Classification Models for News Reliability Prediction

A0223917A, A0275756M, A0275611H, A0216840J, A0275647N
Group 25
Mentored by Rishabh Anand
{e0564958,e1127410,e1127265,e0540397,e1127301}@u.nus.edu

Abstract

Fake news has become a pressing issue due to the dangers it poses in areas such as politics and public health. Thus, fake news detection has become increasingly important. In language-based fake news detection, the attention mechanism has been used in several models achieving state-of-the-art performance. Hence, this project explores 3 attention-based models for fake news detection, specifically as a 4-way classification problem between Satire, Hoax, Propaganda and Real (Trusted) news. We applied these models on the Labeled Unreliable News (LUN) dataset and compared their performances. We found that the attention mechanism is effective for fake news detection and that hierarchical structures can further improve them. However, underperformance in Health news articles suggests that additional training data or a more specialised model may be suitable for detection in this domain. Codes for the project can be found in our GitHub repository.

1 Introduction

The advent of technology and recent major events like the Covid-19 pandemic have accelerated the shift towards a more online and mobile-dominated news media environment (Newman et al., 2023). With the improved convenience and faster dissemination of information came the rapid spread of fake news, which has led to negative impacts such as intensifying social conflict and promoting dangerous health practices (Allcott and Gentzkow, 2017). Despite this, consumers are often not well-equipped or are overconfident of their ability to identify fake news (Ipsos, 2018). Hence, it is crucial to detect fake news to protect against its harmful effects.

One approach to language-based fake news detection is to use manually engineered features such as content length in combination with probabilistic models such as Naive Bayes. More complex approaches make use of word embeddings and recurrence to encode contextualised representations of sequences, which often achieve better performance. Detection efforts further improved with the introduction of the attention mechanism and attention-only architectures (e.g. Kaliyar et al. (2021)). Although much research has been done in the field of fake news detection, many papers focus on either a 2-way classification between fake and real news or a multi-class classification on a scale of truthfulness (Misra and Grover, 2022).

As Rashkin et al. (2017) point out, it is important for fake news detection to be able to recognise both the article's veracity and the author's intent. To this end, they created the Labelled Unreliable News (LUN) dataset, which further divides fake news into 3 sub-categories: Satire, Hoax and Propaganda. Hence, our project focuses on modelling the fake news classification problem as a 4-way classification problem. Specifically, we train 3 attention-based classifiers for 4-way classification on the LUN dataset and compare their performances. We discuss the suitability of different model types and substantiate with empirical results. We also discuss the attention weights of our attention-based models and how they relate to classification performance.

2 Related Work / Background

Past research into language-based fake news detection have revealed several insights about the linguistic differences between fake and real news. Zhou et al. (2020) identified several useful features for fake news detection such as the writing style, total number of words, presence of informal words and number of unique words. Rashkin et al. (2017) analysed 3 different types of fake news (satire, hoax and propaganda) and found that fake news articles tend to use more personal language, such as first and second-person pronouns, and provide fewer statistics compared to trusted articles. These features have been useful in fake news detection through simple models such as the Naive Bayes classifier.

With the rise of deep learning and availability

of large corpora, fake news detection saw great improvement. A significant breakthrough was word embeddings, which capture the semantic and syntactic properties of words in an efficient vector space representation (Sezerer and Tekir, 2021). Recurrent Neural Networks (RNN) and variants such as Long Short Term Memory (LSTM) networks have also proven to be useful in fake news detection due to their ability to handle sequences and hence contextualise words (Mahara and Gangele, 2022). Next, the attention mechanism by Bahdanau et al. (2014) improved the abilities of RNNs to handle longer sequences. This has been applied in satire detection (Yang et al., 2017) and 4-way fake news classification through models such as the Graph Attention Network (Vaibhav et al., 2019).

More recently, (Vaswani et al., 2017) proposed the Transformer architecture, which does not make use recurrence or convolutions and are solely based on (self-)attention. This led to the proposal of the Bidirectional Encoder Representation for Transformer (BERT) by Devlin et al. (2019), as well as student models like DistilBERT (Sanh et al., 2019). State-of-the-art performance has been achieved by the BERT-based models on downstream tasks including multi-class fake news classification (Shushkevich et al., 2023).

