2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-7281-8470-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICMLA51294.2020.00127

Towards Machine Learning Explainability in Text Classification for Fake News Detection

Lukas Kurasinski
Department of Computer Science,
Malmö University
Malmö, Sweden
lukasz.kurasinski@gmail.com

Radu-Casian Mihailescu
Department of Computer Science,
Malmö University
Malmö, Sweden
radu.c.mihailescu@mau.se

***Abstract*—The digital media landscape has been exposed in recent years to an increasing number of deliberately misleading news and disinformation campaigns, a phenomenon popularly referred as *fake news*. In an effort to combat the dissemination of fake news, designing machine learning models that can classify text as fake or not has become an active line of research. While new models are continuously being developed, the focus so far has mainly been aimed at improving the accuracy of the models for given datasets. Hence, there is little research done in the direction of explainability of the deep learning (DL) models constructed for the task of fake news detection.**

In order to add a level of explainability, several aspects have to be taken into consideration. For instance, the pre-processing phase, or the length and complexity of the text play an important role in achieving a successful classification. These aspects need to be considered in conjunction with the model's architecture. All of these issues are addressed and analyzed in this paper. Visualizations are further employed to grasp a better understanding how different models distribute their attention when classifying fake news texts. In addition, statistical data is gathered to deepen the analysis and to provide insights with respect to the model's interpretability.

Keywords: Fake news, Deep learning, Explainability.

I. Introduction

The Internet has become a fast and cost-effective way of sharing information. Nowadays, more and more people rely on online media to provide information, news, and facts about the world via newsfeeds, online newspapers, and social media [1]. The Internet is reaching a broader public than any other paper publication ever did. With the vast amount of information it provides, an everyday user cannot keep up with fact-checking everything he or she consumes. Lack of in-depth knowledge on a subject, gives online sources the benefit of trust from the public, counting on the medium to provide reliable and honest information.

In parallel with the growth of reliable information sources available on the Internet, so did the spread of misinformation [2]. Distribution of false claims can happen, unknowingly by providing facts based on unreliable sources, or knowingly to confuse, mislead, and deceive the reader. Whatever the case may be, fake news has become a challenging problem that internet users face today. For this reason, the process of detecting false information is also becoming a subject of extensive study. There is a clear need for systems that make it possible to identify news articles providing malicious content. Systems based on manual identification of this type of material are costly and time-consuming. That is why an effort is being undertaken to find effective machine learning (ML) approaches to address these issues.

The problem turns out to be very difficult to solve, and researches are yet to find reliable solutions. While proposing different ML models to tackle the problem, researchers are primarily focusing on the effectiveness of a particular model, without trying to achieve an in-depth understanding why a particular model performs better than another one. In short, there is a lack of research concerning the explainability of a model, while effort is mainly focused on its effectiveness only. In this work, we aim to shed more light on the underpinnings of several state-of-the-art ML models used in fake news detection, in order to derive a more rigorous understanding for their performance. The essence of this work is to peek inside the black box architecture that is a neural network and try to understand the mechanisms behind it.

To this end, we investigate two classes of deep neural networks for the task of fake news detection. The first class is represented by an architecture that combines convolutional neural networks (CNN) and bidirectional recurrent neural networks (BiDir-LSTM), so-called BiDir-LSTM-CNN. The second class is based on bidirectional encoder representation from transformers (BERT). These two architectures are representative of the two most popular, general approaches to text classification. Both architectures give good results and are actively developed by researchers today. The results in [3] show that deep neural networks outperform other traditional methods in text classification. For this reason, methods such as support vector machines, bayesian methods, and decision trees are omitted in this analysis.

The rest of the paper is organized as follows: Section 2 reviews several notable research works on fake news detection area and outlines the two neural network models investigated in this work. Section 3 describes the dataset and the methods applied. Section 4 provides the results and discussion. Finally, Section 5 concludes the paper and presents future work

