A Study on the Challenges for Fact-checking Models

A0222378A, A0227494W, A0254385X, A0256648N, A0217139E
Group 26
Mentored by Miao Yisong
{e0559910,e0657007,e0954786,e0966232,e0540696}@u.nus.edu

Abstract

In this information era, an overload of information online poses a challenge in judging whether the information is to be believed or not, which makes automated fact-checking extremely valuable. Current literature on fact-checking involves machine learning-based methods with handcrafted features, neural network-based methods, and prompt learning-based methods, among others. Most of them focused on designing models to achieve better performance for fact-checking tasks, while few considered the characteristics of the task and how they limit the existing models. In this project, we studied the fact-checking text classification task's distinctive traits (such as bias, length variations, subtle clues, and news corpus embedding), and analysed the performance of existing models with regards to these traits. Contrary to our intuition, we found that the overall writing style is the crucial feature for fact-checking tasks, while keeping the contents complete (global feature) doesn't help. However, focusing too much on certain tokens may lead models to be biased, which limits existing models' performance. [1]

1 Introduction

Spotting fake news is an indispensable skill, yet it is not as intuitive as some may think. 80% of Singaporeans are confident in their ability to spot fake news, but the reality shows that 90% of them are wrong when put to the test[2]. Moreover, due to the proliferation of social media platforms, fake news can spread quickly and lead to confusion and panic among users if left unchecked. Furthermore, fake news could lead to serious consequences regarding political outcomes and public opinion (Anderson and Sulistyani, 2020). Hence, it is imperative to develop methods to detect fake news.

There has been extensive research for fake news detection, but only a few focused on fake news' text characteristics. This inspired us to work on this direction. After a few runs of explorative experiments, we noticed the following three challenges: 1) significant length variation (§3.2) in the news domain; 2) a large number of Named Entities resulting in highly biased model (§5.3); and 3) high requirements on local features' capturing capability. In response, our study focused on exploring how these three aspects affect the performance of existing models, by masking Named Entities, summarising raw texts, and analysing how different models work separately.

Using the LUN (Labeled Unreliable News) dataset, we implemented various pre-processing steps and models to predict the news category: $\hat{y} \in \{\text{TRUSTED}, \text{SATIRE}, \text{HOAX}, \text{PROPAGANDA}\}$. We report 4 achievements: 1) obtained a high accuracy on the fact-checking text classification task; 2) analysed the bias of different models and determined which of them can be trained to be less biased; 3) ascertained that writing style is more important to fact-checking tasks, compared to complete semantic information; and 4) explained how each model works.

2 Related Work

News classification is a comprehensive task and tackles various problems such as fact checking (James and Andreas (2018)), stance detection (Kasnesis et al. (2021)), and sentiment analysis (Sanders et al. (2021)). Rashkin et al. (2017) attempted to address these problems by categorising news based on author intention and content truthfulness.

For pre-processing, Dogra et al. (2022) emphasised the importance of proper text representation and dimensionality reduction. Inspired by how humans process and classify texts, Haj-Yahia et al. (2019) calculated text similarity among the most relevant words in each document using a dictionary

[1] https://github.com/SweetBowl/4248_Unreliable_Fake_News_Detection
[2] https://str.sg/oeGD

of keywords for each category that reflects its semantics and lexical meaning. They also performed unsupervised learning, based on the documents similarity and the rich description of the category labels. In addition, embedding is commonly used to capture text semantics. Jang et al. (2019) tested 2 algorithms of Word2Vec model, and found that SBOW outperformed skipgram at news classification. Janakieva et al. (2021) used Vec2Doc model and achieved 99.97% accuracy for fake news detection.

For modelling, machine learning algorithms are widely used (eg. (Ksieniewicz et al., 2019)) and Dogra et al. (2022) compared different machine learning models for their advantages and limitations.

The above related work inspired us to explore the distinctive characteristics of fake news detection, and provided valuable guidance for our preprocessing, modelling and reasoning.

3 Corpus Analysis & Method

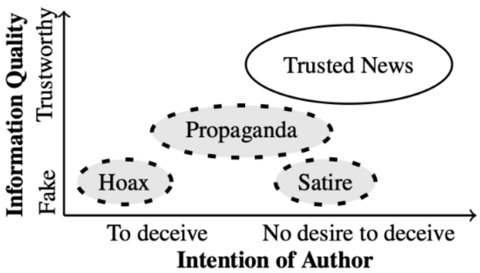

Figure 1: Types of news articles based on the intent and information quality by Rashkin et al. (2017)

3.1 Dataset

The dataset was originally constructed by Rashkin et al. (2017). It consists of more than 40k news records, each comprising of single or multiple sentences. As shown in Fig 1, the dataset provides a label for each record: trusted, satire, hoax and propaganda. Trusted refers to genuine news. Satirical pieces often employ humor and exaggeration with the intention to mock or criticise. Both hoax and propaganda tries to resemble real news to mislead audiences as a joke or for a political cause, respectively.