3 Corpus Analysis & Method

3.1 Labeled Unreliable News (LUN) Dataset

The LUN dataset by Rashkin et al. (2017) consists of 51854 news documents (48854 in training split; 3000 in test split) categorized into 4 labels – satire, hoax, propaganda and trusted. It should be noted that the training and test splits contain articles from different news sources and thus have different distributions. We explore this in detail later.

To deepen our analysis of the models we implement, we augment the test data in the LUN dataset by dividing it into 6 news categories – Business, Entertainment, Health, Politics, Sports and News (representing general news that does not fit into other categories; usually about specific events). The labelling was done using a DistilBERT classification model fine-tuned on the CNN News Articles dataset[1] by user `IT-community` on HuggingFace Hub[2].

[1]Dataset available from https://www.kaggle.com/datasets/hadasu92/cnn-articles-after-basic-cleaning

[2]Model available from https://huggingface.co/IT-community/distilBART_cnn_news_text_classification

Dataset	Satire	Hoax	Propaganda	Trusted
Train	14047	6942	17870	9995
Test	750	750	750	750

Table 1: Label distribution between Train and Test datasets. **Train is imbalanced while test is balanced.**

3.2 Preprocessing

For our baseline model, we first converted the text into lower case to ensure consistency in the counting of words in later vectorization. This also helps to alleviate sparsity issues in the resulting vector. We also removed punctuations to get a more standardized version of each piece of text and focus more on words and semantic content themselves. Finally, lemmatization was done to reduce the inflectional forms of words and improve the efficiency of storage and retrieval.

For the attention-based models, only lowercasing is done for compatibility with the pre-trained components of the models. This is because punctuations, stop words and word forms can provide contextual information for these models to better encode the text. Details of tokenization and padding are explained in our experimental settings in §4.

3.3 Corpus Analysis

First, we analyzed the basic information about the datasets. We found that both training set and test set have only 2 columns, which are the texts and the corresponding labels. There are 48854 samples in the training set and 3000 samples in the test set with no missing or duplicate values.

Next, we examined the label distributions in the datasets (Table 1). We found that the label distribution in training set is imbalanced, that in the test set is balanced. Finally, we extracted several linguistic features from the datasets. One is the word count. We counted the average number of words in the text across four categories. Articles labelled satire and hoax have much shorter lengths with an average of around 200 words, compared to propaganda (averaging 900 words) and trusted (450 words). We also found that there are some extreme values with regard to the text length, which might be removed as outliers or utilized as useful features in the feature engineering part.

Another feature is the frequent words in articles of each class. On comparing the words with top 5 frequency across the 4 classes, we found some

Words	Satire	Hoax	Propaganda	Trusted
word1	said	obama	will	said
word2	just	think	can	will
word3	will	trump	people	percent
word4	one	will	one	one
word5	time	just	government	new

Table 2: Top 5 words in each class.

differences in the frequent words appearing in different classes (Table 2). For example, articles labelled hoax and propaganda have more political terms and names like Obama, Trump and government, while trusted articles have more statistical words like numbers, year and percent. In particular, we hypothesise that the statistical words may be useful features for our models and improve their performance on the "Trusted" class.

3.4 Baseline Models

We chose Multinomial Naive Bayes(MNB) as one of baseline models for its simplicity and efficiency. MNB can learn large datasets like LUN quickly and may even outperform some complicated models.

The word embedding technique is based on Term Frequency-Inverse Document Frequency(TF-IDF). Since MNB works with probabilities, term frequency is intuitively a useful feature as it measures how frequently a term occurs in a document:

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in a document } d}{\text{Total number of terms in the document } d}$$

Inverse Document Frequency measures how important a term is:

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Total number of documents } |D|}{1 + \text{Number of documents with term } t}\right)$$

Some versions of formula don't have the +1 in denominator(can set on `smooth_idf` parameter in Scikit-learn), which is often added to avoid division-by-zero when a term does not appear in the documents. TF-IDF score is simply the product of TF and IDF:

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

In TF-IDF vectorization, the process involves creating a matrix where each row represents a document and each column represents a term from the entire corpus vocabulary. The value in each cell is the TF-IDF score of the term in that specific document.

Besides word frequency, we also explored other features such as number of sentences, number of words and number of unique words in each piece of news. We attempted to improve baseline performance by concatenating these features and TF-IDF.

