International Journal of Information Management Data Insights 2 (2022) 100064

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

Detection of hyperpartisan news articles using natural language processing technique

Navakanth Reddy Naredla[a], Festus Fatai Adedoyin[b,*]

[a] Department of Computing and Informatics, Bournemouth University, Bournemouth, UK
[b] Department of Computing and Informatics, Bournemouth University, UK

A R T I C L E I N F O

Keywords:
BERT
Transformers
Word embedding's
ELMo
NLP
Word2vec
Tensorflow
Bidirectional

A B S T R A C T

Yellow journalism has increased the spread of hyperpartisan news on the internet. It is very difficult for online news article readers to distinguish hyperpartisan news articles from mainstream news articles. There is a need for an automated model that can detect hyperpartisan news on the internet and tag them as hyperpartisan so that it is very easy for readers to avoid that news. A hyperpartisan news detection article was developed by using three different natural language processing techniques named BERT, ELMo, and Word2vec. This research used the bi-article dataset published at SEMEVAL-2019. The ELMo word embeddings which are trained on a Random forest classifier has got an accuracy of 0.88, which is much better than other state of art models. The BERT and Word2vec models have got the same accuracy of 0.83. This research tried different sentence input lengths to BERT and proved that BERT can extract context from local words. Evidenced from the described ML models, this study will assist the governments, news' readers, and other political stakeholders to detect any hyperpartisan news, and also helps policy to track, and regulate, misinformation about the political parties and their leaders.

1. Introduction

Hyperpartisan news is an extreme right or left biased version of particular political news, which is intended to favor or defame a politician. There is a rapid growth in yellow journalism in the era of social media. The three main threats on social media nowadays are fake news, clickbait, and hyperpartisan news. Fake news is unreal news. There was much fake news spread about Covid-19 throughout the pandemic which caused unnecessary fear among the people. Unlike fake news, Click baits are very short news or URL links that are intended to grab the attention of internet users to make them go through a particular website or news. These click baits are less harmful compared to fake news, as these are mainly focused on the entertainment sector. Hyperpartisan news is political news that is extremely left or right biased. This hyperpartisan news is often used in times of elections to favor or defame a particular politician. There were so many speculations spread around Brexit (Bastos & Mercea, 2019) and the US president election which spread wrong perceptions among people.

There is a rapid growth in the usage of social media platforms in rural areas as well. This makes it easy for the yellow journalist to spread false news through social media because people in rural areas are unaware of knowing the ground truth. The spread of hyperpartisan news misshapes public opinion and also spoils the credibility of journalism. Nowadays, news that is more hyper partitioned is spreading more quickly than genuine news. So, there is a need for an automated system that can detect this hyperpartisan news on the internet. The news that is detected as fake will be marked as hyperpartisan on the internet, which is very helpful for readers to avoid reading that news.

The evolution of Artificial intelligence (AI) over the years have made researchers use it in HCI (Human-computer interface) systems. Natural language processing (NLP) is a sub-stream of Artificial intelligence. These language processing models will perform tasks like classification (Pal & Patel, 2020), question and answer, fill in the blanks, and weather forecast (Fan et al., 2018) etc. BERT (Devlin, Chang, Lee & Toutanova, 2019), ELMo (Huang & Lee, 2019), Word2vec (Ji, Satish, Li & Dubey, 2019) are emerging neural networks in the field of natural language processing. These neural networks are very efficient in contextual text processing. Before the introduction of these deep neural networks machine learning algorithms (Pal & Patel, 2020) like Naïve Bayes, Logistic regression, Random forest classifier etc. to perform language processing. These machine learning algorithms are not capable of extracting the syntactic and semantic meaning of words used in the text. These machine learning models can be used along with neural networks to perform language processing more accurately.

This research makes use of the above stated natural language processing models to detect hyperpartisan news articles. Deep neural networks will be used in conjunction with machine learning algorithms to perform the prediction (binary classification). The model will predict

* Corresponding author.
E-mail addresses: s5229355@bournemouth.ac.uk (N.R. Naredla), fadedoyin@bournemouth.ac.uk (F.F. Adedoyin).

https://doi.org/10.1016/j.jjimei.2022.100064
Received 1 October 2021; Received in revised form 22 February 2022; Accepted 25 February 2022

whether a news article is hyperpartisan or not. Thus, this research aims to build a natural language processing model that can be used to detect hyperpartisan news or articles found on the internet. The research aims to include more context while processing text. There are several natural language processing models that can perform contextual language processing. These processes will make use of language processers such as BERT, ELMo, and Word2vec to detect hyperpartisan news articles. As ELMo and Word2vec are not prediction models, the Random Forest classifier can be used to make predictions.

In the present world of digitalization, there is a rapid growth in the number of online news publishers, who are taking sides of political parties and spreading biased news to support them. It is very difficult for a reader to get to know whether the news article he is reading is biased or not, so there is a need for an automated model to detect such type of biased one-sided political news. Also, the rise of social media enabled individuals to disseminate information at little expense, with minimal examination and fewer restrictions on what is being disseminated. This exacerbated the longstanding problem of false news, which has been a big worry in recent years owing to the bad impact it has on the communities and countries. To combat the emergence and transmission of false news, researchers have developed automatic detection algorithms based on machine learning and artificial intelligence. Deep learning methods' recent breakthroughs in complicated natural language processing tasks make them a possible answer for false news identification as well. Hence, this research aims to support the previous studies in a bid to an automated model that can be used to detect hyperpartisan news without human intervention.

2. Literature review

2.1. Sources and effects of hyperpartisan news

The main source of hyperpartisan news is online blog writers. Those who are adopted to yellow journalism are spreading biased or fake news to earn money. These articles are mainly intended to favor or defame a particular person.

Nowadays, Bots are more often used to spread hyperpartisan news. Bots (Dukic, Keca & Stipic, 2020) are intelligent systems or software's that can interact with people on the internet. The research of Bastos and Mercea (2019) gives us clear evidence of how efficiently Twitter bots were used to spread hyperpartisan news during Brexit and the US presidential election. Around 10 million tweets are from 39 hashtags which are about the Brexit referendum were collected by Bastos and Mercea (2019), and used filtering and thresholding techniques to identify bots from real persons. This research found that 40% of the accounts out of 794,949 accounts has deactivated or changed profiles, which means all these 40% of accounts used were bots to spread fake news. But there was no proper evidence of spreading fake news by using political bots.

The research of was about how fake news influenced voters during the US presidential election in 2016. Out of 171 M tweets that are tweeted during the election, 25% of the tweets are fake which were intended to influence voters. The effects of Russian Twitter trolls on the US presidential election was studied by Badawy, Ferrara and Lerman (2018). Around 5.7 million tweets from Russian accounts that were tweeted during the US election campaign 2016 were collected by Ferrara and Lerman (2018) and used data analytic algorithms to find the influence of tweets on voters and also to find how many of these Twitter accounts are bots. Badawy, Ferrara and Lerman also found that around 4–6% of Twitter accounts were handled by automatic bots to propagate biased news during the election campaign.