3.2 Methods

As shown in Fig 2, our pipeline consists of three parts: summarisation, pre-processing, and modelling. For pre-processing, we employ the usual NLP techniques as well as masking Named Entities to remove bias. For summarisation, the significant length differences in news records was reduced by LexRank. For modelling, traditional machine learning models, neural network-based models and unsupervised models were implemented. In addition, pre-trained large language models were also used for news classification for comparison.

Text Selection & Summarisation The excessive length disparity in news text is problematic: padding all news to the maximum length (2k+) leads to a large number of meaningless padding tokens in the short news, as well as a large memory consumption; on the contrary, direct truncation may result in information loss. In response, our poject summarised long texts using LexRank. As a graph-based method, LexRank calculates the cosine similarity among sentence pairs, constructs a graph based on the similarity matrix, and assigns each sentence a score using PageRank algorithm. The text summary is the concatenation of all sentences extracted.

Masking of Named Entities In fake news detection, named entities may introduce biases. For instance, if a name appears frequently in 'propaganda' news articles, the trained model may interpret this name as 'propaganda' news equivalent, which is not always true. To mitigate this issue, named entities were masked during pre-processing. It also makes our model more generalised for real life applications where the training data is free of named entities for privacy protection.

Traditional Machine Learning Classifiers

We started with unsupervised learning (k means) to gain a high-level understanding of the data distribution, including the frequent words for each label.

We then proceeded to train Naive Bayes model, which assumes conditional independence and ignores the impact of inter-word interactions on label prediction. It is fast to train and served as our baseline model. In comparison, Logistic Regression model handles high-dimensional input efficiently, and was used to explore fake news patterns by studying the inter-word interactions.

Neural Network-Based Classifiers

Neural Network-based methods are commonly used for text classification tasks mainly because: 1) they are good at capturing complex patterns; 2) they take texts' semantics into account, instead of just

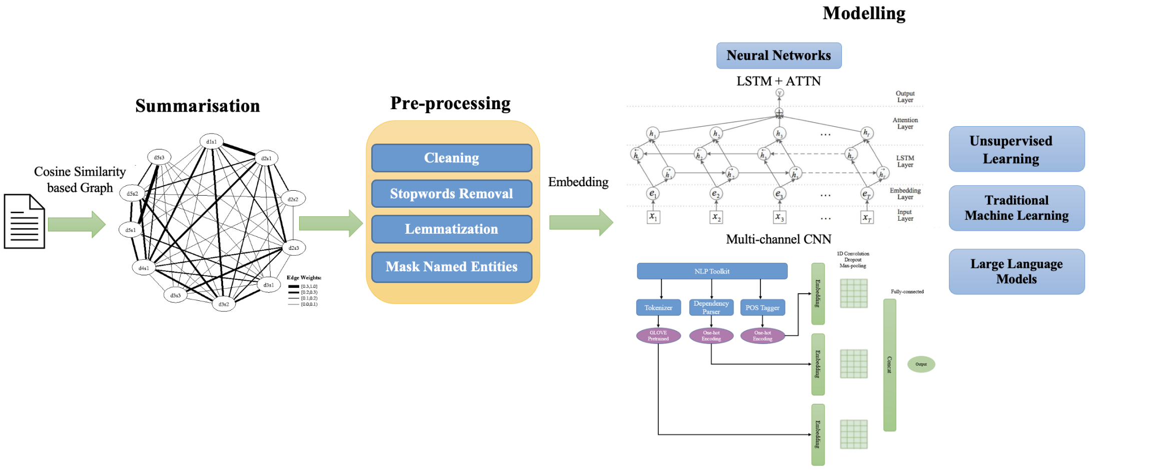

Figure 2: Proposed Pipeline. We firstly use LexRank to summarise news articles and further pre-process the data. After embedding processed text, four types of models are used to classify the news.

statistical features like traditional machine learning methods. Hence, we implemented MLP (Multi-Layer Perceptron), LSTM (Long Short-Term Memory Network) and CNN (Convolutional Neural Network) and explored two variations.

MLP is a simple but efficient deep learning model, which is able to capture non-linear complex patterns. It performs well for relatively simple input data. As some news articles have obvious clues showing their reliability, MLP was used as our baseline neural network model.

Compared to MLP, LSTM takes the order of tokens into account. It is useful for detecting fake news, which tends to re-arrange facts and opinions, or disorganise the information to make it more difficult for readers to follow the logic of the argument. Moreover, LSTM is good at capturing long-term dependencies, making it a good candidate to understand the lengthy news records.