3.5 The Attention Mechanism

The attention mechanism (Bahdanau et al., 2014) is a key building block of our deep learning models. It draws inspiration from human processing of language. When processing documents, we selectively focus on the important parts to get the information necessary for understanding it. Moreover, whether we focus on a particular word often depends on the context. Similarly, the attention mechanism allows a neural network to selectively focus on features that it determines are relevant to the current task. Formally, given a query and a set of N key-value pairs, attention consists of 3 operations:

$$e_j^{(i)} = a(q_i, k_j) \tag{1}$$

$$\alpha_j^{(i)} = \frac{\exp(e_j^{(i)})}{\sum_{\ell=1}^{N} \exp(e_\ell^{(i)})} \tag{2}$$

$$c_i = \sum_{j=1}^{N} \alpha_j^{(i)} v_j \tag{3}$$

(1) computes the alignment between the query q_i and the key k_j using an alignment function a. (2) then takes the softmax to obtain a weight $\alpha_j^{(i)}$. Finally, (3) takes the weighted sum of the values v_j using the weights from (2), giving a representation that has "selectively focused" on each of the values.

3.6 Hierarchical Attention Network

We implement the Hierarchical Attention Network (HAN), which was proposed by Yang et al. (2016) as a model that creates better representations of documents for document classification compared to baseline models. The first key idea of the HAN is to exploit the hierarchical structure of documents, which refers to the idea that documents are composed of sentences that are in turn a sequence of words. This is done by building up sentence representations from word representations and subsequently building document representations from these sentence representations. The second key idea is to use the attention mechanism as explained in §3.5 so that the model can learn to pay more or less attention to each word or sentence.

The overall structure of the HAN is shown in Figure 1 and readers are advised to refer to the original paper for full technical details. We summarize the core components of the model below.

Word Embeddings The embedding layer represents each word as a real-valued vector that captures its properties, so that words that are closer in

the embedding space have similar properties (such as semantic or syntactic properties). For our implementation, this layer is initialized using the pre-trained Global Vectors for Word Representation (GloVe) embeddings by Pennington et al. (2014), who showed that GloVe outperforms similar sized models in semantic similarity tasks while maintaining competitive performance for syntactic tasks. Since syntactic and semantic patterns are both useful in fake news detection (de Beer and Matthee, 2021), we chose GloVe to initialize this layer.

Forming Sentence Representations A bi-directional Long-Short Term Memory (bi-LSTM) network is used to encode each sentence n of length T (represented by a sequence of word embeddings) into a sequence of hidden representations $\{h_{nt}\}_{t=1}^{T}$. Each h_{nt} is formed from the concatenation of the forward and backward hidden states produced by the forward and backward LSTMs at step t. Hence, h_{nt} contains information about the sentence with a focus on the part surrounding the t-th word of sentence n. These representations are passed into an attention layer, which starts with a 1-layer feed-forward neural network with a $\tanh$ activation function to form a sequence of keys. The query is a single trainable vector u_w that Yang et al. calls the "word-level context vector" and the same query is used for every key. The alignment function is a dot product. Since our input is padded, we mask out the steps corresponding to padding by replacing the alignment score with $-\infty$. In other words, equation (1) becomes:

$$e_{nt} = \begin{cases} -\infty & h_{nt} \text{ is padding} \\ u_w^T \tanh\left(W_w h_{nt} + b_w\right) & \text{otherwise} \end{cases}$$

where W_w and b_w are parameters of the feed-forward layer. Note that this may be different from the originally proposed HAN as the authors did not specify whether masking was used. Next, softmax is performed as stated in (2). This gives an attention weight of 0 for all padding values and non-zero for non-padding values, thereby ensuring that attention is only paid to actual words. Finally, $\{h_{nt}\}_{t=1}^{T}$ is used as the values for the weighted sum in (3). The resulting vector s_n is thus a representation of sentence n that encapsulates both the sequential context of its constituent words and their relative importances measured with respect to the trainable u_w. According to Yang et al., u_w may be seen as a high-level representation of the query "what is the informative word (for the task)?"