II. Background

In [4] the authors compare different BiDir-LSTM models for fake news detection. The study proves the effectiveness of sequential models in the context of text classification and fake news detection, pointing towards a future comparison with BERT. The study concludes that a deeper insight into the data being used would be beneficial. In a similar application, Facebook's artificial intelligence initiative [5], utilizes BERT as a part of its machine learning approach for hate speech detection. Also in the context of social media, in [20] the authors investigate the role of bots in spreading fake news across social networks. The study suggests that during the 2016 presidential elections in the US almost 19 million bot accounts tweeted in support of either candidates. Other approaches to fake news detection include identifying user profiles that may be indicative of proliferating false information (e.g. [17]), or visual-based approaches, which attempt to recognize fake images, and images aimed to elicit strong emotional responses (e.g. [16]). In this work we are strictly concerned

978-1-7281-8470-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICMLA51294.2020.00127

with linguistic-based modalities for fake news detection, under the assumption that fake news is typically characterised by inflammatory and sensational writing styles (e.g. clickbait), that could give away their malicious intent. Moreover, the focus of this work is not on producing a new neural network model for Fake News detection, but on the analysis of the following two models, and providing a deeper degree of explainability into the process.

A. Bidirectional long-short term memory neural network with attention mechanism and convolutional layer (BiDir-LSTM-CNN)

The BiDir-LSTM-CNN architecture [6], uses the CNN layer for extraction of high-level representation from the embedding layer, while the BiDir-LSTM layer is used for extraction of past and future context. The attention layer gives focus to different outputs from the hidden layers of the BiDir-LSTM. The model has proven to be effective for the task of text classification, giving a mean accuracy of 90% on a high variety of data-sets [6]. This model has been chosen for the analysis since it is representative of the sequential approach to text processing and has proven to give similarly good results independently of the dataset used.

The network maintains two sets of hidden states. The first state is a forward state, where the network processes inputs from beginning to end. The second state is a backward state, where the input is processed back to front. Knowledge of the context from both sides of the input improves the performance of the model, when dealing with sequential data like texts [7]. Using LSTM makes it possible to avoid gradient issues in data with dependencies extended over longer sequences. While conceptually CNN is similar to BiDir-LSTM, their inner workings are very different. BiDir-LSTM processes input data in sequences from both sides, moving the relationships further to the next hidden layer. CNN, in contrast, filters out partial chunks of input projecting them onto the next layer.

The attention mechanism was first developed by Bahdanau et al. [9], and later refined by Luong et al. [10]. The main idea behind it is to establish a connection between the input layer and the output layer, by assigning additional weights (attentions) to every word that is processed by the network. Words having substantial influence on the result retain higher weight value then the words that are less important. The attention mechanism improves the processing of longer sentences, making it an additional improvement over LSTM.

The BiDir-LSTM-CNN architecture is shown in Fig. 1. First, the data-set is pre-processed, tokenized, and fed into the Embedding layer. Then, high-level feature extraction is performed by the CNN layer. After that, features are fed into the BiDir layer. Then, a forward and a backward inner layer extract context from past and future sequences and passes them further to the two attention layers. There is one attention layer for each forward and backward state. Finally, both contexts are concatenated and processed through the last feed-forward layer, outputting the classification.

B. Bidirectional Encoder Representation from Transformers (BERT)

This neural network model was proposed in [11] by Google researchers in 2019 and is based on the transformer architecture, first introduced in [12]. It is chosen for the comparison in this analysis, because of its innovative approach to text classification, which makes use of a partial transformer structure and BiDir layers. This architecture is based exclusively on an attention mechanism and a feed-forward network. The architecture of this model is presented in Figure 2. BERT differs from the BiDir-LSTM-CNN model in the way it processes data. It does not look at the input in sequences, from left to right, and from right to left. Instead, it processes the input as

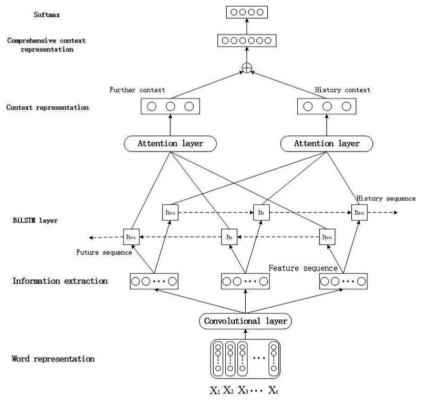

Fig. 1: Overall architecture of BiDir-LSTM-CNN [13]

a whole, like a transformer would do. After learning the context in the encoding block, outputs are fed into the BiDir layers, where they are trained according to the given classification task.