However, LSTM suffers from 1)strong dependency on last hidden state, which may not well represent the previous context; 2) deteriorated performance when the input data contains much irrelevant or misleading information. Thus, we integrated LSTM with Attention Mechanism. Let H denote all hidden states at all time steps, w denote a set of trainable parameters with the size of $(hidden\_dimension, 1)$, α denote the attention score. The calculation process is as follows:

$$M = tanh(H) \tag{1}$$

$$\alpha = softmax(w^T M) \tag{2}$$

$$r = H\alpha^T \tag{3}$$

Finally, the classification result is based on the r. We expected the attention mechanism to help the model focus on more important features of the input data.

Fake news producers may use specific language patterns or phrases to deceive readers. We overcame the subtle classification clues challenge by implementing Text-CNN model, which is good at capturing local features.

Syntactic features can be helpful for fake news detection too. For example, POS tags can be utilised to reveal certain grammatical structures used to confound the readers. We integrated syntactic features into Text-CNN and expected an improved performance Mehdy et al. (2019).

Large Language Models

GPT-3.5 is pre-trained on large-scale text data, and thus is capable of capturing a wide range of linguistic patterns and subtle clues in news articles. It served as the SOTA model for our project.

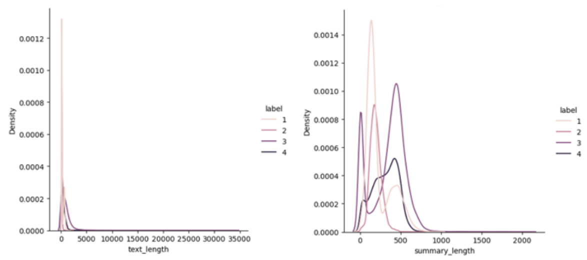

Figure 3: Length distribution before (Left) and after (Right) summarisation

Table 1: Models results on different text data. Scores are F1 macro score except accuracy for GPT-3.5.

Models	Processed Original Text	Processed Summary	Processed Masked Summary
Naive Bayes	0.906	0.878	0.858
Logistic Regression	**0.947**	0.939	0.913
K-means	0.610	0.572	0.371
MLP	0.776	0.610	0.663
LSTM	0.940	0.857	0.898
LSTM+Attention	0.945	0.883	**0.927**
CNN	0.932	0.854	0.885
CNN W/ Syntactic	-	-	0.899
GPT-3.5	-	**0.993***	0.782*

4 Experiments

4.1 Data Pre-processing

Text summarisation We firstly remove one data with length of more than 110K, which is nearly four times longer than the second longest text. And then we set the threshold to 500, i.e., we only summarise text with length larger than 500. After analysing the distribution of summarisation length, we set the number of summarised sentences to 20. Fig 3 shows that the summarised text distribution is less extreme.

F.B.I **ORG** had been under immense political pressure by Mr. Trump **PERSON** to dismiss Mr. Strzok **PERSON**, who was removed last summer **DATE** from the staff of the special counsel, Robert S. Mueller **PERSON**. The president has repeatedly denounced Mr. Strzok **PERSON** in posts on Twitter **EVENT**, and on Monday **DATE** expressed satisfaction that he has been sacked.

Figure 4: Masked named entities

Masked named entities As shown in Fig 4, we replace named entities with their generic tags that do not reveal their specific identity. By removing the specific names of entities, we force the models to focus on the actual content of the articles. We expect this to improve the robustness and reliability of the models.

Cleaning To reduce noise information, we firstly remove some special character by only keeping words, numbers and some punctuation like exclamation and question marks. Also, we do the case folding to reduce the complexity of the dataset.

Stop words removal Stopwords provide no meaningful info but occupy memory space, thus we removed them using customised NLTK library: several meaningful words were still retained, including negation (eg. no, never), subjective (eg. only, very), and some conjunction (eg. but) words.

Lemmatisation We lemmatise the words to further simplify the dataset

4.2 Word Embedding

To improve news classification by considering the semantic meaning of the news, we experimented two types of embedding on Logistic Regression:

Averaged Word2Vec Embedding Word2Vec embedding matrix is obtained from CBOW and skipgram models (both yield similar F1) using top 10k words. Sentence embedding matrix is the average of component words' embedding matrices.

Doc2Vec Embedding gensim Doc2Vec model is trained on training set, and then used to transform both training and testing set.

For Neural Networks, we utilised **Pre-trained GLOVE** (200 dimensions pre-trained on Wikipedia) to embed tokens, which helps us extract semantic features better. For Multi-channel CNN, we embedded two syntactic features via **one-hot encoding** and a linear layer.