2.2. Features of hyperpartisan news

The structural, linguistic, and temporal characteristics of the text to identify the text features of hyperpartisan news that are different from real news were examined by Kwon, Cha, Jung, Chen and Wang (2013). According to Kwon et al. (2013), titles are the more prominent factor that differs between biased and real news, biased news will use more attractive titles. The use of more proper nouns and repetitive content in the body of the article is also the salient features that differentiate hyperpartisan news from the real news. Also found is that biased news draws attention through heuristics, while real news is through arguments.

The importance of contextual and linguistic features of the text in the identification of misinformation was studied by George, Skariah and Xavier (2020). George, Skariah and Aleena Xavier used several machine learning algorithms like Naïve Bayes, random forest, and KNN; and neural networks like hybrid CNN and LSTM's on different datasets like Facebook, Twitter and LIAR datasets. Machine learning algorithms can process only linguistic characteristics of text, so hybrid CNN and LSTM's were used to analyze the contextual features of the text. The poor performance of hybrid CNN with 27% accuracy on the LIAR dataset proves that the sequential order of contextual features affects the accuracy of the model.

According to Hlaing and Kham (2020), news content only is insufficient to define the authenticity of the news, social context of news is also required. Misinformation detection based on social context and news content on a multidimensional dataset by using a new approach of synonym based feature extraction and classification was done by Hlaing and Kham (2020). Decision trees, a random forest classifier, and a boosted classifier were used to classify the news articles. The use of social context along with the news content has boosted the performance of all three classifiers. Contextual clues applied to news headlines are not always helpful for readers in finding misinformation, further in some cases they might mislead readers (Caramancion, 2020). The source and background of the news also mislead users in detecting misinformation. Caramancion also proved that fake news will be aided by the use of fake videos or images in it.

2.3. Hyperpartisan, clickbait, and fake news detection using natural language processing techniques

Hyperpatisan news, fake news, and clickbait are related to each other. Fake news is unreal news, whereas hyperpartisan news is biased political news. Clickbait is short news or hyperlinks that are intended to attract the attention of internet users. The following section focuses on the literature related to addressing these three problems.

2.3.1. Fake news detection using NLP techniques

An automated model called SPOTFAKE was developed by Giachanou, Zhang and Rosso (2020) to detect fake news spreading across social media. Twitter and Weibo datasets were used to train and validate the model. Accuracy, precision, recall and F1-score were used as evaluation metrics. BERT was used to exploit the textual feature of data, whereas VGG- 19 was used to exploit the image features present in the data. This research proved that image features used along with textual features improved the accuracy of SPOTFAKE by 6% compared to other models that only used textual features for classification. This SPOTFAKE model still needs improvement to efficiently work with longer length articles.

Two different approaches were used by Madhusudhan, Mahurkar and Nagarajan (2020) to build a multimodal fake news detection system. In the first approach, BERT and SBERT were trained with textual features and ResNet-18 was trained with image features. Visual Attention for fake news detection was used as the second approach. It was found that BERT extracted more context than SBERT and this multimodal system was very effective in capturing frequency-based relations.

A similar model like SPOTFAKE (Giachanou et al., 2020) was developed. This model used both textual and visual information to detect fake news like SPOTFAKE (Giachanou et al., 2020). The textual features were trained on BERT, and the visual features were trained on VGG-16 instead of VGG-19 that was used in SPOTFAKE. A baseline model was developed by training only textual features on BERT and the second model used BERT to train textual features and vgg-16 to train visual features. This multi model fake news detection system outperformed the baseline BERT that was trained with only textual features by 4.19% in terms of accuracy proving that the visual information present in the news adds extra advantage while making prediction. This multi modal fake news detection system also outperformed the state of art SPOTFAKE model by 5.39% in terms of accuracy.

SPOTFAKE (Giachanou et al., 2020) and the multi-modal fake news detection system by Shivangi Singhal, Shah, Chakraborty, Kumaraguru and Satoh (2019) showed that the textual features along with the visual information in the news were used to train deep neural networks to improve the accuracy of the models.

A dataset consisting of 200 tweets on Hillary Clinton was prepared by Dey, Rafi, Hasan Parash, Arko and Chakrabarty (2019), and an analysis of linguistic features of news was done to find fake news spread about the US presidential election 2016. Features in the news were extracted by extracting bag-of-words, and a further k-nearest neighbor algorithm was applied for the classification of features.

2.3.2. Automatic clickbait detection

Clickbait and hyperpartisan news are less explored compared to fake news. Clickbait is a short attractive headline or hyperlink intended to grab the attention of the reader to click on it. Machining learning algorithms were used by Genc et al. (2019) to detect clickbait. This research used the dataset consisting of twitter headlines in the Turkish language. The accuracy of the model was around 0.91 and the F1-score is also 0.91. Machine learning models treat words as bag-of-words and does not take into account, the context of words while processing, thereby causing lower accuracy of machine learning models in predictions.

To improve the accuracy of the model built by Genc et al. (2019), which used machine learning algorithms for prediction, built a model by using deep neural networks, that can detect clickbait in the Thai language. This model used three different neural networks called Feed-forward neural network, long short-term memory (LSTM), and gate recurring unit was trained with two types of word embeddings i.e., one is character level embedding and the second one is word-level embedding. Accuracy, recall, precision, F1-score, and ROC-AUC were used as evaluation metrics with 10-fold cross-validation. BiLSTM with word-level embedding performs very well achieving accuracy rate of 0.98, f1-score of 0.98. BiLSTM with word embedding got the best result compared to other model architectures. This research also found that deep neural networks can detect clickbait effectively without knowing the context of words in the language.

RCNN (Recurrent Convolutional Neural Networks) was used by Chawda, Patil, Singh and Save (2019) to predict clickbait. LSTM (long short-term memory) and GRU (Gated Recurrent Unit) were also used along RCNN. The model with RCNN+LSTM+GRU gave almost 4% extra accuracy when compared with the standard SVM (Support Vector Machine) classifier. This research used only the content of titles while making a prediction. In future, there is a scope of including article body as well for making predictions.

2.3.3. Hyperpartisan news detection

In SEMEVAL-2019, Palic et al. (2019) presents several approaches used to perform hyperpartisan news articles. This research used approaches like deep learning Giachanou et al. (2020), machine learning (Palic et al., 2019), ELMo (Jiang, Petrak, Song, Bontcheva & Maynard, 2019), and BERT case (Huang & Lee, 2019) to detect hyperpartisan news by using by-article and by-publisher datasets. This research proved that deep learning models are more accurate in the prediction of hyperpartisan news and failed to utilize BERT more efficiently.