BERT is a very effective model for text classification. Unfortunately it is also very resource demanding. For this reason, in this paper, a simplified version of BERT is used called Distilled BERT. This implementation, as described in [14], achieves 98% of the accuracy of BERT, with greater speed and lighter processing. This architecture simplifies the classification layer, and optimizes dimensionality, without sacrificing the quality of the outcome.

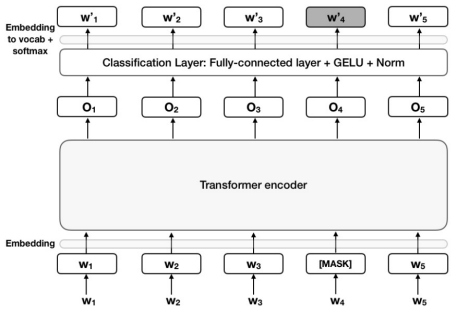

Fig. 2: BERT architecture with a Transformer encoding block for context processing and RNN layers for classification [11].

III. METHOD

A. Dataset

The data-sets used in this analysis is the "Fake News Corpus" dataset [1]. This dataset is composed of over nine million texts labeled as one of the 11 categories: false, satire, bias, conspiracy, state, junksci, hate, click bait, unreliable, political, true. Additionally, this set is categorized by topic using the Term Association Technique [15]. For the comparison in this paper, two smaller sets are derived from the "Fake News Corpus", categorized as "sports" or "tech". For the purpose of our experiments, each article from the two abovementioned classes was annotated as "fake" or "real". A detailed outline of all sets used is presented in Table I. The "tech" and "sports" datasets consist of 1000 randomly selected articles respectively. The

[1]"Fake News Corpus Data-set", *github.com* [Internet]. 2019. [cited 2020 April 7] Available from:
https://github.com/several27/FakeNewsCorpus

TABLE I: A summary of all of the data-sets used in the analysis.

	Data-set	Size	Summarized	Stemmed	Lemmatized
1	”sports”	1000	-	-	-
2	”sports”	1000	-	Yes	-
3	”sports”	1000	-	-	Yes
4	”sports”	1000	max 5 sentences	-	-
5	”sports”	1000	max 5 sentences	Yes	-
6	”sports”	1000	max 5 sentences	-	Yes
7	”sports”	1000	max 10 sentences	-	-
8	”sports”	1000	max 10 sentences	Yes	-
9	”sports”	1000	max 10 sentences	-	Yes
10	”tech”	1000	-	-	-
11	”tech”	1000	-	Yes	-
12	”tech”	1000	-	-	Yes
13	”tech”	1000	max 5 sentences	-	-
14	”tech”	1000	max 5 sentences	Yes	-
15	”tech”	1000	max 5 sentences	-	Yes
16	”tech”	1000	max 10 sentences	-	-
17	”tech”	1000	max 10 sentences	Yes	-
18	”tech”	1000	max 10 sentences	-	Yes

composition of each set is balanced between 500 articles labeled as ”fake”, and an additional 500 labeled as ”real”.

B. Pre-processing

To deepen the analysis, 16 additional sets are derived from the initial two, resulting in a total of 18 balanced sets each containing 1000 articles. There are 9 sets produced, for each category (i.e. ”sports” and ”tech”). Both categories contain articles subjected to several pre-processing operations. In each category there is a ”vanilla” set of articles (i.e. no pre-processing), a set of articles summarized to a maximum of 5 sentences each, and a set of articles summarized to a maximum of 10 sentences each. Summarization is based on the work of H. P. Luhn et al. [18], where texts are summarized taking into consideration word frequencies and stop-words removal. In addition, lemmatization and stemming are also used as a pre-processing step, as depicted in Table I.

Summarization: Text summarization is utilized in many neural network models used for text classification. The method used in this work is based on the Term Frequency-Inverse Document Frequency (TF-IDF) method described in [19]. It produces a summary accuracy of 68%, which is better than other similar summarization methods available. Summary accuracy is a measure of conveyance of meaning. It describes to what extent the meaning of a summarized text generated by the model resembles the meaning of a summarized text created by a human. The technique is based on a process of producing summaries, by extracting sentences from a text, as they are in the original texts, preserving the order of appearance.

