboratory, which provides a Jupyter notebook environment. More importantly, it executes code on Google's cloud servers, allowing access to their GPUs.

Regarding machine learning libraries utilized in this study, TorchText 0.3 was used for dataloading, while PyTorch 1.5 was primarily used for model construction and model training. Furthermore, TorchText 0.3 directly supports Word2Vec and GloVe, AllenNLP 0.9 was used to implement ELMo, while SacreMoses 0.0 was used for Moses tokenization before application of these word embedding layers. Additionally, Huggingface 2.9, best known for its open-source development in transformers, was used to implement BERT, ALBERT, and DistilBERT.

4.3 Hyperparameter Settings

Tables 1 and 2 summarize the hyperparameter settings of the models with word embedding layers and transformer-based models used in this study, respectively.

Table 1: Hyperparameter settings of models with word embedding layers.

Embedding Layer	Word2Vec	Word2Vec_2	GloVe	ELMo
Tokenizer	Moses Tokenizer			
Input Restrictions	-			$\leq$ 50 char's
Embedding Dimensions	300	256	300	256
Encoder	1-layer BiLSTM with 128 dimensions			
Batch Size	32			
Learning Rate	1×10^{-3}			3×10^{-3}
Training Details	5 epochs, Adam optimizer, cross entropy loss			
Validation Details	Holdout validation, evaluated twice per epoch			

Table 2: Hyperparameter settings of transformer-based models.

Transformer	BERT	ALBERT	DistilBERT
Tokenizer	BERT Tokenizer	ALBERT Tokenizer	DistilBERT Tokenizer
Input Restrictions	$\leq$ 128 tokens		
Dimensions	768		
Batch Size	16		
Learning Rate	2×10^{-5}		
Training Details	5 epochs, Adam optimizer, cross entropy loss		
Validation Details	Holdout validation, evaluated twice per epoch		

5 Experiments and Results

The experiment stage of our study consists of three phases focusing on different analyses of fake news detection: Phase 1 emphasized on comparing the performances of different embedding layers and transformers in a general setting, Phase 2 attempted to observe the capabilities of representative models under low-resource settings, and Phase 3 explored the possibility of transfer learning through pretraining of these representative models. Due to the balanced nature of the datasets and stratified splitting used in this study, the metric for performance was set to be the test accuracy. More details regarding each phase are explained in the following sections.

5.1 Phase 1

Phase 1 focused on comparing the performances of different embedding layers and transformers in a general setting, and on identifying important information for detecting fake news. In this phase, a grid search between "models" and "versions" was conducted. Specifically for the "models", we studied four different embedding methods (Word2Vec; Word2Vec_2; GloVe; ELMo) each implemented with a BiLSTM, and three different transformers (BERT; ALBERT; DistilBERT). For the "versions", four different variations of a given model were studied: "Title" uses only the title, "Text" uses only the text, "Title-Text" uses the concatenation of the title and the text in this specific order, and "Title+Text" is an ensemble of "Title" and "Text" with their outputs concatenated. Furthermore, after encoding, a single fully-connected layer was used as the common classifier for all models.

The dataset used in this phase is the Real and Fake News Kaggle Dataset. Furthermore, the train-valid-test split ratio is 72-18-10, and indeed, we found that this ratio provides a good amount of training data while also setting sufficient test data to produce accurate results. Details of the preprocessing are given in Section 3.

Table 3: Test accuracy (%) of models in Phase 1.

	Title	Text	Title-Text	Title+Text
Word2Vec	83.84	92.87	90.49	92.55
Word2Vec_2	82.88	92.71	90.02	91.92
GloVe	82.73	93.66	92.87	94.77
ELMo	79.56	93.98	90.81	93.34
BERT	87.48	95.88	97.46	96.83
ALBERT	81.30	96.99	93.50	94.77
DistilBERT	86.37	95.72	95.40	96.51

5.2 Phase 2

Phase 2 focused on testing the robustness of representative models under low-resource settings. In particular, the Real and Fake News Kaggle Dataset was used and preprocessed similarly in Phase 1; however, the train-valid-test split ratio was adjusted to simulate different amounts of available data. Specifically, the selected models were trained, validated, and evaluated under the setting "Ratio X" where $X \in 10, 20, 30, 40, 50, 60, 70, 80, 90$ represents the percentage of the entire dataset used for training and validation, which mathematically implies that the train-valid-test split ratio is set to $(0.8X)$-$(0.2X)$-$(100 - X)$.