An automated model using a Support Vector Machine (SVM) and TF-IDF tokens for detecting hyperpartisan news was developed by Alabddulkarim and Alhindi (2019). Textual features like word count, paragraph count, and hyperlink count of 645 hyperpartisan and mainstream articles were used to train SVM. This methodology used SVD to do dimension reduction and SVM with default settings for doing prediction. The accuracy (0.742) of the model is very low as SVM failed to grab the contextual meaning of words. The dataset that was used is very small and very much imbalanced because the number of mainstream articles is far greater than the number of hyperpartisan news articles.

BERT can take a maximum of 512 tokens or words of a sentence at a time as its input Drissi, Sandoval, Ojha and Medero (2019). varied the input sentence length of articles imputed into BERT to find the optimum sentence length that yields greater accuracy. BERT-CASE and BERT-LARGE were trained with 80% of the bi-article Palic et al. (2019) dataset and the remaining 20% was used to test the model while the F-1 score was used as an evaluation metric. The outcome of the research proved that the accuracy of the BERT model depends on the imputed word count of a sentence. It also showed that BERT does not need the whole sentence as input, sometimes 100 words give better accuracy over 500 words of a sentence. That means BERT is getting context from local words.

According to Celena, Park, Dwyer and Medero (2019), the accuracy of any bag-of-words model can be increased by using features of the article's titles and sentiment analysis Celena et al. (2019). used spaCy to tokenize the text and to sentence segmentation. A python standard library called TextBlob worked very efficiently in sentiment analysis of articles and their titles. Feature section along with optimized classifier would give more significant insight into the classification of hyperpartisan news articles. The naïve bayers classifier trained with the tokens extracted by spaCy performed very poor when compared to baseline models developed by Drissi et al. (2019) and Alabdulkarim and Alhindi (2019). The same bag-of-words approach (Celena et al., 2019) was used by Bestgen (2019). Uni-gram tokens to train the logistic regression classifier was used by Bestgen (2019), whereas Celena et al. (2019) used bi-gram tokens to train the naïve bayers classifier. The use of uni-gram tokens improved the accuracy of the model over bi-gram tokens.

Word embedding's trained on deep neural networks will perform more contextually when compared to the bag of words approach followed by Celena et al. (2019) and Bestgen (2019). ELMo was used to generate the word embeddings for the text and these word embeddings were used to train a deep contextualized neural network to make predictions. This approach worked very well on the bi-article (Palic et al., 2019) dataset with an accuracy of 80% but failed to make the predictions accurately with by-publisher dataset. The longer length of articles and the unavailability of the source of articles in the by-publisher dataset might be the reason for the poor performance of the ELMo model to detect hyperpartisan news.

Rather than using tokens extracted from articles and titles of the articles for the classification of biased news, Kwon et al. (2013) focused on the temporal and textual features of articles for predicting biased news. In this approach, first feature selection was done and, secondly, SVM was trained with the selected features. 2-fold cross-validation was performed 100 times to calculate the performance metrics like F1-score, precision, and recall. The accuracy of this feature selection and SVM model was very much high when compared with the models developed by Bestgen (2019). According to Kwon et al. (2013) in a biased news article, the content will be used more repeatedly and more proper nouns will be used compared to real news.

A similar approach to Jiang et al. (2019) was followed by Huang and Lee (2019) to detect hyperpartisan news articles. Huang and Lee also used the BERT case version along with ELMo for generating word embeddings and the BERT classifier for prediction. The accuracies of BERT and ELMo were 68.4% and 60.8%, respectively on validation data and these accuracies were achieved after fine-tuning both the models. Before fine-tuning, the models achieved accuracies below 40%. This research concluded that the writing style of articles contributes more towards the detection of articles than the title of the articles. Huang and Lee also tested several lengths of articles as input to BERT like Drissi et al.

(2019), to find the optimum length of articles that yields higher accuracy as BERT can take a maximum of 512 tokens of an article as input. Huang and Lee also tried different combinations like the first 100 words of the article, last 100 words of the article, and middle 100 words of the article as input to BERT and concluded that BERT can extract context from local words, and does not need to train with the entire sentence for doing predictions.

A new approach of calculating the bias score of online articles was done by Patankar and Bose (2017). Patankar and Bose have developed a web browser extension that is capable of calculating bias scores. When a user opens an online news article, the browser extension will calculate the biased score and displays it to the user. An article with a high biased score means it contains hyperpartisan news, and the reader can avoid the article without reading it. The word vectors of the articles were compared with the words in Wikipedia Neutral point of view (NPOV) and generated a biased score based on the similarity. According to Patankar and Bose (2017), Wikipedia articles and opinion articles has got less biased score than news articles.

Furthermore, a similar approach, such as calculating bias score (Patankar & Bose, 2017) was followed by to detect politically biased news on the internet. Patankar, Bose and Khanna calculated the biased score for each article and displayed it with the article so that the article with a very high biased score could be avoided by the readers. When calculating the bias scores, Word2vec was to find the number of words in the news article that are matching with the biased words in the NPOV lexicon. The biased score was mainly based on the number of biased words present in the articles, and also the number of similar articles found on the internet.

A lightweight neural network named WiSARD classifier was tested and compared against machine learning algorithms, such as naïve bayers, logistic regression, and SVM by Cavalcanti et al. (2017). WiSARD performed extremely well in extracting the context of words in online news articles. The WiSARD classifier and all the machine learning algorithms were trained with unigram, bigram, and trigram tokens. The accuracy of trigram models is very high compared to unigram and bigram models. A 4-bit trigram WiSARD classifier outperformed the SVM by 5% of accuracy. Whatsapp is a major source of fake news propagation as there was no authority to control the bulk messages sent through it. An LSTM based approach was proposed by Gaglani, Gandhi, Gogate and Halbe (2020) to detect the fake news spreading on WhatsApp. The authenticity of the news will be verified by comparing WhatsApp news with the real sources of news. Sentiment analysis of tweets during the 2019 India elections was done by Hitesh, Vaibhav, Kalki, Kamtam and Kumari (2019), which is similar to the approach followed by Alabdulkarim and Alhindi (2019) Hitesh et al. (2019). used Word2vec word embeddings to train random forest classifier, whereas Alabdulkarim and Alhindi (2019) used TF-IDF vectors trained on SVM. According to Hitesh et al. (2019), the use of more contextualized word representations has improved the accuracy of the model compared to the prediction model by using bag-of-words representation.