Stemming: This data pre-processing technique involves reducing words to their root form, mapping them to the same stem. Stems don’t necessarily have to be valid words by themselves. Stemming is a suffix stripping technique, removing suffixes or prefixes such as: ”-ed”, ”-ize”, ”-s”, ”-de”, ”mis-” from a word. The Porter Stemmer, as it is called, reduces words like: ”cats” to ”cat”, and words like ”trouble”, ”troubling”, ”troubled”, to ”trouble”. This process positively contributes to results in many different supervised architectures, as presented in [21]. In this analysis, the Python NLTK package[2] is used to stem our dataset.

Lemmatization: This type of data pre-processing also uses the Pythons NLTK package. Lemmatization, unlike stemming, reduces words to their proper dictionary root form. While words like: ”cats” are still reduced to the root form ”cat”, words like ”misstated” on the other hand, are not. Lemmatization is often used as an alternative to stemming if dictionary vocabulary is to be preserved. As shown in [22], models using texts that undergo these pre-processing operation can still achieve an acceptable result. Stemming technique is quicker than Lemmatization, but texts can end up having nonsensical, non-dictionary words.

After creating the datasets (see Table I), two models, the BiDir-LSTM-CNN, and BERT are built in accordance to their architectures described in section II. Each data-set is used in the training phase of both models. Following, a prediction is obtained with regards to the ”fake” or ”real” nature of the text, and the attention weights are extracted. To aid the analysis, attention weights are presented visually as color-coded texts, representing the interest given by the network for each processed word. Lastly, statistical data is computed for the analysis. Statistics, representing attention distribution among words in all of the texts, give a deeper insight into the classification process.

Visualizations are generated with a color-coded scheme and should be understood as follows. A background color is incorporated for every word in the text. Colorations represent the level of attention a model pays to a particular word when classifying a text. Words marked as red should be understood as not having a substantial influence in the model’s classification. In contrast, words marked as green should be recognized as having a strong effect on the outcome of the text’s classification. Yellow words should be interpreted as those with a moderate influence in the classification. In addition to the three principal colors used as background, a color strength, or alpha, is used to further define the levels of attention. Words with stronger colors, the ones with higher alpha value for any of the 3 principle colors, have a higher influence on the classification, than words with the same color and weaker alpha value. For BERT, a yellow background color is omitted to increase the readability of the visualization, given the manner in which the model distributes attention. This difference in visualizations has no effect on the results and analysis. An entire lack of coloration indicates a virtually complete absence of attention for a particular word.

IV. Results

Models are built using Tensorflow 2.0[3] library in Python. For BERT implementation, the additional packages Huggingface[4] and KTrain[5] have been used as well. After training both models, and extracting the attention weights, visualizations of data and statistics have been generated. While visualizations are constructed for a larger number of texts, only a particular 2 examples are showcased here. Texts presented exemplify typical results achieved for the majority of the input data, for both contextual domains (”sports”, ”tech”), over the datasets under consideration. Visualizations exhibit the same texts,

[2]”Natural Language Toolkit” *nltk.org* [Internet] [cited 2020] Available from: https://www.nltk.org/

[3]https://www.tensorflow.org/

[4]”State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch”, *gihub.com* [Internet] [cited 2020] Available from: https://github.com/huggingface/transformers

[5]”Github repository: Ktrain.”, *github.com* Available from: https://github.com/amaiya/ktrain/blob/master/ktrain/text/preprocessor.py

subjected to the aforementioned pre-processing operations in relation to the two DL models. For illustration purposes, for both the "sports" and "tech" domains, representative texts are chosen for visualizing the results. Models are expected to learn to classify texts as "fake" or "real" with a specific domain in mind. By leaning on domain-specific data-sets, models can achieve a better understanding of the importance of individual words, for that specific domain.

A. Visualizations

Results for the BiDir-LSTM-CNN model are color-coded with the red-yellow-green scheme. Colors correspond to the level of attention to words. In BERT, the yellow background color is not present. In addition, the lack of color represents the model's absence of interest in a particular word. For BiDir-LSTM-CNN, in order to signal different attention strengths for each specific color, attention thresholds are divided into 1/3 of the maximum attention weight for each attention type. For BERT the division is 1/2 for each attention type. Visualizations in Figures 3-6 show differences in attention depending on the texts' lengths, word diversity, type of pre-processing, and contextual domain. As can be noted, the BiDir-LSTM-CNN model has the tendency to either allocate a high level of attention to the majority of words in the text, or to just a few (see Fig. 3 and Fig. 5). Alternatively, BERT keeps a balanced proportion of strong, weak and no attention, regardless of the pre-processing operation performed on the text or its domain (see Fig. 4 and Fig. 6). Results also show, how the DL model shifts focus to different words when the same text is structured differently. Models trained on the original texts, present different results than the same models trained on the same texts summarized to 5 and 10 sentences. Similar observations can be made about texts that are lemmatized and stemmed. It can be observed that the influence that text formatting has on the BiDir-LSTM-CNN model's attention differs substantially from that of BERT's.