Word2Vec and GloVe were selected for Phase 2 because in Phase 1, they were the better-performing baseline and intermediate models, respectively. On the other hand, DistilBERT was chosen to represent the advanced transformer models due to its compact size, fast training speed, and relatively competitive performance which is only slightly below BERT's. In this

phase, only the Text version was used since it can be concluded from Phase 1 that the text is the most relevant information for fake news detection, which will be further explained in Section 6.

Table 4: Test accuracy (%) of models in Phase 2.

	Word2Vec	GloVe	DistilBERT
Ratio 90	92.23	94.29	96.83
Ratio 80	92.07	94.29	95.32
Ratio 70	90.90	92.97	95.45
Ratio 60	90.44	92.26	95.32
Ratio 50	89.87	91.24	95.05
Ratio 40	88.52	91.08	94.52
Ratio 30	88.62	90.07	92.99
Ratio 20	86.31	88.08	92.84
Ratio 10	83.32	86.68	91.04

5.3 Phase 3

Phase 3 focused on the generalization ability of a pretrained model on other similar datasets. In order to measure generalization and the effects of pretraining, two other datasets were brought in for this phase: the ISOT Fake News Dataset and the LIAR Dataset. The datasets were preprocessed as stated in Section 3.

In order to observe the effects of pretraining, we evaluated three variations of each model in Phase 2: pretraining only, training only, and both pretraining and training. Specifically, pretraining was conducted on the Real and Fake News Kaggle Dataset under Ratio 100, while training and fine-tuning used the external datasets ISOT Fake News and LIAR.

For the ISOT Fake News Dataset, the train-valid-test ratio was set to 2-0-98, and notably, the validation set was ignored due to minimal training data. This ratio was used to simulate low-resource settings given the large dataset and to compensate for the quality of data which allowed easier fake news detection.

On the other hand, for the LIAR Dataset, the train-valid-test ratio was set to 72-18-10. While minimal training data is desirable to test the effects of pretraining, it should be noted that LIAR is a different kind of dataset containing statements instead of news articles, which would require significantly more training data due to its shorter text length. Hence, to address this difference, the split settings in Phase 1 were used.

Table 5: Test accuracy (%) of models in Phase 3.

		Word2Vec	GloVe	DistilBERT
ISOT Fake News Dataset	Pretraining	72.04	71.19	77.71
	Training	94.75	96.04	95.26
	Both	95.89	96.68	95.93
LIAR Dataset	Pretraining	55.27	57.62	58.86
	Training	63.07	65.92	66.42
	Both	64.06	67.16	67.41

6 Discussion and Analysis

The following discussion and analysis of experiment results are similarly separated according to the three previously defined phases.

6.1 Phase 1

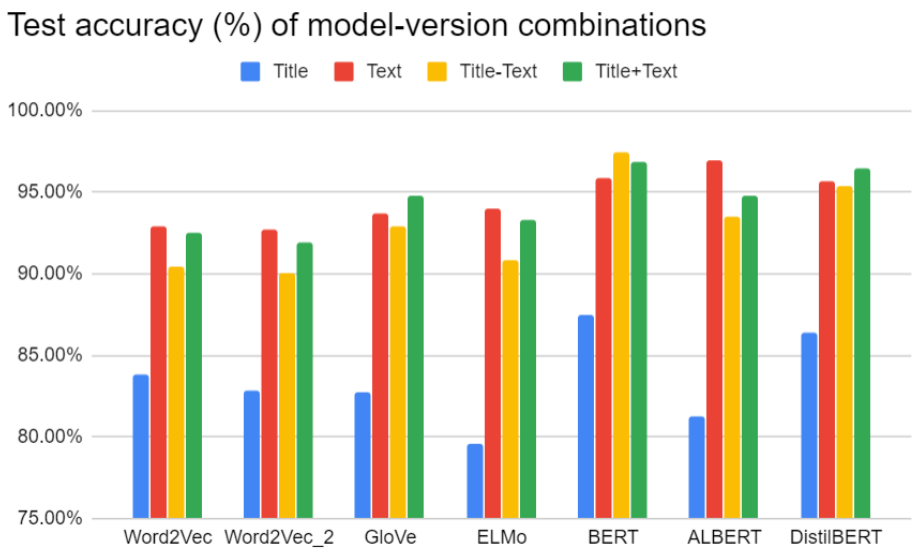

Figure 1: Test accuracy (%) of models in Phase 1.