2.4. Empirical review of other related past studies aside hypertisans news articles

More additionally, Aswani, Kar and Ilavarasan (2018) employed hybrid methods, modified K-means -integrated Levy flight Firefly Algorithm LFA, and chaotic map as an extension with Firefly algorithms – and Fuzzy C-Means clustering algorithms to spam profiles using Twitter media analytics. The outcome revealed that out of a total of over 184,000 tweets from 14,200 profiles analyzed, 13 significant factors, which is Fuzzy C-Means further identified into two categories – spammers vs non-spammers, were derived from the Twitter media analytics. The result also showed that FA works as the fastest means of detecting the spams while LFA works better to give the best accuracy Rajendran and Sundarraj (2021). finds different categories of user preferences the Latent Dirichlet Allocation on Wikipedia data to extract browsing topics – movies and restaurants in specific. In a bid to examine the factors that contribute to the rapid spread of misinformation in the Twitter domain Aswani, Kar and Ilavarasan (2019) investigated an average of about 1.5 million tweets involving misinformation.

According to the research results, tweet sentimentality and polarizability play an important role in assessing whether or not the posts is credible. A deeper investigation reveals that such tweets contain a higher level of astonishment, along with other emotions. Furthermore, when it comes to misleading information, tweets with case-neutral content frequently lack viral potential Ghosh and Sanyal (2021). employed different machine learning approaches, such as gradient boosting, LSTM, deep neural network, and extra tree regression to analyze market volatility returns and uncertainty using India VIX and NIFTRY returns as case study timelining it on the coronavirus outbreak, the result suggested that both target studies can be effectively projected to serve actionable visions for the governments and policymakers.

For false news classification, Nasir, Khan and Varlamis (2021) also introduces a unique hybrid deep learning model that blends convolutional and recurrent neural networks. The model was successfully verified on two misinformation databases (ISO and FA-KES), yielding detection accuracy that is much better than existing non-hybrid baseline techniques. Further studies on the applicability of the suggested model across diverse datasets yielded encouraging results Rathore, Kar and Ilavarasan (2017). did an empirical review of social media analytics for business purposes, they conclude that different methods such as tag detections, cross-media data classification, geospatial data processing, and signatures of tweet activities could be used as novel methods to gain meaningful insights from media analytics to improve business strategy Ahn, Son and Chung (2021). also used regression model analysis to analyze collected tweets Ridgecrest earthquake in Southern California that happened in July 2019. The tweets were from some targeted organizations such as media companies, research groups, government agencies, and nonprofit organizations. The result revealed a positive and significant relationship between the topics from media outlets and the number of favourite retweets messages, although the topics do not include topics from the Twitter users in the targeted organization.

2.5. Benchmark datasets for hyperpartisan news detection

As hyperpartisan news detection was less explored there were very few datasets available when compared with the number of datasets available for fake news detection. A benchmark dataset named “Liar, Liar Pants on Fire” for hyperpartisan news detection was developed by Wang (2017). This dataset was prepared by collecting and manually labeling 12.8 K short news articles from politifact.com. This dataset can be used to train machine learning models, such as SVM, random forest, logistic regression etc. and also proved that the combination of text and meta-data yields greater performance. The organizers of SEMEVAL-2019 (Palic et al., 2019) also developed a dataset that can be used to detect hyperpartisan news. The bi-article dataset consists of 1273 manually labelled articles and the by-publisher dataset consists of 754,000 articles. Both these datasets are in HTML format as these are collected from online articles.

There are two datasets developed by Palic et al. (2019). One is the by-article dataset, and the second one is the by-publisher dataset. This research will make use of the 1273 by-article datasets of which 645 articles will be used to train the model, and the remaining 628 articles will be used for validation. The validation dataset would be very well balanced with 50% of biased articles and 50% of unbiased articles.

3. Methodology

To detect hyperpartisan news articles, three different machine learning models, which are explained below, were employed. This research makes use of the by-article dataset published at SEMEVAL-2019 (Palic et al., 2019). The bi-article dataset consists of 1273 news articles, out of

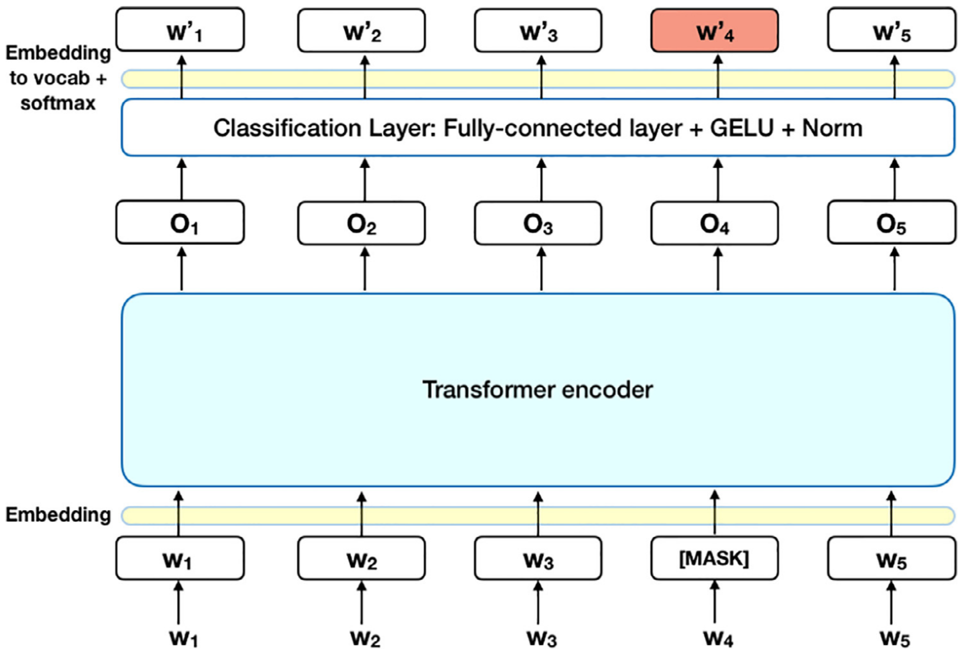

Fig. 1. architecture and input format of BERT.

which 645 articles were used for training the model and the remaining 628 articles were used to validate the model. The training and the validation data has no overlap. The news articles contain HTML tags and elements as these were extracted from online news sources. In preprocessing of data all the HTML links, hash, and tags were removed. The browser error messages and the advertisements were also removed. All the punctuations and special characters were preserved because the hyperpartisan news predictions are based on the style of writing of news.

To run the pre-trained BERT model in the TensorFlow interface, transformers (Wolf et al., 2019) from the Hugging Face library were installed. Hugging Face library is the most widely PyTorch and Tensorflow interface to implement BERT. BERT can be used for our classification purpose as this library included task-specific pre-built classes for text classification, question and answering, and next word or sentence prediction.

The "uncased" version of the BERT tokenizer was used to convert the text into tokens as BERT takes only tokens as input. A special token called [CLS] was used at the beginning of every sentence, and sentences were separated by using a special token called [SEP]. The architecture and the input format to BERT were shown in Fig. 1.