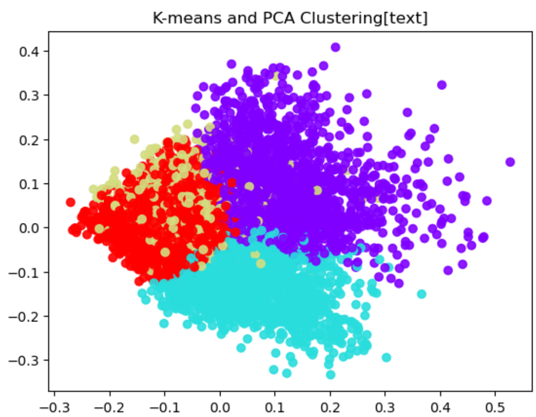

Figure 5: K-means clustering after PCA

4.3 Unsupervised Methods

We use TF-IDF as input features for K-means. And to visualise the results of K-means clustering, we use Principal Component Analysis (PCA) to reduce the dimensionality of the text data and plot it in two dimentsions. The clustering and PCA result is shown in Fig 5.

4.4 Machine Learning Methods

The input of Naive Bayes and Logistic Regression is each document's TF-IDF vector. For Logistic Regression, we set the $penalty$, $solver$ as $l2$ and $liblinear$ separately. We utilised the $MultinomialNB$ model for Naive Bayes and set fit_prior as $True$.

4.5 Neural Network-based Methods

For Deep-learning methods, a training strategy is crucial to get ideal results. At first, the LSTM's loss function suddenly increased after arriving at a certain level (Fig.10). Manual learning rate decay (Fig. 11) solves the problem and improves its performance. Hence, for all Neural Networks, we reset the learning rate as $0.9 * learning\_rate$ after every three epochs.

Generally, for all the models, the number of epochs, the batch size is 32, 512 separately, the maximum length for all input texts is 128, the loss function is a cross-entropy loss function, and we used the Adam optimiser to update parameters. For LSTM and its variation, we set the parameters $hidden\_dim$ and num_layers as 128 and 2 respectively. For Text-CNN and its variation, we separately set the parameters $kernel\_sizes$, $num\_filters$ as $[3, 4, 5]$ and 100.

During the experiments, there were also 2 observations: 1) CNN converges faster than LSTM (Fig.12 just shows the first 4 epochs, all models converge within 32 epochs); 2) CNN is less time-consuming for training (Table 2). Hence, we conjectured that local features contribute to faster convergence, and Convolutional Layers are more effective than LSTM.

Table 2: CNN, LSTM, LSTM+ATTN's Parameters

Models	# Parameters	Seconds / Batch
CNN	241504	0.0030
LSTM	301572	0.0086
LSTM+ATTN	301700	0.0088

4.6 Large Language Models

We use 8-shot to prompt GPT-3.5-turbo[3]. Due to time limit, we only tested a subset of the test set containing 1457 data, with the same proportion of data under different labels as in the original test set. All prompt used is listed in the Appendix.

5 Discussion

5.1 What is the impact of embedding techniques

Two types of embedding were applied to the best performing Logistic Regression model: averaged Word2Vec embedding and Doc2Vec embedding. Results are shown in Table 3, a few insights are drawn as follows:

Averaged Word2Vec embedding achieved a noticeably worse performance compared to Doc2Vec (Row 1.1 & 2). This is because Word2Vec considers local context only (i.e. single sentence), whereas Doc2Vec captures additional inter-sentence relationship. For Word2Vec, we did not horizontally stack each word's embedding matrix to form the sentence embedding matrix, because it would cause the model overfit: the longest news record after summarisation has 2k words and thus would produce 600k features (recall that vector size is 300), which is much larger than the 34k sample size. We would recommend to test out the horizontally stacked Word2Vec vectors after summarising the sentences further, collecting more samples and upgrading computing resource.

For our dataset, Doc2Vec classification performance was optimised using top 1% vocab (Row 2.1-2.4). When % vocab was too small, Doc2Vec did not capture enough semantics; when % vocab was too large, the embedding vector was diluted with meaningless common words. In addition, model performance also improved with as vector size increased, but the improvement diminished after vector size of 600. (Row 2.4-2.6, 2.8)

Doc2Vec embedding made the model performance less prone to Named Entity masking. Referring to Row 2.6-2.7 and Table 1, masking decreased TD-IDF / embedding F1 by 0.036 / 0.016, respectively. However, when vector size increased, the embedding also became more sensitive to masking as it retained more info (Row 2.6-2.9). Thus for a less-biased news classification, too large vector size shall be avoided.