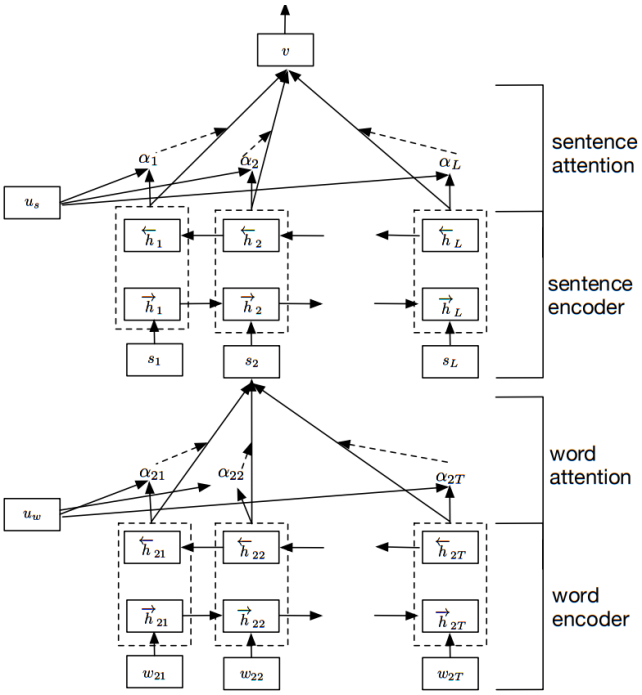


Figure 1: Architecture of the Hierarchical Attention Network proposed by Yang et al. (2016). Figure also produced by Yang et al. (2016). **The document vector v is a weighted sum of sentence encodings, which are in turn fromed from weighted sums of word encodings.**

Forming Document Representations Document vectors are formed by repeating the same concept at the sentence level. Given a length-L document consisting of a sequence of sentence vectors $\{s_n\}_{n=1}^{L}$, a second bi-LSTM encodes it, and another attention layer with a sentence-level context vector u_s transforms the encoding into a single vector that represents the document as a weighted sum of the sentence encoder hidden states.

Classification Since the resulting document vector represents the aggregated information of the news article, it is passed through a feed-forward output layer. No activation function is used as we used the PyTorch implementation of Cross Entropy Loss, which expects unnormalized logits as it performs softmax on its inputs.

3.7 Flat Attention Network

To investigate the importance of the hierarchical structure of HAN for our task, we also implement a "flat" version of the HAN for comparison, named the Flat Attention Network (FAN). Similar to HAN, the FAN starts with an embedding layer initialized from GloVe pretrained vectors. Unlike HAN, the FAN encodes the sequence of word embeddings of

an entire document using a bi-LSTM and passes these hidden representations through an attention layer to directly produce a document vector.

3.8 DistilBERT

To leverage on the power of parallelised Bidirectional encoding of transformers, we used a pretrained DistilBERT model (Sanh et al., 2019). DistilBERT is a student model of the Bidirectional Encoder Representation for Transformer (BERT), meaning that it has the same architecture but with fewer layers and is trained to reproduce the behaviour of BERT. Being a transformer, BERT uses self-attention – a variant of the attention mechanism where query, key and value are all the input sequence multiplied by learnable weight matrices W_q, W_k, W_v. The alignment function is a dot product, scaled by the dimension of the input features. Moreover, it uses multi-head attention, which allows it to compute multiple self-attention operations in parallel so that each head can learn different features of the input sequence. It also achieves bidirectional encodings by using a masked language model objective during pre-training, where the input sequence has a token masked out and the model is trained to predict the masked token based on the context from the remaining surrounding tokens. Devlin et al. (2019) showed that these representations of sequences can outperform concatenated uni-directional representations such as those by BiLSTMs. Hence, we implement this model to compare its effectiveness against the BiLSTM-based HAN and FAN.

Our model consists of adding a pre-classifier layer, which is a feed-forward neural network, on top of the pre-trained transformer. This is followed by a dropout layer and another feed-forward layer for classification. For training, we decided to freeze the layers in the pre-trained model and only trained the parameters in the pre-classifier, dropout and classifier layers. While this was largely due to compute constraints, we note that both van Aken et al. (2019) and Peters et al. (2019) found that fine-tuning BERT produces minimal changes in the model weights and hence downstream performance. Since DistilBERT is a student of BERT, these findings suggested that performance on our task will not be severely reduced by this decision.