3.1. Padding and attention mask

Sentences that are given as input to BERT must have the same length with a maximum of 512 sentences, which were either padded or truncated to have a fixed input sentence length. Since the articles in the dataset have different lengths, padding or attention masks were used to convert all the articles to be of the same length. Since the maximum length of the sentence will affect the training size and test accuracy, three different article lengths (100, 250, and 512) were achieved, padding or attention masks, to find the optimum length of articles that yields the maximum accuracy. A special token called [PAD] was used to perform padding. By default, all the tokens are generated by the BERT tokenizer which is "1″. [PAD] will, then, change the token id of a token from "1″ to "0″. This makes the real tokens to be distinguished from padded tokens by observing the token id. That is, if the token id is "0″ then it is a padded token, otherwise, it is a real token.

The pre-trained BERT model was trained with 80% of the training dataset, and the remaining 20% was used to test the model. The sequence of tokens given as input to BERT would be processed in the transformers neural network and produces word embeddings or vectors for each token based on the context of the text. These word embeddings were used to train the BERT classifier present in the output layer of BERT to perform the prediction task. After training the model with 80% of the training data in the by-article dataset, the BERT model was tested with 20% of the dataset and performance metrics, such as precision, recall and F1-Score were calculated to analyze the performance of the model. This process was repeated by training BERT with different article lengths and performance metrics were noted.

Finally, the trained BERT model was validated with some unseen data. The validation dataset consists of 628 news articles. The performance metrics were calculated by validating the BERT model. Traditional natural language processing models perform basic tasks efficiently. Their performance is poor when they have to follow the context of the text. The contextual meaning of the text must be followed while performing classification or prediction to obtain good accuracy. In this research, ELMo was used to detect hyperpartisan news articles based on their content. ELMo stands for Embeddings from Language Models and was developed by AllenNLP. ELMo represents words in vector space as embeddings which are useful in achieving better results in NLP tasks.

ELMo represents words as vectors that are computed by using a bi-directional long short-term memory (LSTM). Each layer in the bi-directional LSTM will have two passes, one is the forward pass and the other one is the backward pass. The architecture of ELMo is shown in Fig. 2.

ELMo uses a convolutional neural network called bi-LSTM, a bi-directional two-layer neural network, to generate word embeddings for text. The first layer in the bi-LSTM takes raw text as input. In the forward pass, the context of a particular word i retrieved based on the context provided by its previous neighbourhood words. In the backward pass, the context of a particular word is retrieved based on the words which are next to it. By using the information from these pairs of passes, intermediate word embeddings or vectors will be formed. The second layer in the bi-LSTM will be fed with these intermediate word embeddings. Finally, word embeddings are the weighted average of the intermediate word embeddings and the raw text words.

3.2. Preparing ELMo word embeddings for the dataset

A pre-trained ELMo model, above 350 mb in size, was loaded. Then to get the ELMo vectors, which is a three-dimensional tensor, for an article, a list of strings were passed into the object "Elmo", and also printed

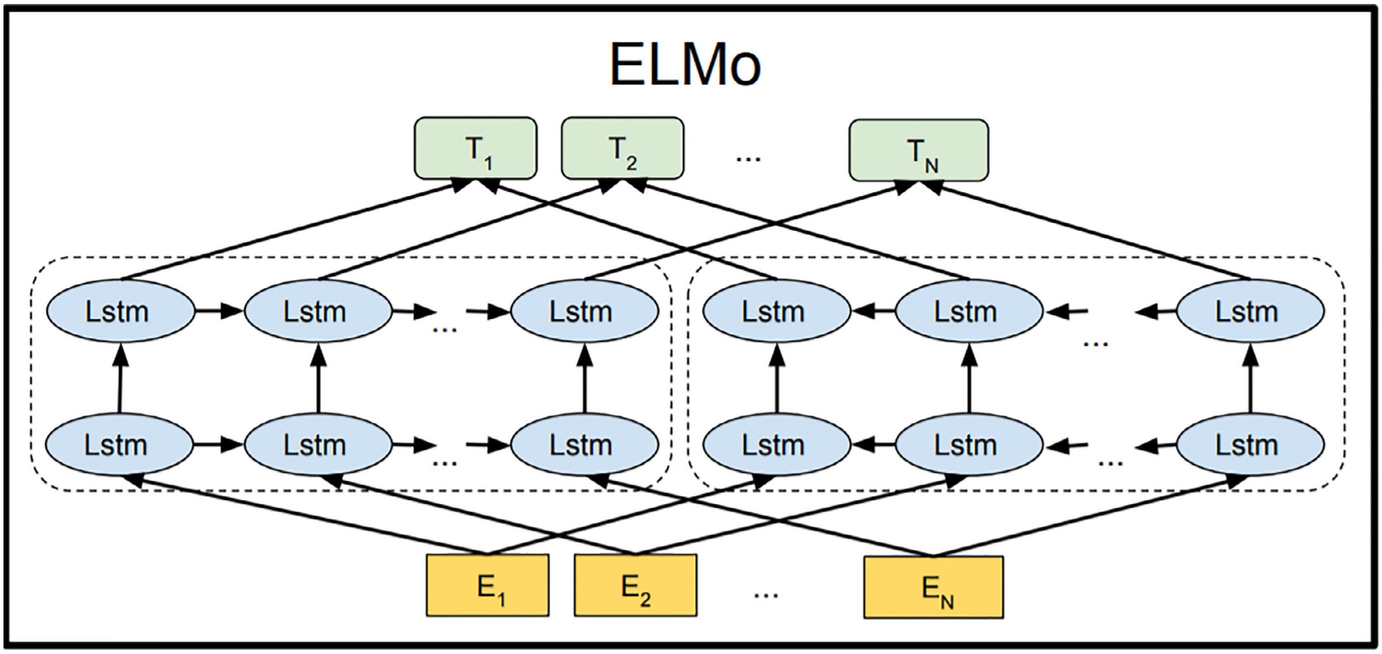

Fig. 2. ELMo architecture.

the shape of ELMo word embeddings. The first dimension in the tensor corresponds to the number of training samples in the dataset. The second dimension in the tensor corresponds to the maximum length of the longest string in the input list of strings. The third dimension in the tensor is the length of the ELMo vector. A user-defined function named elmo_vectors () was used to extract word embeddings of the entire dataset. However, to get the vector representation of the entire news article, the mean of the ELMo vectors of constituent words of the news article was calculated.

Since the dataset consists of 645 news articles and the length of each article is very high, the computational time to calculate the word embeddings of all the news articles in a single go is very high. Hence, the dataset was split into batches of 100 news articles and passed each batch to the elmo_vectors () function to calculate the vectors. This process was repeated through all the batches of news articles in the dataset to extract the ELMo word embeddings. Once extracted vectors for all the batches were concatenated as a single array of vectors. As the computational time to generate word embeddings for the entire dataset is very high because of the large number of news articles in the dataset, the word embeddings were saved in a pickle file to use in future.