[3]https://platform.openai.com/docs/models/gpt-3-5

Table 3: Impact of Embedding to Logistic Regression Model

No.	Embedding Method	Vector Size	Vocab Size*	Minimum Word Count*	% Vocab	Masked	F1
1.1	Averaged Word2Vec Embedding	300	10k	50	top 3%	N	0.820
2.1	Doc2Vec Embedding	300	25k	10	top 7%	N	0.893
2.2		300	10k	50	top 3%	N	0.896
2.3		300	3k	250	top 1%	N	0.899
2.4		300	1k	1250	top 0.3%	N	0.895
2.5		150	3k	250	top 1%	N	0.881
2.6		600	3k	250	top 1%	N	**0.903**
2.7		600	3k	250	top 1%	Y	0.887
2.8		1200	3k	250	top 1%	N	**0.903**
2.9		1200	3k	250	top 1%	Y	0.884

*To control % vocab for embedding, Word2Vec / Doc2Vec use vocab size / min word count, respectively

Dov2Vec embedding underperformed TF-IDF by 0.036 (Row 2.6 and best TD-IDF F1 of 0.939 on summarised text.). TF-IDF uses the whole vocab, including less-frequent words with special meaning. In contrast, embedding captures sentence semantic based on a small portion of the vocab only (0.3-7% in all runs). Therefore, sentences mainly made up of less-common (but possibly with special meaning) words are not well-interpreted. Tying back to the above-mentioned best % vocab of 1%, it optimised the overall F1, but not individual news record's. Nevertheless, this 0.036 gap is expected to be narrowed down by training Vec2Doc model on a larger collections of news corpus.

In short, for news classification task, Doc2Vec is preferred over Word2Vec due to its noticeably higher F1. By training on a larger news corpus, its F1 is expected to catch up with TD-IDF, and can be used to classify news in a less biased manner.

Table 4: CNN trained on the first 128 tokens V.S. CNN trained on the last 128 tokens

Corpus	Acc.	Pre.	Recall	F1
First 128	0.936	0.928	0.937	0.932
Last 128	0.935	0.928	0.932	0.930

5.2 Is complete semantic information important for fact-checking?

After summarisation, we preserved the most relevant information instead of truncating them directly, but the performance of all models dropped, which contradicted our initial hypothesis. Thus we carried out further analysis.

For the 3 machine learning models (Logistic Regression, Naive Bayes, and K-Means), they took the TF-IDF vector of documents as the input. Summarisation caused information loss and shrinked vocabulary size, although it kept more relevant information. Therefore, the removal of certain low-frequency tokens, which were originally crucial for classification, negatively affected the model performance.

For deep learning models, they rely on semantic features of the text, and thus shall benefit from summarisation as it helps to preserve the most relevant news content. For the resulting lower F1, we suspected that retaining full content may not be beneficial for fact-checking tasks, and that it even removed some crucial subtle clues due to reduced text fluency. To validate our suspicion, we conducted one more experiment: we trained the CNN model on the first and last 128 tokens of each news record, respectively, and they resulted in similar performance. This corroborated our previous conjecture that, for fake news classification, capturing writing style is more important than preserving complete news content.

5.3 Do models have a bias against named entities?

Interestingly, masked Named Entities affected different model performance differently: traditional machine learning methods and large language models experienced a significant drop in F1 score, whereas deep learning models had an increased F1. This observation indicated that these models were biased to different degrees.

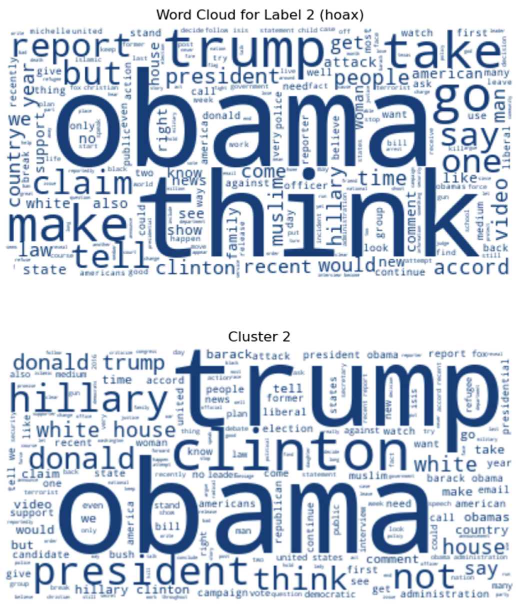

Figure 6: WordCloud for label 2 and K-means cluster 2

The large F1 drop suggested that machine learning methods are remarkably biased. These models may use specific named entities to classify news articles as shown in Fig. 6, thus masking named entities caused loss of information and lowered F1 score, as if these information were completely removed.

For deep learning models, their improved F1 can be explained from following three aspects: 1) when embedding using pre-trained GLOVE, named entities might already be mapped to <unk>, which conveyed little information. Replacement with Named Entities labels granted these <unk> tokens more meanings such as 'PERSON' and 'DATES'; 2) masking named entities helped to eliminate noise in the raw texts, allowing the models to focus on more useful patterns; 3) masking named entities also made the models become more generalised, leading to better performance on the test set.