4 Experiments

4.1 Model Configurations & Training

The final feature set used for the baseline MNB classifier was the TF-IDF vector. The additive smoothing factor is set at $\alpha = 0.01$. We didn't set up a validation set for MNB because TF-IDF vector was of large size and our machine often ran out of memory in its operation.

For the HAN and FAN, each document is split into sentences and tokenized using the Natural Language ToolKit's (NLTK) default tokenizer. For the HAN, each document is truncated/padded to have 30 sentences, and each sentence is truncated/padded to 30 words, whereas each document is truncated/padded to 500 words for the FAN. Embeddings are initialized from 100-dimensional GloVe embeddings pre-trained on the Wikipedia 2014 + Gigaword 5 corpus[3] and set to be trainable. Each bi-LSTM has a hidden dimension of 100, and the dimensions of the context vectors and 1-layer feed forward neural networks at word-level and sentence-level (for HAN) are all set at 200.

Both models were trained using the Adam optimizer with a weight decay factor of $\lambda = 5 \times 10^{-6}$ and a batch size of 256 for 10 epochs. The initial learning rate was set at 5×10^{-4}. Learning rate annealing was used as it has been shown to improve model generalization (Li et al., 2020). Specifically, we halve the learning rate for every 2 consecutive epochs where validation loss did not decrease.

For the DistilBERT classifier, each document is tokenized using the pretrained 'bert-base-uncased' tokenizer. Each document is padded or truncated to a max length of 512 words. The preclassifier layer has a dimension of 768 and is followed by a dropout layer with probability 0.5. The final classifier layer is a feed-forward layer of shape 768 by 4. The model was trained using the Adam optimizer for 5 epochs with training batch size of 8 and a constant learning rate of 1×10^{-5}.

HAN, FAN, and DistilBERT were trained using a 80:20 train-validation split of the training dataset, while MNB was trained on the entire training set.

4.2 Evaluation Metrics

To align with similar recent works (e.g. Padalko et al., 2024), we used accuracy, precision, recall and F1 scores to evaluate model performance. Since the training and validation splits each have

[3]Embeddings are available for download from https://nlp.stanford.edu/projects/glove/

Model	Training	Validation	Test
MaxEnt (Rashkin et al., 2017)	-	91	65
MNB	96.1	-	66.4
HAN	**99.9**	**97.2**	**75.6**
FAN	**99.9**	96.5	70.3
DistilBERT	92.9	92.4	63.0

Table 3: Macro-averaged F1 scores in percentage for classification on training, validation and test sets. **HAN achieved the best performance in all 3 sets.**

Model	Satire	Hoax	Prop	Trusted
MNB	65	46	69	80
HAN	85	64	71	83
FAN	81	58	64	78
DistilBERT	63	48	50	76

Table 4: F1 scores for classification on test set in percentage, broken down by labels. Prop stands for propaganda. **The trusted class performs consistently well.**

imbalanced label distributions, macro-averaging is used for each metric so that the dominant class does not skew the score. In our experimental results, we mostly report the F1 score as it provides the most robust evaluation of the models by taking into account both precision and recall.

4.3 Experimental Results

Since the test dataset contains documents from unseen sources and has a different label distribution, all models were tested on it to measure their abilities to generalize. Table 3 shows the results of our experiments and the Max-Entropy classifier by Rashkin et al. (2017) as an additional baseline.

Both the HAN and FAN outperform the baseline models. The HAN architecture proposed by Yang et al. (2016) had the highest F1 score on all 3 splits, in particular outperforming the DistilBERT classifier despite having about $\frac{1}{10}$ the number of parameters. Moreover, the FAN is able to achieve the same training F1 score and maintains a competitive validation score compared to the HAN despite being half the size. Since the validation set is similar to the training set, this suggests that the lack of hierarchical structure did not affect its ability to learn the training distribution. However, its score on the test set is 5.3 lower than the HAN, indicating that the hierarchical structure of the HAN has contributed to its generalization abilities. We explore this further in the subsequent section.