3.3. Training and validation

The ELMo vectors of the news articles were used to build a classification model that can predict hyperpartisan news articles. Since ELMo alone cannot classify text, a classifier has to be added to the output layer of the ELMo model. So Random Forest classifier was added to the ELMo output layer to detect hyperpartisan news articles. The dataset was divided into two parts, the training dataset with 80% of news articles and the test dataset with 20% of news articles from the training dataset. Random Forest classifier was loaded from sklearn.linear_model and f1_score from sklearn metrics. To build a classification model using the ELMo word vectors, the Random Forest classifier was fit with the ELMo vectors and labels. Since the training dataset which was used is largely imbalanced, the class weight parameter was set as "balance" in the Random Forest classifier. The Random Forest classifier was trained with the word embeddings generated for the news articles using ELMo. The contextual features of genuine and hyperpartisan news articles were trained to the Random Forest classifier. After that, the prediction model was tested by using the test dataset which is 20% of the training dataset.

After training and testing, the ELMo and Random Forest prediction models were tested with the unseen validation data. The validation dataset consists of 628 news articles, out of which 50% of news articles are tagged as hyperpartisan and the remaining 50% are tagged as non-hyperpartisan. The validation dataset is very well balanced. Five-fold cross-validation was used to validate the model and the performance metrics were calculated. The performance metrics that are used to evaluate the model are F1 score, precision, and recall.

3.4. Hyperpartisan article detection using Word2vec and random forest classifier

This model makes use of word embeddings generated using Word2vec to train the Random Forest classifier to predict hyperpartisan news articles. The methodology and analysis are detailed below. Word2vec model requires removing noise and unwanted characters. Stop words such as "a", "an", "the", "as" were removed from the dataset as these do not contribute to the context of the news article and also converted the text to lowercase and removed white spaces. Google's word2vec gensim pre-trained model was used to generate word embeddings for 645 news articles in the dataset. This pre-trained model which was trained on three billion running words was used to generate word embeddings for the news articles which are of dimension 300. These word vectors were used to train the Random Forest classifier to detect hyperpartisan news articles.

Random Forest classifier is an ensemble of decision trees. A group of decision trees works together and each decision tree in the group contributes to the class prediction. The ensemble decision of all the decision trees in the Random Forest classifier is the final prediction. Random Forest classifier is very accurate in prediction as the ensemble prediction is accurate than individual prediction. Predictions of some of the trees might be wrong but as a group, they protect each other and produces a more accurate class prediction. The accuracy of the Random Forest clas-

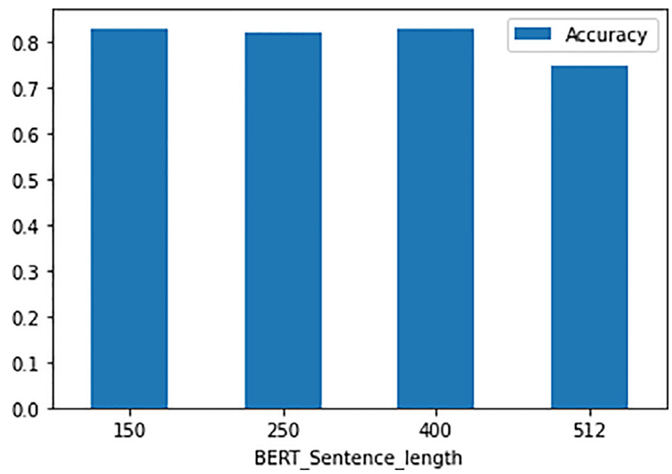

Fig. 3. Accuracies of BERT for different input lengths.

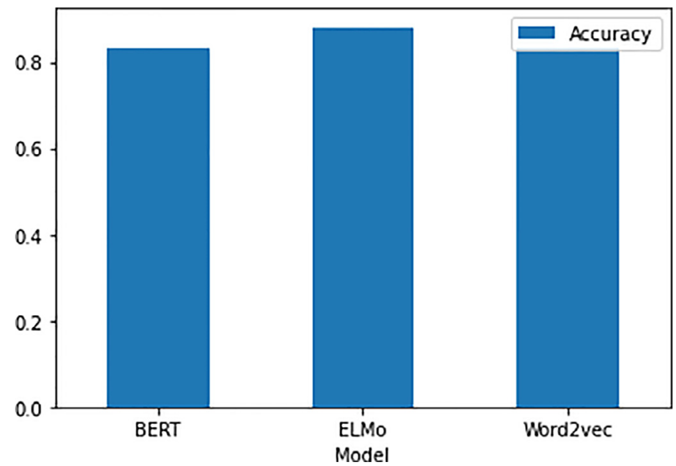

Fig. 4. Accuracies of BERT, ELMo, and Word2vec.

sifier depends on the number of decision trees it uses. As the number of decision trees is high, greater is the correlation between them which contributes towards greater accuracy.

The word2vec word vectors extracted for the news articles and the labels were used to train the Random Forest classifier. The word vectors and their labels were split in an 80:20 ratio i.e. 80% of the data is used to train the Random Forest classifier and 20% is used to test the classifier. After training the model with the training dataset, it was tested with the test data. Random Forest classifier has a built-in function to handle largely imbalanced classes. The class weight parameter can be used to make sure the training data consists of balanced classes. To do this the class weight parameter was set as "balance" in the Random Forest classifier as the training dataset is imbalanced.

After completion of training and testing, the Word2vec and Random Forest classifier model was validated on an unseen validation dataset. The validation dataset consists of 628 news articles. The validation dataset is very well balanced as it contains equal proportions of each class. Five-fold cross-validation was performed on validation data to calculate the performance metrics. Precision, Recall, and F1-score were used as performance metrics to compare with the other state-of-art models.

4. Results

The word embeddings generated using BERT are trained on the in-built BERT classifier present on the output layer of BERT. The maximum sentence length that BERT can take is 512. To find the optimum length of the article that can fetch more accuracy out of BERT, different lengths of articles were used to train and test BERT. The different article lengths that were used are 150, 250, 400, and 512. BERT model with input sentence length 400 got the least accuracy of 0.75, where are remaining all sentence lengths achieved almost equal accuracies of 0.83. This shows enough evidence that BERT can achieve context from local words, and it does not need the entire article to make predictions. The bar chart showing the accuracies of the BERT model for different sentence lengths are shown in Fig. 3.

The Word2vec model along with the Random forest classifier got an accuracy of 0.83 which is almost equal to the accuracy of the BERT model. Despite being a non-contextual model Word2vec performed very well compared to BERT. The ELMo and Random forest classifier got an accuracy of 0.88, which is par better than BERT and Word2vec. The accuracies of the three different models were compared by using the bar chart in Fig. 4.