For large language models, GPT-3.5 performed almost perfectly on unmasked data with an accuracy of more than 99%. However, its performance declined significantly after named entities were masked. Firstly, texts became more vague after being masked, and provided less clue for GPT to reason about; secondly, the results indicated the presence of bias in LLMs, which may arise from a biased training corpus.

5.4 How different models work?

Why clustering performed significantly worse than others?

The first aspect is input data. Firstly we simply transformed input using TF-IDF, which might not be discriminative enough. Secondly, the input is extremely high-dimensional, and unfortunately clustering algorithms are not good at handling high-dimensional data.

Another aspect is how to map a cluster to a specific label. For simplicity, we assigned each cluster the label that donimates in that cluster. However, that might yield suboptimal result as the label distribution among clusters can be quite close sometimes. Besides, as shown in Fig. 5, clusters overlap with each other, indicating some points cannot be definitely assigned to one cluster.

Although clustering did not achieve a good performance, Fig 6 shows that there are some clusters containing similar word frequency with ground truth labels. Unsupervised and semi-supervised methods may be a promising direction for future work, especially when labeled data is limited.

Why is syntactic features' effect on CNN limited?

As shown in Table 1, integrating syntactic features with CNN improved baseline CNN, but the effect was limited. To find out the reason, we tried to interpret the black box by visualising parameter updates to find out which channels contributed more.

In specific, we focused on the linear layer, whose input was the concatenation of the outputs from three channels. As this layer was where the three channels interacted first, we tried to interpret its training process. Each channel had 3 kernel sizes and 100 kernel filters, so the linear layer totally had 900 weights for each predicted label (4 labels text classification). Among them, the first 300 parameters corresponded to the semantics channel; the middle 300 parameters were about POS Tag channel; and the last 300 weights were related to the parsing features.

In general, multi-channel CNN achieved more than 90% accuracy on the training set within 8 epochs. We plotted heatmaps (Fig.7) to illustrate how much each parameter changed in every epoch. Images indicates that syntactic features converged within just 3 epochs, while semantic features needed more gradient descent; as for the parameters' values after convergence, the weights

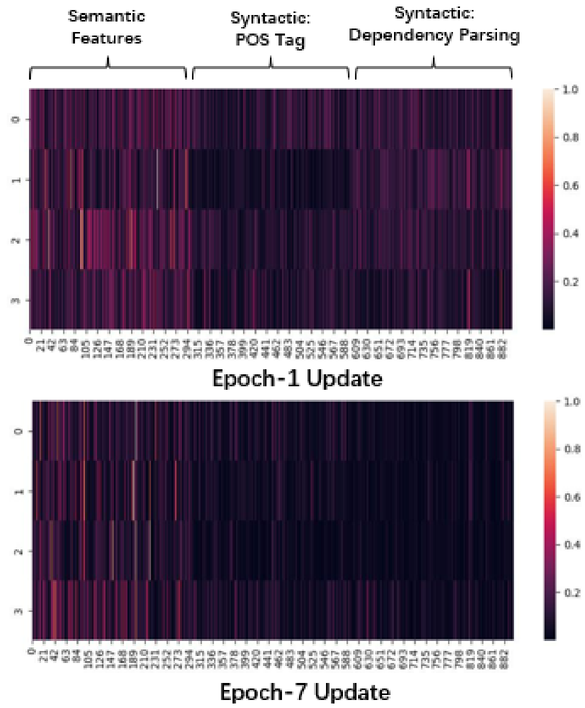

Figure 7: Weights Update on different epochs

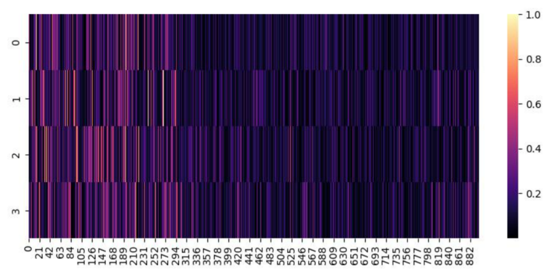

Figure 8: The weight values of the linear layer where three channels first interact after converging

of tokens' representations were also significantly larger than the other two channels' (Fig. 8). Hence, there are 2 reasons for syntactic features' limited improvement: (1) syntactic converged rapidly, and the pattern was simple; (2) the weights of syntactic features were tiny. The results indicate that syntactic features have the potential to boost fact-checking networks, but existing methods have not integrated them effectively.

What insight we can get from best performing LSTM with Attention?