Visualizations organization:

1) Original text.
2) Original text - lemmatized.
3) Original text - stemmed.
4) Text summarized max 5 sentences.
5) Text summarized max 5 sentences - lemmatized.
6) Text summarized max 5 sentences - stemmed.
7) Text summarized max 10 sentences.
8) Text summarized max 10 sentences - lemmatized.
9) Text summarized max 10 sentences - stemmed.

B. Model accuracy

The accuracy of a model is a measure of the number of correctly classified texts in a data-set. For BiDir-LSTM-CNN, the highest accuracy is 85% and for BERT is 52%, for original texts. All accuracy values for both models are presented in Table II for all dataset types. For the "sports" domain, a noticeable advantage is demonstrated by BiDir-LSTM-CNN, on the original text, and original lemmatized text, with values 85% and 66% accuracy. BERT, despite consistently low values, demonstrates slightly higher values for the original text, original text stemmed, and text summarized to 5 sentences. For "tech", both models show higher performance on the original text, and text summarized to 5 sentences. The highest value for BIDir-LSTM-CNN in "tech" is achieved in lemmatized text summarized to 5 sentences. For BERT the highest value in "tech" is achieved in text summarized to 5 sentences. As can be concluded by analyzing the texts, the summarization procedure tends to overlook key messages conveyed

Fig. 3: BiDir-LSTM-CNN "sports" - Visual overview of results.

Fig. 4: BERT "sports" - Visual overview of results.

in the text, which can explain the subsequent lower accuracy for the classification. Similarly, this can be the cause for the shift in attention focus of the model once the text is summarized.

C. Attention coverage

Attention coverage represents the percentual amount of text, subjected to any of the attention types. BERT's results show a more selective approach to assigning attention to words, while BiDir-LSTM-CNN shows a broader attention coverage. As presented in Table III, BiDir-LSTM-CNN has approximately double the coverage

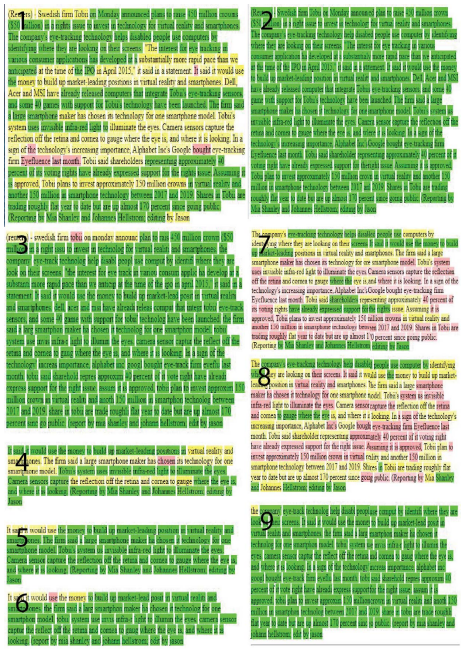

Fig. 5: BIDir-LSTM-CNN "tech" - Visual overview of results.

Fig. 6: BERT "tech" - Visual overview of results.

of strong attention than BERT. For moderate attention in BiDir-LSTM-CNN and weak attention in BERT, it is BERT that has double the attention coverage in comparison to BiDir-LSTM-CNN. The weak attention coverage for BiDir-LSTM-CNN and no attention for BERT, appear to be more similar.

D. Attention span over successive words

Attention span represents the number of adjacent words, covered with the same attention type. For BiDir-LSTM-CNN, on the original

TABLE II: BiDir-LSTM-CNN and BERT - Accuracy for all texts.