Category	n	HAN	FAN	DistilBERT
Business	179	69.5	59.4	47.6
Entertainment	112	73.2	57.1	55.2
Health	559	52.1	47.1	34.5
News	930	70.9	68.4	57.4
Politics	1051	70.0	61.1	53.1
Sports	169	69.9	66.3	63.0

Table 5: Number of articles (n) and macro-averaged F1 scores for classification per news category of test set. **All 3 models underperform on the Health category.**

We also observe that the DistilBERT model performed poorly compared to the other models. This could be due to our earlier decision to freeze the transformer layers and only train the feed-forward neural networks. These results suggest that the language used in news articles are significantly different from those used in Wikipedia articles and books in the Toronto BookCorpus on which BERT was originally trained. Thus, the encoder representations may not accurately reflect the nuances in news text that are useful for fake news detection.

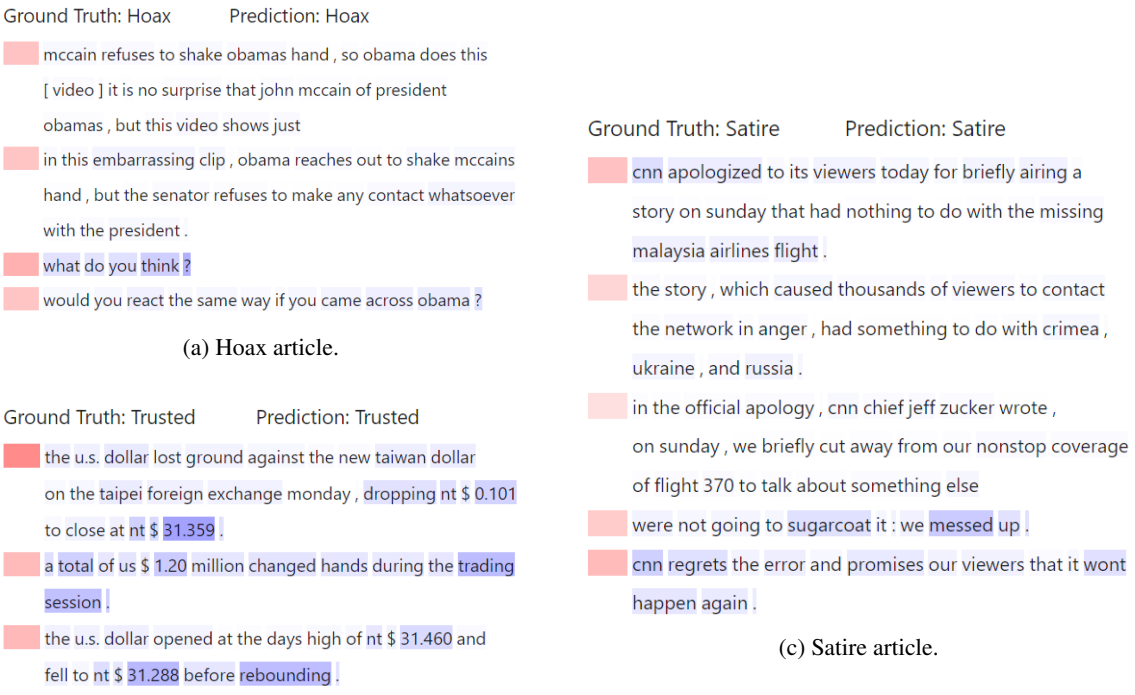
Finally, we further break down the model performance on each of the 4 labels in the test set. As seen in Table 4, the lowest scoring label for all models is "Hoax", likely due to the lack of "Hoax" articles in the training data as explored in §3.3. Interestingly, all the models perform consistently well on the "Trusted" label despite "Trusted" being a minority class in the training data. This supports our hypothesis that there are distinguishable features in the "Trusted" class that were useful for the models, such as statistical words (e.g. numbers). Lastly, we also notice that "Propaganda", despite being the majority class, has relatively poor performance in our attention-based models.

5 Discussion

This section goes in-depth into our experimental results and further analysis in order to understand how the models implemented have achieved the performance seen in §4.3. We first look at subpopulation performance before moving on to discuss the effects of the attention mechanism and hierarchical structure.

5.1 Subpopulation Performance

As explained in §3.1, we augmented the LUN test dataset by categorizing the articles into 6 categories.



(a) Hoax article.

(b) Trusted article.

(c) Satire article.