6.1.1 Version Analysis

Out of the four studied versions, Title was significantly inferior. The probable reason for this is because the headlines of fake news are made to look similar to real news in order to attract attention. Thus, using the title alone is a relatively difficult method for the models to tell whether the news is real or fake compared with other methods of information, despite Title being able to achieve above 80% test accuracy for almost all models.

Title-Text performed better than Title but fell short against Text and Title+Text. This is most likely because the title takes up space which can be made for more text instead. As we limited our textual data to the first 200 words in our datasets, the title takes up considerable space while not containing very effective information for distinguishing between fake or real news. However, Title-Text can still achieve at least 90% test accuracy for all models.

Lastly, Text and Title-Text performed the best and around a similar level. The probable reason for this is because in the Title+Text ensemble, the Text sub-model dominates the Title sub-model due to the title's relative ineffectiveness. This leads to a similar structure and hence performance of these two versions. Furthermore, Title+Text does not perform similarly to Title-Text because in the former, the title and the text are separated and given to two different encoders so the title does not take up the space of text, compared to the latter.

6.1.2 Word Embedding Analysis

For the word embeddings, GloVe generally performed better because it is pretrained on Gigaword-5 which contains mostly news information. It performed worse than the Word2Vec baselines only in Title most likely because some fake news headlines were made to be similar in structure with previous news articles that GloVe has likely seen.

On the other hand, ELMo performed marginally well, better than the Word2Vec baselines but worse than GloVe. The most likely reason is that in the default settings of ELMo, the embedding weights in its BiLSTMs are frozen and do not update during training, contrary to GloVe's updating parameters. Furthermore, ELMo has a smaller vector dimension of 256 than GloVe's 300, which gives it a disadvantage as seen in the consistently better performance of Word2Vec against Word2Vec_2. Additionally, ELMo performed the worst among all models in Title since as a contextual representation, it requires a lot of context which a headline cannot fully provide. Moreover, headlines, especially those of fake news, are exaggerated to attract readers' attention instead of providing useful context for them to understand the real content.

Indeed, the Word2Vec baselines provided reasonable results: over 90% test accuracy for all versions of information except Title. Having learnable weights in the embedding layer and BiLSTM as the base encoder allowed Word2Vec to perform reasonably well. Word2Vec with 300 dimensions outperformed Word2Vec_2 with 256 dimensions by a slight margin due to the former's higher representational power.

6.1.3 Transformer Analysis

Among all models, BERT had the best general performance, dominating almost all versions of information. Aside from its deeply bidirectional structure, this is most likely because BERT contains more parameters and therefore has stronger feature learning capacity than its derived counterparts ALBERT and DistilBERT. BERT contains 110M parameters, ALBERT contains 11M parameters, while DistilBERT contains 66M parameters. BERT is the original base model that ALBERT and DistilBERT aimed to reduce the size of. However, in doing so, some representational power was lost, and thus, BERT remains at the top in the results.

ALBERT generally performed well but came last out of the three transformer models. Its parameter reduction techniques might have caused ALBERT to lose some of the representation power of BERT as it is the transformer with the least number of parameters. However, it performed the best in Text, which may imply that ALBERT's parameter reduction techniques only work well on strong and meaningful data, such as the text and not the title.

DistilBERT performed better than ALBERT in most cases but slightly worse than BERT. This is not surprising since DistilBERT's primary advantage is being faster and cheaper to train than BERT while also keeping most of BERT's characteristics.