	BiDir-LSTM-CNN "sports"	BERT "sports"	BiDir-LSTM-CNN "tech"	BERT "tech"
Original text	85%	52%	52%	51%
Original text lemm.	66%	50%	50%	50%
Original text stemm.	51%	51%	51%	50%
Text max 5 sen.	50%	51%	53%	52%
Text max 5 sen. lemm.	50%	50%	56%	51%
Text max 5 sen. stemm.	50%	50%	50%	50%
Text max 10 sen.	51%	50%	50%	51%
Text max 10 sen. lemm.	50%	50%	50%	50%
Text max 10 sen. stemm.	55%	50%	53%	50%

TABLE III: BiDir-LSTM-CNN and BERT - Overall attention coverage of texts.

	BiDir-LSTM-CNN			BERT		
"sports"	Strong	Mod.	Weak	Strong	Weak	None
Original texts	64%	9%	27%	36%	40%	24%
Max 5 sen. texts	15%	37%	48%	50%	33%	17%
Max 10 sen. texts	43%	25%	32%	40%	51%	9%
Total	59%	13%	28%	37%	41%	22%
"tech"	Strong	Mod.	Weak	Strong	Weak	None
Original texts	79%	20%	1%	37%	46%	17%
Max 5 sen. texts	83%	16%	1%	43%	48%	9%
Max 10 sen. texts	45%	26%	29%	36%	52%	12%
Total	68%	21%	11%	38%	48%	14%

texts, attention tends to stretching through a higher number of words than for BERT. As presented in Table IV, an average of 39 and 129 successive words are covered by strong attention in BiDir-LSTM-CNN for the two domains respectively. BERT on the other hand, has an average attention span of 2 words regardless of the pre-processing operation applied. The median of attention span is also provided since average values are exaggerated due to the BiDir-LSTM-CNN's overblown attention span in some of the texts. Nevertheless, as can be seen in Table IV, the difference in attention span indicates BiDir-LSTM-CNN's broader contextual consideration of texts than BERT. The result across all of the text groups shows a median of 15 and 9 words span in strong attention for BiDir-LSTM-CNN in both domains respectively. Meanwhile, BERT holds a 1 or 2 words attention span across all of the texts.

E. Domain specific words

In Table V we provide an indication on how attention is distributed over domain-specific words. That is, the share of domain specific

TABLE IV: BiDir-LSTM-CNN and BERT average and median word attention span in texts.

BiDir-LSTM-CNN	Average			Median		
"sports"	Strong	Mod.	Weak	Strong	Mod.	Weak
Original texts	39	5	33	18	4	38
Max 5 sen. texts	3	5	13	3	3	17
Max 10 sen. texts	20	5	9	7	2	4
Total	31	5	21	15	4	17
"tech"						
Original texts	129	5	1	18	4	1
Max 5 sen. texts	37	6	1	20	5	1
Max 10 sen. texts	17	4	6	4	3	4
Total	49	5	5	9	3	3

BERT	Average			Median		
"sports"	Strong	Weak	None	Strong	Weak	None
Original texts	2	2	1	2	2	1
Max 5 sen. texts	2	1	1	1	1	1
Max 10 sen. texts	2	2	1	2	2	1
Total	2	2	1	2	2	1
"tech"						
Original texts	2	2	1	2	2	1
Max 5 sen. texts	2	2	1	2	2	1
Max 10 sen. texts	2	3	1	1	2	1

TABLE V: Percentual share of domain specific words in attentions.

BiDir-LSTM-CNN	Strong	Moderate	Weak
Average "sports"	6.5%	0%	0%
Average "tech"	3%	1%	1%

BERT	Strong	Weak	None
Average "sports"	4%	6%	4%
Average "tech"	2%	3%	2%

words out of all the words in the text that the network pays attention to. The contextual grouping of words for the two domains was done with the help of keyword association. Only nouns have been used as domain-specific keywords. For BiDir-LSTM-CNN strong attention is covering 6.5% of words for the "sports" domain and 3% for "tech". BERT seems to follow a similar pattern, with slightly lower values. It is worth mentioning that the data is consistent for both models with very few exceptions. It can be noted that in the case of BiDir-LSTM-CNN the values for moderate and weak attention is very low, either 1% or 0%, while BERT maintains approximately the same values for weak and no attention as in the case of strong attention.

TABLE VI: Cumulative percentual average share of Nouns and Verbs in a given attention type.