According to the results, there are two observations: 1) Attention generally improves the performance of baseline LSTM (as we expected); 2) LSTM with Attention Mechanism is influenced the least after masking named entities. A probable reason is that Attention helps LSTM focus on more crucial tokens, and the model pays more attention to other keywords after masking the original important tokens. For example (Fig.9), after removing stopwords, for "President Barack Obama travels Guadalajara Mexico sunday twoday meeting", the model focused on the token "Obama". After masking named entities, attention was transferred to other tokens. The model's good results also indicate the importance of capturing local features.

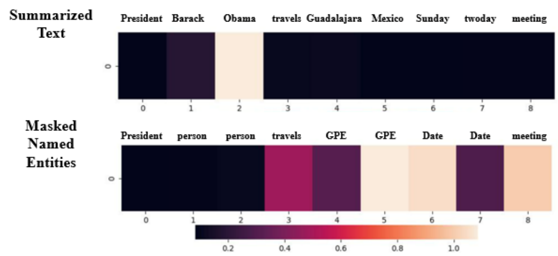

Figure 9: Compare the attention score before and after masking Named Entities

6 Conclusion

This paper analyses the intrinsic properties of news and models, including subtle clues, key information extraction, model bias, and embedding effects. We have the following conclusions: 1) maintaining authors' writing styles is more important than keeping the news contents complete; 2) masking named entities can help to reduce bias, and deep learning models become less biased in a more efficient manner compared to machine learning methods; 3) syntactic features have the potential to boost fact-checking models, but the way of integrating syntactic features with semantic features is not effective for now; 4) embedding makes the model less prone to bias, and is expected to achieve high performance after trained in larger corpus.

Overall, our analysis highlights the importance of considering the specific characteristics of models and input data, and LLMs may not be the panacea for all NLP tasks. Their strengths and weaknesses should be carefully considered with reference to the context of specific tasks. Due to time constraint, our project has following limitations: 1) only explored one single dataset; 2) did not re-implement each model multiple times to ensure result robustness; 3) yet to cover more popular models.

Future work may consider the topics covered in this report and further explore how to mitigate fact-checking models bias, and how to integrate features from other models (like syntactic features) more effectively.

References

Samuel Anderson and Hapsari Sulistyani. 2020. Detecting and combating fake news on web 2.0 technology in the 2019 political season indonesia. *Journal of Social Studies (JSS)*, 15:103–116.

Varun Dogra, Sahil Verma, Kavita ., Pushpita Chatterjee, Jana Shafi, Jaeyoung Choi, and Muhammad Fazal Ijaz. 2022. A complete process of text classification system using state-of-the-art nlp models. *Computational Intelligence and Neuroscience*, 2022:1–26.

Zied Haj-Yahia, Adrien Sieg, and Lea Deleris. 2019. Towards unsupervised text classification leveraging experts and word embeddings. pages 371–379.

Thorne James and Vlachos Andreas. 2018. Automated fact checking: Task formulations, methods and future directions. *CoRR*, abs/1806.07687.

D. Janakieva, G. Mirceva, and S. Gievska. 2021. Fake news detection by using doc2vec representation model and various classification algorithms. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 223–228.

Beakcheol Jang, Inhwan Kim, and Jong Wook Kim. 2019. Word2vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE*, 14(8):1–20.

Panagiotis Kasnesis, Lazaros Toumanidis, and Charalampos Z. Patrikakis. 2021. Combating fake news with transformers: A comparative analysis of stance detection and subjectivity analysis. *Information*, 12(10).

Pawel Ksieniewicz, Michał Choraś, Rafał Kozik, and Michał Woźniak. 2019. Machine learning methods for fake news classification. In *Ideal*.

Nuhil Mehdy, Casey Kennington, and Hoda Mehrpouyan. 2019. Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks. In *Security and Privacy in New Computing Environments: Second EAI International Conference, SPNCE 2019, Tianjin, China, April 13–14, 2019, Proceedings 2*, pages 152–177. Springer.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. pages 2931–2937.

AC Sanders, RC White, LS Severson, R Ma, R McQueen, HC Alcântara, Paulo, Y Zhang, JS Erickson, and KP Bennett. 2021. Unmasking the conversation on masks: Natural language processing for topical sentiment analysis of covid-19 twitter discourse. *AMIA Jt Summits Transl Sci Proc*, pages 555–564.

Acknowledgements

We would like to thank our project mentor Miao Yisong for his invaluable guidance throughout our project.

Statement of Independent Work

1A. Declaration of Original Work. By entering our Student IDs below, we certify that we completed our assignment independently of all others (except where sanctioned during in-class sessions), obeying the class policy outlined in the introductory lecture. In particular, we are allowed to discuss the problems and solutions in this assignment, but have waited at least 30 minutes by doing other activities unrelated to class before attempting to complete or modify our answers as per the class policy.