Figure 2: Attention Weights of HAN for a (a) hoax article, (b) trusted article, (c) satire article. The red box beside each sentence indicates the sentence weight. The blue box around each word indicates the word weight. **Attention weights focus on features such as questions, numbers and informal terms.**

We evaluated the FAN, HAN and DistilBERT models on each category and the results are shown in Table 5. As with the overall results, the HAN performed the best in each category, followed by the FAN and lastly DistilBERT.

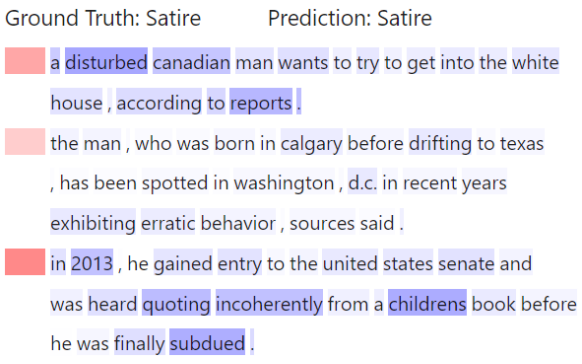
As seen in Table 5, "Health" had the worst performance compared to all other categories across all 3 models. One avenue of investigation we attempted was to label the training data with these categories as well. We found that "Health" is also the smallest class in our training data, constituting 1790 out of the 48854 articles. Hence, we believe that the lack of training examples is the main reason why all 3 models perform poorly in this category.

Beyond this, we also looked at the precision and recall of the models for each class among articles in the "Health" category. We found that precision for the "Trusted" class was remarkably low (21% and 18% for HAN and FAN respectively). Further inspection revealed that the model often predicted that Health articles as trusted when they were actually propaganda. Qualitatively, we found that these propaganda articles often made medical claims supported by sources and research, and that it was hard to determine the veracity or intent of these claims from the article alone. Hence, other than the lack of training examples, we surmise that the lack of domain knowledge may have caused poor performance. Thus, we believe fake news detection in the Health domain can benefit from an alternative approach of more specialised models making use of knowledge bases or domain-oriented datasets.

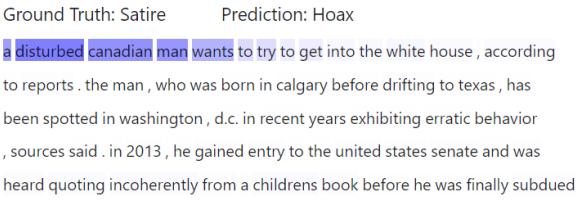
5.2 Analysis of Attention Weights

We visualized the attention weights of HAN, FAN and DistilBERT to understand which words were being given more weight for model predictions. Some examples of the visualizations for HAN are shown in Figure 2. Note that similar behaviour was seen for FAN for these examples.

Figure 2a is an example of how the models put more weight on question indicators (e.g. "?" and "what") and on sentences with questions. Meanwhile, figure 2c shows that some weight is put on casual terms such as "sugarcoat" and "messed up". Relatedly, Esteban-Bravo et al. (2024) found that question marks and exclamation marks were more likely to appear in the title of fake news than real news, while Pérez-Rosas et al. (2018) found that fake news tends to be more likely to use informal language and second-person pronouns. Our findings complement these and suggest that other than



(a) Attention weights from HAN.



(b) Attention weights from FAN.

Figure 3: Comparison of attention weights of (a) HAN and (b) FAN for a satire article. Word weights of HAN are normalized by sentence weight for fair comparison. **FAN has mostly ignored words past the first sentence.**

the title, the appearance of questions in the body text, especially those directed at the reader through second-person pronouns, are features associated with fake news.

Comparing Figures 2b and 2c, we observe that the attention weights for numbers (e.g. 370) in the satire article are small, while the weights for numbers (e.g. $31.359) in the trusted article are large. Numbers were often found to have larger weights in trusted articles within the corpus. Moreover, units such as "gallons" and "$" also often had larger weights in trusted articles, while numbers used in the context of a name such as "windows 8" had smaller weights. Similar behaviour was seen for the FAN in most cases (not shown in the figure). This suggests that both the HAN and FAN find numbers and statistical words representing quantities to be useful features for classifying the article as trusted. This could partially explain their high recall scores for the "Trusted" label (92% and 93% respectively). These findings support our earlier hypothesis and are also consistent with previous studies that have shown that words used to offer concrete figures such as numbers and money appear more often in truthful news than in fake news (Rashkin et al., 2017).