6.2 Phase 2

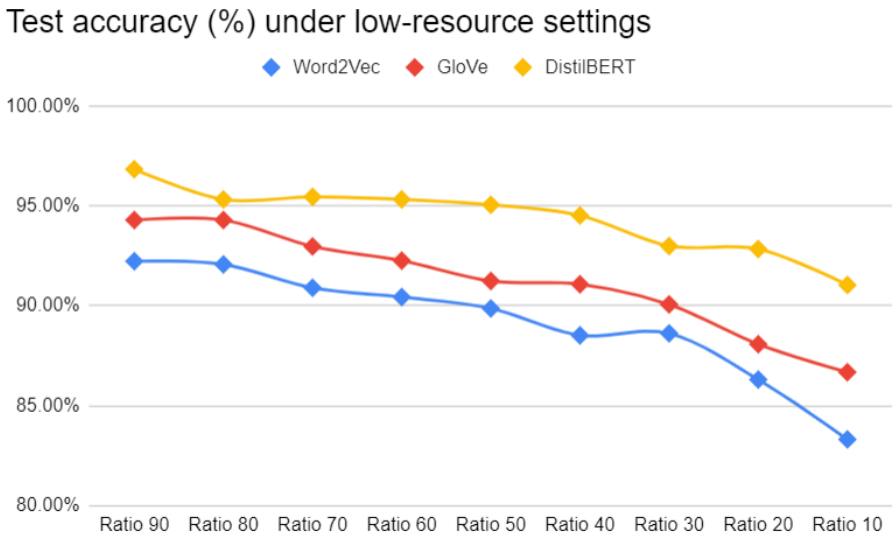

Figure 2: Test accuracy (%) of models in Phase 2.

As shown in the graph, the test accuracy of the representative models decreases as the percentage of training data decreases. Intuitively, this follows from the fact that performance of a model is directly related to its training, and in this case, the number of training examples. However, among all ratio settings, DistilBERT performs the best, followed by GloVe, and then by Word2Vec. This pattern is expected as DistilBERT is an advanced state-of-the-art transformer, while GloVe and Word2Vec are relatively simple word embedding approaches combined with a shallow biLSTM.

DistilBERT proved to be quite robust when placed under low-resource settings. It can achieve a test accuracy of 95% and 90% with only 50% and 10% of the available data, which is impressive. On the other hand, GloVe performed well when placed in harsh condition. It can achieve 90% test accuracy with only 30% of the available data. As a baseline, Word2Vec performed respectably with a 90% test accuracy with only 50% of the available data.

As our analysis has shown, DistilBERT and GloVe are relatively robust in environments with a significant scarcity of data. GloVe can achieve the same level of performance with Word2Vec using 40% less training data. Similarly, DistilBERT can achieve the same level of performance with Word2Vec using 80% less training data and the same level of performance with GloVe using 70% less training data.

6.3 Phase 3

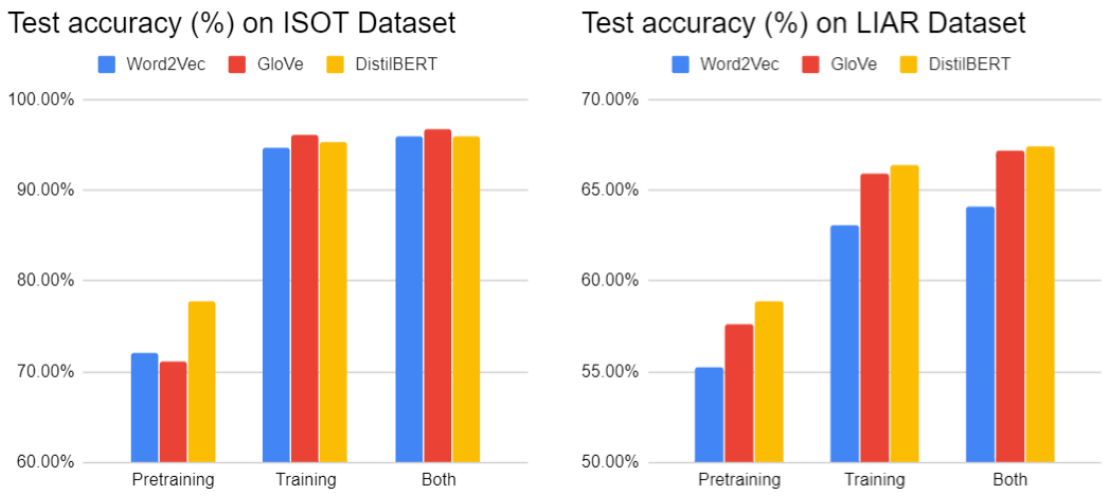

Figure 3: Test accuracy (%) of models in Phase 3.