5. Discussion

5.1. Contributions to literature

The word embeddings for news articles in the bi-article dataset were generated by using the BERT base version which has 12 transformer layers. The word embeddings for all the 645 articles in the bi-article dataset were developed by using the BERT base, and these embeddings were used to train the in-built BERT classifier to make the predictions. The dataset was divided into two parts, 80% for training the model and the remaining 20% to test the model. The training dataset was further split into two parts, 20% of the training dataset was used to validate the model. The maximum sentence length that BERT can take as input is 512, but the maximum length of the article in the dataset is 2700, and the average length of all articles is 1600. As BERT cannot take articles having more than 512 words, different input lengths to BERT were tested. Different article lengths like 150, 250, 400, and 512 were used to train and test the BERT model to find the optimum input length to BERT that can yield maximum accuracy. BERT models with 150, 250, and 512 input sentence lengths have got almost equal accuracy of 83%, but the BERT model with input sentence length has got only 75%. The performance of this model is compared with the other state of art models in the below paragraphs.

The bi-article dataset used in this research has only text features, it does not have any images. Thus only the BERT Base model was used to extract text features, whereas the SPOTFAKE model developed by Giachanou et al. (2020) to detect fake news spreading across social media has used both textual features and image features to make the classification. BERT was used in SPOTFAKE to analyze textual features, and VGG- 19 was used to analyze image feature. The SPOTFAKE model failed to deal with longer length articles, but the BERT model developed in this research to detect hyperpartisan news articles has experimented with different article lengths and proved that BERT can extract context from local words, and we do not need to give the entire article as input to BERT. There is no evidence found on the optimum length of the article as input to BERT that can yield maximum accuracy. The BERT model with 400 words maximum sentence length has got 8% less accurate when compared with other sentence lengths 150, 250, and 512. The difficulty of handling higher length articles in SPOTFAKE developed by Giachanou et al. (2020) was overcome in this research by setting the input length to BERT.

A similar model like SPOTFAKE (Giachanou et al., 2020) was developed by Singhal, Shah, Chakraborty, Kumaraguru and Satoh (2019). A baseline BERT model was developed by Singhal et al. (2019) by training the textual features, later VGG-16 was used to train visual information along with textual features using BERT. This multi-model fake news detection system outperformed the baseline BERT that was trained with only textual features by 4.19% in terms of accuracy proving that the visual information present in the news adds extra advantage while making the prediction. As the bi-article dataset used in this research to detect hyperpartisan news has no visual information only BERT was used to train textual features, but the performance of the model is as good as the fake news detection system developed by Singhal et al. (2019). The fake news detection system developed by Singhal et al. (2019) has not focused on training the BERT model with different sentence lengths, whereas the hyperpartisan news detection system has trained and tested with different input article lengths and different sentence lengths. The hyperpartisan news detection system has got 8% less accuracy when compared with SPOTFAKE.

Clickbait is a short attractive headline or hyperlink intended to grab the attention of the reader to click on it. A machine learning model was used by Genç (2019) to detect clickbait in the Twitter dataset in the Turkish language. The accuracy and F1-score of the clickbait detection model developed by Genç (2019) are a bit more than the hyperpartisan news detection system because Genç (2019) has only used article headings to train the model, not the entire article. When dealing with articles with huge text BERT and ELMo should be preferred as they are more contextual compared to machine learning models which treat text as a bag of words.

This research to detect hyperpartisan news articles has outperformed all the machine learning and neural networks that are used to detect hyperpartisan news in SEMEVAL-2019 by Palić et al. (2019). SEMEVAL-2019 has proved that neural networks work more effectively in detecting hyperpartisan news articles than machine learning models, but failed to implement BERT more effectively.

The machine learning model developed by Alabdulkarim and Alhindi (2019) using a Support vector machine (SVM) has failed to process the text contextually. The accuracy of the model developed by Alabdulkarim and Alhindi (2019) is only 0.742 as SVM failed to grab the contextual meaning of words, whereas the accuracy of the hyperpartisan news detection model that was developed by using BERT in this study work is 0.83. This gives enough evidence that neural networks are more suitable to handle more complex datasets.

The accuracy of the hyperpartisan article detection model developed by Huang and Lee (2019) using BERT was 0.68, whereas the accuracy of the hyperpartisan detection model developed in this research by using BERT is 0.83 with the same dataset, which is 20% higher. Huang and Lee tried with different lengths of input to BERT and also concluded that BERT can extract context from local words. The BERT model developed in this research calculated weights and adjusted the learning rate of BERT efficiently when compared to the model developed by Huang and Lee, which is the key reason for improving the accuracy of the model. The research by Huang and Lee (2019) also tried different combinations like first 100 words of the article, the last 100 words of the article, and the middle 100 words of the article as input to BERT, whereas the BERT model developed in this research used only starting 150, 250, 400, or 512 words of the news article.

The hyperpartisan news article detection model developed in this research used word embeddings of the news articles to train the classifier, where as Patankar and Bose (2017) used these word embeddings to compare with words in Wikipedia Neutral point of view (NPOV) and generated bias scores for the articles. Though Patankar and Bose have got accuracy same as the hyperpartisan news detection model developed in this research, it is very difficult to implement and develop a web extension because the web extension has to compare word vectors every article with the words in NPOV.

The 12 - layer transformer network of BERT developed in this research outperformed the light weight WISARD classifier developed by Cavalcanti, Lima, De Gregorio and Menasche (2017). Both BERT and WISARD classifier has worked very well compared with the machine learning algorithms like na ve bayers, logistic regression, and support vector machine. WiSARD performed extremely well in extracting the context of words in online news articles. The WiSARD classifier and all the machine learning algorithms were trained with unigram, bigram, and trigram tokens. The accuracy of trigram models is very high compared to unigram and bigram models. A 4-bit trigram WiSARD classifier outperformed the support vector machine by 5% of accuracy.

Over all, the BERT model has performed very well in detecting hyperpartisan news articles when compared to machine learning models like na ve bayers, logistic regression, and support vector machine, but failed to compete with ELMo. Though Word2vec is not a contextual model it has got accuracy very close to BERT. Finally, the main advantage of BERT is having an inbuilt classifier to perform downstream tasks, whereas ELMo and Word2vec have to employ a classifier to perform the classification task as they do only can generate word embeddings for text.

5.2. Practical implications

Word2vec was used to generate the word vectors for all the 645 news articles present in the dataset. These word vectors were later trained on a random forest classifier to make the prediction model. This prediction model yielded an accuracy of 0.83, which is very high when compared to the machine learning models that were developed in past to detect hyperpartisan news articles. Word2vec (Ji et al., 2019) is the first neural network that is developed to generate word embeddings as machine learning models failed to grab the contextual meaning of words. This Word2vec can be used along with the machine learning models to perform downstream tasks with greater accuracy. A word embedding is a vector representation of a word in space. Word2vec treats the words with similar meaning as same and assigns them with the same word vector. Word2vec calculates the cosine similarity between two words and if the similarity is “1″ then those two words will be assigned with the same vector, otherwise, it treats the two words as different.