On the other hand, there were also some differences between the attention weight distributions of HAN and FAN as shown in Figure 3. There were several instances of the FAN's attention weights being concentrated in the first sentence of the document. Figure 3b shows a more extreme example where the weights of FAN are concentrated only in the first few words. In contrast, figure 3a shows that attention weights of HAN are spread across words in different sentences. They include strong subjective adjectives ("disturbed") and action adverbs which are associated with fake news (Horne and Adali, 2017). Moreover, the last sentence is seen to be the most important by HAN, whereas no weight was given by FAN. Since these attention weights are used to form the final document vector in each network, these findings indicate that the FAN is sometimes unable to incorporate information from different sentences. In contrast, the hierarchical structure of the HAN has enabled it to form a more representative document vector based on information from all its important sentences. This explains the performance differences between the 2 networks.

6 Conclusion

We have shown the effectiveness of the attention mechanism in 4-way fake news classification when combined with recurrent structures such as Bi-LSTMs. Further, our analysis found that hierarchical modelling of documents enhanced the abilities of the attention mechanism, producing more representative document encodings and thereby improving downstream classification performance. However, the underperformance of our models in the Health domain suggests that more training data or more specialised models may be needed. Finally, we recognise that our analysis of attention weights gives some level of interpretation but does not necessarily constitute explanation. Future directions building on our work should include verifying these findings with more well-understood model saliency techniques such as integrated gradients.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Dylan de Beer and Machdel Matthee. 2021. Approaches to identify fake news: A systematic literature review. In *Integrated Science in Digital Age 2020*, Lecture notes in networks and systems, pages 13–22. Springer International Publishing, Cham.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Mercedes Esteban-Bravo, Lisbeth d. l. M. Jiménez-Rubido, and Jose M. Vidal-Sanz. 2024. Predicting the virality of fake news at the early stage of dissemination. *Expert Systems with Applications*, 248:123390.

Benjamin D. Horne and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *CoRR*, abs/1703.09398.

Ipsos. 2018. Trust and confidence in news sources.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia Tools and Applications*, 80(8):11765–11788.

Yuanzhi Li, Colin Wei, and Tengyu Ma. 2020. Towards explaining the regularization effect of initial large learning rate in training neural networks.

Govind Singh Mahara and Sharad Gangele. 2022. Fake news detection: A rnn-lstm, bi-lstm based deep learning approach. In *2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS)*, pages 01–06.

Rishabh Misra and Jigyasa Grover. 2022. Do not 'fake it till you make it'! synopsis of trending fake news detection methodologies using deep learning. In *Deep Learning for Social Media Data Analytics*, pages 213–235. Springer.

Nic Newman, Richard Fletcher, Craig T. Robertson, Kirsten Eddy, and Rasmus K. Nielsen. 2023. Reuters institute digital news report 2023. Technical report, Reuters Institute for the Study of Journalism.

Halyna Padalko, Vasyl Chomko, and Dmytro Chumachenko. 2024. A novel approach to fake news classification using lstm-based deep learning models. *Frontiers in Big Data*, 6.

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. Association for Computational Linguistics.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

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. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Erhan Sezerer and Selma Tekir. 2021. A survey on neural word embeddings.

Elena Shushkevich, Mikhail Alexandrov, and John Cardiff. 2023. Improving multiclass classification of fake news using bert-based models and chatgpt-augmented data. *Inventions*, 8(5).

Vaibhav Vaibhav, Raghuram Mandyam Annasamy, and Eduard Hovy. 2019. Do sentence interactions matter? leveraging sentence level representations for fake news classification.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1823–1832, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. In *Proceedings of*

the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1979–1989, Copenhagen, Denmark. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Xinyi Zhou, Atishay Jain, Vir V. Phoha, and Reza Zafarani. 2020. Fake news early detection: A theory-driven model. *Digital Threats*, 1(2).

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 a 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.

If the production of your report used AI Tools (inclusive of Generative AI), do keep detailed logs of how you used AI Tools, as your project requires the accountability of an audit trail of your interaction(s) with such tools (prompts, output).

Signed, A0223917A (e0564958), A0275756M (e1127410), A0275611H (e1127265), A0216840J (e0540397), A0275647N (e1127301)

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