BiDir-LSTM-CNN	Strong	Moderate	Weak
"sports"	32%	49%	34%
"tech"	40%	38%	52%

BERT	Strong	Weak	None.
"sports"	45%	49%	49%
"tech"	57%	57%	81%

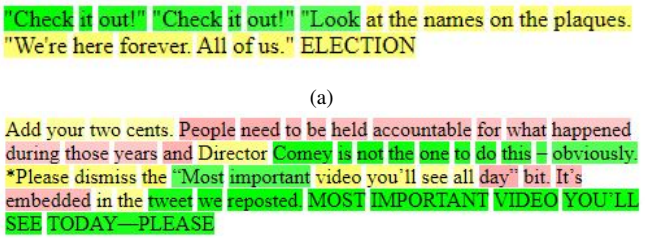

Fig. 7: BiDir-LSTM-CNN shows strong attention to *clickbait* phrasing when classifying *fake news*.

F. Parts of speech

Not only words from a specific domain, but also different parts of speech influence the learning and classification process. Both models show a significant importance allocated to nouns, followed by verbs as the parts of speech with the highest attention. With a couple of exceptions, these two groups of words seem to have the foremost influence on both models classification of texts. In every attention group, for both models nouns and verbs are the dominant part of speech taken into consideration. In most cases, for both models, the percentage of nouns and verbs in all attention types is between 30%-60%. Values can be seen in Table VI. Noticeably, the percentages for BERT are higher, which can be explained by the more selective manner in which the model allocates attention, in comparison to BiDir-LSTM-CNN, which is shown to apply attention over extended sequences, largely disregarding the type of words.

G. Further insights

An interesting finding, points out the correlation between *clikbait* type of content and fake news. Clickbait content can be regarded as a semantic class, characterized by a propensity for exaggeration, sensationalization and eye-catching phrasing. Importantly, both DL classifiers show good results in terms of identifying linguistic patterns which are indicative of clickbait wording. This result is consistent across both of the two domains included in our evaluation. In Figure 7 we depict several examples using the better performing BiDir-LSTM-CNN model, which emphasize that the model can correctly pick-up clickbait language when this is present in the text, and assigns to it high attention values during the classification process. Although it is not necessarily the case that all fake news contain clickbait phrasing or headlines, in practice, this can often serve as a strong indicator. In fact, previous studies also report an existing underlying correspondence between clickbait news, which generaly does not contain well-researched information, and fake news [23].

V. Conclusion and future work

Although it plays an important part, the architecture is not a definitive reason why a natural language processing model is successful or not. For it to be successful, several additional aspects have to be taken into consideration. The length of the text, its complexity, and pre-processing operations, have a big impact on the training process. To this end, in this paper we aim to provide a level of explainability of deep learning models for the task of fake news detection.

Two architectures have been analyzed in this work. First, BiDir-LSTM-CNN that looks at the data sequentially word by word, from left to right, and from right to left. Second, BERT, which finds dependencies between words across all of the text. Both models have proven to be successful in different degrees when classifying texts in the direction of fake news detection. The outcome of the analysis showed how different is the approach taken by each of these two models to classifying texts as "fake" or "real". BiDir-LSTM-CNN takes larger chunks of text under consideration or ignores them entirely. BERT is more selective in distributing attention focusing on specific words. There is also a considerable difference when it comes to shorter texts or text pre-processed in some specific manner.

On the one hand, in the case of BiDir-LSTM-CNN when text is too long and the model cannot handle it, it tries to overcompensate by paying attention to the whole text. In this situation, pre-processing, although less intelligible from a human perspective, can help reduce text complexity and ease the model's ability to learn efficiently. On the other hand, pre-processing texts does not necessarily mean that a model is going to achieve improved accuracy, especially in situations where the message conveyed by the text is significantly altered during this operation. Importantly, our results show that pre-processing largely decreases the performance of both models, with variations depending on the specific domain. Another aspect is the fact that nouns and verbs are primary candidates to influence the classification. This remark suggests that the best-suited texts for classification, are those rich in words belonging to these types of speech groups. Overall, BiDir-LSTM-CNN proves to be significantly superior in its accuracy in comparison to BERT, despite the apparent more selective manner of considering important words displayed by the latter. Finally, an insight which remains consistent across both models, regardless of the domain under investigations, is the strong correlation between clickbait content and fake news.