We have documented our use of AI tools (if applicable) in the following table, as suggested in the NUS AI Tools policy[4]. This particular document did not use any AI Tools to proofcheck and was constructed and edited purely by manual work.

Signed, A0222378A, A0227494W, A0254385X, A0256648N, A0217139E

Ethical Statement

This project does not raise any ethical issues.

A Appendix

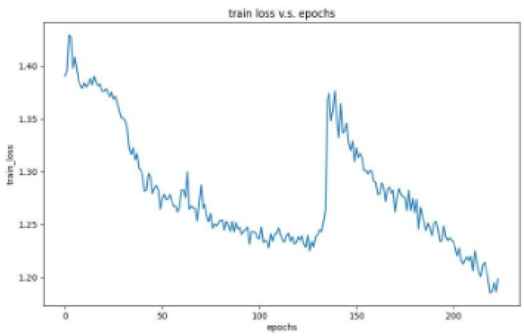

Figure 10: Training W/O Learning Rate Decay

[4] https://libguides.nus.edu.sg/new2nus/acadintegrity, tab "AI Tools: Guidelines on Use in Academic Work"

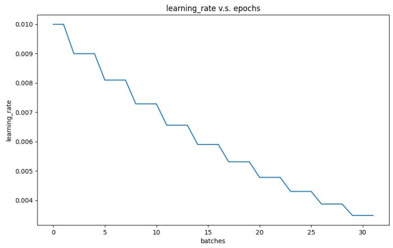

Figure 11: Manual Learning Rate Decay

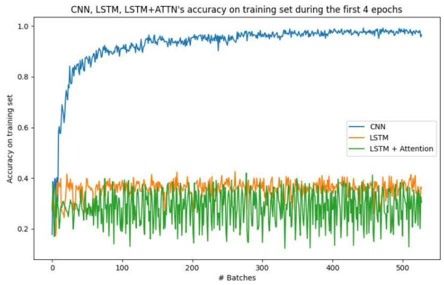

Figure 12: CNN, LSTM, LSTM+ATTN's accuracy on training set during the first 4 epochs

Table 5: Examples of few-shot of news classification for **summary** text.

SYSTEM: You are a helpful assistant that can classify the political news articles.
USER: There are some political news. Some of them are fake news, some of them are reliable news. You need to classify the pilitical news articles into 4 categories: 1. Satire: Mimics real news but still cues the reader that it is not meant to be taken seriously 2. Hoax: Convinces readers of the validity of a paranoia-fueled story 3. Propaganda: Misleads readers so that they believe a particular political/social agenda 4. Reliable News: Presents the facts Here is the first political new article. Please classify it into one of the four categories above. Article: Longtime television viewer Abraham Frank, 78, expressed outrage Tuesday over a Married With Children episode in which a swimsuit beauty pageant featured many scantily clad young ladies, many with oiled skin. 'I am outraged,' Frank said. 'I did not care for that program one bit.' In the past, Frank has been outraged by televised displays of violence, interracial romance and the use of the word 'booty.
ASSISTANT: Class: Satire

Table 6: Examples of few-shot of news classification for **masked summary** text.

SYSTEM: You are a helpful assistant that can classify the political news articles.

USER: What does Name Entity mean in natural language processing? And give me some Name Entity types

ASSISTANT: NER stands for Named Entity Recognition, which is a technique used in natural language processing to identify and extract named entities from text. Named entities are words or phrases that refer to specific entities such as people, organisations, locations, dates, and other types of entities. NER algorithm is useful in various applications such as information extraction, text classification, and sentiment analysis.

Here are some examples of Named Entity types:
- Person: Barack Obama, Michael Jordan
- Organisation: Google, Microsoft, United Nations
- Location: New York City, Paris, Mount Everest
- Date: January 1, 2021, 2022

USER: There are some political news. Some of them are fake news, some of them are reliable news. But Name Entities in these political news are masked, like Jack is masked as 'PERSON'. You need to classify the pilitical news articles into 4 categories with masked Name Entities.
1. Satire: Mimics real news but still cues the reader that it is not meant to be taken seriously
2. Hoax: Convinces readers of the validity of a paranoia-fueled story
3. Propaganda: Misleads readers so that they believe a particular political/social agenda
4. Reliable News: Presents the facts

Here is the first political new article. Please classify it into one of the four categories above.
Article: Longtime television viewer PERSON, DATE, expressed outrage DATE over WORK_OF_ART in which a swimsuit beauty pageant featured many scantily clad young ladies, many with oiled skin. 'I am outraged,' PERSON said. 'I did not care for that program one bit.' In the past, PERSON has been outraged by televised displays of violence, interracial romance and the use of the word 'booty.

ASSISTANT: Class: Satire