6.3.1 ISOT Fake News Dataset

Among the three models, the diagram shows that pretraining and then fine-tuning performed better than training without any pretraining by only a slight margin. While this indicates that pretraining on the Real and Fake News Kaggle Dataset is beneficial, the effect is minimal as the ISOT Fake News Dataset contains a wider variety of news articles. This may imply that in general, pretraining on political news yields only a small increase in performance for other types of news.

The models which underwent training performed competitively on this new dataset; however, models with only pretraining performed poorly. Even the most advanced model DistilBERT had a test accuracy below 80%, proving the pretraining without any fine-tuning is far from sufficient.

6.3.2 LIAR Dataset

Similar to the ISOT Fake News Dataset, pretraining and then fine-tuning performed best, followed closely by training without pretraining, and with pretraining without fine-tuning performing noticeably the worst. Indeed, this also signifies that pretraining on the Real and Fake News Kaggle Dataset has a small positive effect on performance. However, compared to the results on the ISOT Fake News Dataset, the selected models performed significantly poorer on the LIAR Dataset, with the pretrained and fine-tuned DistilBERT having less than 70% test accuracy. This can be explained by the vast dissimilarity between the LIAR Dataset and the Real and Fake News Kaggle Dataset, especially since LIAR contains true and false statements instead of real and fake news.

6.3.3 Analysis on Pretraining

Following our results from the two datasets, we found that pretraining helps a model learn another dataset better, but the extent of the improvement depends substantially on the similarity between the dataset for pretraining and the dataset for fine-tuning, and also on the quality of the both datasets.

For the ISOT Fake News Dataset, our Word2Vec baseline without pretraining was able to perform fairly well, indicating that this dataset does not provide a meaningful challenge and is of relatively low quality, especially considering the train-valid-test split ratio of 2-0-98. Thus, adding pretraining to the model boosted the performance only a little despite the dataset's similarity with the pretraining dataset.

On the other hand, for the LIAR Dataset, all of our models without pretraining performed poorly. Even our most advanced model DistilBERT achieved a test accuracy under 70%, implying that two datasets are drastically different. Indeed, this is the case as several pieces of important contextual data in LIAR were removed during preprocessing.

In terms of the different models, DistilBERT dominates in transferring useful knowledge, as seen in the models with only pretraining. However, as mentioned earlier, this transferred knowledge can easily be overshadowed by newly learned information from fine-tuning and training on the new dataset.

7 Conclusion

In our study, we investigated different word embeddings and transformers for fake news detection. In Phase 1, due to the dominance of the text over the title, the Text and the Title+Text versions tied as the best. Furthermore, GloVe outperformed other word embedding approaches, while among the transformers, BERT performed outstandingly, followed closely by DistilBERT. In Phase 2, it was observed that DistilBERT is relatively robust and can withstand low-resource settings with only several hundreds of examples. Finally, in Phase 3, we found that pretraining on a similar dataset has a positive effect for all models. However, the magnitude of improvement is sensitive to the similarity between the two datasets and their quality of data.

References

Real and Fake News Kaggle Dataset:	https://www.kaggle.com/nopdev/real-and-fake-news-dataset
ISOT Fake News Dataset:	https://www.uvic.ca/engineering/ece/isot/datasets/
LIAR Dataset:	https://sites.cs.ucsb.edu/ william/software.html
Word2Vec:	https://arxiv.org/pdf/1301.3781.pdf
GloVe:	https://nlp.stanford.edu/projects/glove/
ELMo:	https://arxiv.org/pdf/1802.05365.pdf
BERT:	https://arxiv.org/pdf/1810.04805.pdf
ALBERT:	https://arxiv.org/pdf/1909.11942.pdf
DistilBERT:	https://arxiv.org/pdf/1910.01108.pdf
TorchText:	https://pytorch.org/text/
AllenNLP:	https://allennlp.org/
Huggingface:	https://huggingface.co/

Division of Labor

	Albert	Raymond
· Experiment Design	Methodology (Primary), Consulting, Finding Datasets	Methodology (Secondary), Selecting NLP Models
· Preprocessing Code	Phases 1, 2, and 3	Phase 3
· Model-related Code	Word Embedding Models (TorchText, AllenNLP)	Transformer Models (TorchText, Huggingface)
· Report	Writing, Formatting, Visualizations	Writing
· Video	Speaker, Presentation	Speaker, Presentation, Editing
Overall Contribution	60 %	40 %

Hyperlink to YouTube video

https://youtu.be/vbLEZKiNxpI