In future work, we plan to extend the analysis to include broader datasets with an increased number of domains. In addition, we aim to investigate different transfer learning mechanisms in order to determine the extent to which commonalities in fake news linguistic patterns are distributed across different domains. Moreover, we plan to extent this work by investigating different approaches to fake news detection that go beyond linguistic-based approaches.

Acknowledgements

This research was partially funded by Crafoord project grant number 20200953.

References

[1] "Individuals using the internet 2005-2019." *itu.int* [Cited 2020 May 14] Available from: https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

[2] "How often you encounter fake news? Opinions in Europe 2018, by country.", *statista.com* [Cited 2020 May 14] Available from: https://www.statista.com/statistics/1076701/fake-news-frequency-europe/

[3] D. Katsaros, G. Stavropoulos, D. Papakostats. *Which machine learning paradigm for fake news detection?* 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pages 382-387, Thessaloniki, Greece, Greece, 14-17 Oct. 2019, IEEE.

[4] Abadella A, Al-Sadi A, Abdullah M. "A Closer look at Fake News Detection: A Deep Learning perspective" *ICAAI* 2019.

[5] "AI advances to better detect hate speech.", *ai.facebook.com* Available from: https://ai.facebook.com/blog/ai-advances-to-better ct-hate-speech

[6] Liu G, Guo J. "Biderectional LSTM with attention mechanism and convolutional layer." *Neurocomputing 337.* 2019; 325-338

[7] Charu C Aggarwal. *Neural Networks and Deep Learning.* 1 ed. Springer; 2018. 315-371.

[8] Kamath U, Liu J, Whitaker J. *Deep Learning for NLP and Speech Recognition.* 1 ed. Springer; 2018. 263-314.

[9] Bahdanau D, Cho K, H, Bengio J. "Neural Machine Translation by Jointly Learning to Align and Translate." *ICLR.* 2015

[10] Luong M, Pham H, Manning C. "Effective Aproaches to Attention-based Neural Machine Translation." *Proceedings of the 2015 Conference on Empirical Methods in Neural LanguageProcessing.* September 2015; 1412-1421

[11] Devlin J, Chang M, W, Lee K, Toutanova K. "BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding." *Cornell University*, arXiv:1810.04805v2. May 2019

[12] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, N, Kaizer L, Polosukhin L. "Attention is All You Need." *Cornell University.* arXiv:1706.03762v5 [cs.CL]. 6 Dec 2017

[13] Liu G, Guo J. "Biderectional LSTM with attention mechanism and convolutional layer." *Neurocomputing 337.* 2019; 325-338. Figure 3, The architecture of the AC-BiLSTM; p. 330

[14] Sanh V, Debut L, Chaumond J, Wolf T. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *Cornell University.* arXiv:1910.01108v4 [cs.CL]. March 2020

[15] Antonie Maria-Luiza, Zaiane O. R. "Text Document Categorization by Term Association". Data Mining 2002. *ICDM Proceedings IEEE.* 2002.

[16] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia*, 19(3):598–608, 2017.

[17] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. *In Proceedings of the 20th international conference on World wide web (WWW '11)*. Association for Computing Machinery, New York, NY, USA, 675–684.

[18] Luhn H.P. "The Automatic Creation of Literature Abstracts." *IBM Journal.* April 1958; 159-165

[19] Hans C, Pramodana Agus M, Suhartono D. "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)." *ComTech* Vol. 7 No. 4 December 2016. 285-294

[20] Alessandro Bessi and Emilio Ferrara. Social bots dis- tort the 2016 us presidential election online discussion. First Monday, 21(11), 2016.

[21] Altunbey Ozbay F. Alats B. "Fake News detection with online social media using supervised artificial intelligence algorithms." *Physica* A 540. 2020.

[22] Agarwal V, Parveen Sultana H, Malhotra S, Sarkat A. "Analysis of Classifiers for Fake News Detection." *International Conference on Recent Trends in Advanced Computing, ICRTAC.* 2019

[23] Prakhar Biyani, Kostas Tsioutsiouliklis, and John Blackmer. "8 amazing secrets for getting more clicks": detecting clickbaits in news streams using article informality. *In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16). AAAI Press,* p. 94–100, 2016