The word embeddings generated by Word2vec for the bi-article dataset was in dimension 300×645. It means it uses a vector space with three hundred coordinates to represent one article, and there are 645 articles in the dataset. As the computational time for generating the word vectors for all the articles is very high, those are saved as a pickle file for future use. The pickle file containing the word vectors for all the 645 news articles were used to train the Random forest classifier. Word2vec failed to generate the word vectors for 4 articles in the dataset, it assigned null vectors for those four articles. As the random forest classifier does not take null values, the word vectors for those four articles were filled with the average values of the remaining 641 articles. This could be done by replacing the null values with zeros as well, but it is not a good practice as it affects the accuracy of the model. The Random Forest classifier was trained with 80% of the articles in the dataset and the remaining 20% was used to test the model. Accuracy, Precision, recall, and F1-score were calculated to evaluate the model.

The accuracy of Word2vec and Random forest classifier (0.83) is the same as the accuracy of the BERT model (0.83) developed in this study work. Even though Word2vec is not a contextual model, it performed very well up to the accuracy of the deep contextual neural network like BERT. The implementation of Word2vec is very simple, but it needs an additional machine learning algorithm to perform the prediction task, whereas BERT has its inbuilt classifier to perform the downstream tasks. The ELMo and Random forest classification model developed in this research work to detect hyperpartisan news article detection have outperformed the Word2vec and Random Forest model by 5% in terms of accuracy. The contextual behavior of ELMo and its ability to process text right to left and from left to right makes it more accurate compared to Word2vec.

In terms of how this research contributes towards information management of hyperpartisian news there are notable practical implications. Firstly, both ELMo and BERT can generate word embeddings for words based on their context, whereas Word2vec fails to generate word embeddings based on the context or position of the words used in a sentence.

Word2vec is a context-independent model like machine learning models which treats text as a bag of words and generates vectors. Word2vec does not consider the position or meaning of the word in the text while generating word embeddings. The use of transformers (Wolf et al., 2019) in BERT (Devlin et al., 2019) made it bi-directional, thus it can generate a vector for a word in a text-based on the context it is used. In the forward pass of the transformer, layer text will be processed from left to right and BERT will extract the context of a word based on the previous words to it. In the backward pass of the transformer, layer text will be processed right to left and BERT will extract the context of a word based on the words next to it. The forward and backward passes will be performed at the same time which makes BERT bi-directional. ELMo uses LSTM's which are capable of processing text bi-directionally but the forward pass and the backward pass will be performed one after another which makes it pseudo-bi-directional.

For practical usage, consider the following example to clearly understand the differences in the way of generating word embeddings by BERT, ELMo, and Word2vec. "The police officer went to the cell with his cell phone to capture the image of the prisoner". The word cell was used two times in the above sentence, and it has two different meanings. The two different contexts of the word cell are prison and mobile. Word2vec generates the same word vector for the word cell irrespective of it having two different meanings which lead to wrong predictions, whereas ELMo and BERT generate two different vectors for the word cell based on the position and the context of the word.

Since BERT and ELMo are context-dependent, these models are still needed even after used to generate vectors, to train the models that are used to perform downstream tasks. So we will include an output layer in BERT and ELMo to perform our classification task. Whereas pre-trained Word2vec can be used to generate word embeddings for the text, and these word embeddings can be used to train any classifier model. BERT has an inbuilt classifier on the output layer to perform downstream tasks, so BERT is still required after the generation of word embeddings as well to perform classification tasks.

The accuracy of ELMo and Word2vec depends on the type of classifier that is used along with them. As ELMo and Word2vec do not have any inbuilt classifier to make predictions, there is a need for an external machine learning model to perform downstream tasks. BERT has an inbuilt BERT classifier on its output layer to perform downstream tasks like classification, fill in the blanks, question and answer etc.

6. Conclusion

The three models BERT, ELMo, and Word2vec performed very well in detecting hyperpartisan news articles. Though BERT is a bi-directional neural network, its accuracy is not fair enough when compared to uni-direction ELMo, and non-contextual Word2vec models. The advantage with BERT when compared to ELMo and Word2vec is, we do not need to pass the entire sentence as input as it can extract context from local words. This is more advantageous when dealing with datasets that contain lengthy articles.

When it comes to false news, it is hyperpartisan or any other news, news contents often lack viral potential. The false news maybe spread deliberately through the recognized news agency or unintentionally, which can also be deliberate, to gain television rating point, or involuntarily, under the assumption that the information is accurate. Hence, this research, evidenced from the described machine learning models, would help the governments, news' readers, and other political stakeholders to detect any hyperpartisan news, and also help policy to track, and regulate, misinformation about the political parties and their leaders.

This research used the bi-article dataset published in SEMEVAL-2019 (Palić et al., 2019). There are 645 news articles in the dataset, which were used to train the three neural networks BERT, ELMo, and Word2vec. There is also another dataset published in SEMEVAL-2019 (Palić et al., 2019), named as bi-publisher which contains 7,50,000 news articles. This research work is limited only to bi-article dataset because the bi-publisher dataset requires large CPU and GPU powers.

In terms of future research, while the current study applied BERT, ELMo, and Word2vec models to generate word embeddings, and the Random Forest classifier was used to make predictions, BERT model requires lots of data to train with. Hence, in furthering the findings from this study, the BERT model can be trained with a by-publisher dataset that has more than seven lakh political news articles, but this requires larger training time, and high CPU and GPU configurations. And also different combinations of data augmentation techniques like random swap, random deletion, random addition, and synonym replacement can be used to enhance the bi-article dataset. Different classification algorithms like Na ve bayers, Support vector machine (SVM), Decision trees, and Logistic regression can be used to train the word embeddings generated by ELMo, and Word2vec.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahn, J., Son, H., & Chung, A. D. (2021). Understanding public engagement on twitter using topic modeling: The 2019 ridgecrest earthquake case. *International Journal of Information Management Data Insights, 1*(2), Article 100033.

Alabdulkarim, A., & Alhindi, T. (2019). Spider-jerusalem at SemEval-2019 Task 4: Hyperpartisan. *News Detection, 238*, 985–989. 10.18653/v1/s19-2170.

Aswani, R., Kar, A. K., & Ilavarasan, P. V. (2018). Detection of spammers in twitter marketing: A hybrid approach using social media analytics and bio inspired computing. *Information Systems Frontiers, 20*(3), 515–530.

Aswani, R., Kar, A. K., & Ilavarasan, P. V. (2019). Experience: Managing misinformation in social media—Insights for policymakers from Twitter analytics". *Journal of Data and Information Quality, (JDIQ), 12*(1), 1–18.

Badawy, A., Ferrara,.E., & Lerman,.K. (2018). "Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign," arXiv, pp. 258–265,

Bastos, M. T., & Mercea, D. (2019). The brexit botnet and user-generated hyperpartisan news. *Social Science Computer Review, 37*(1), 38–54. 10.1177/0894439317734157.

Bestgen, Y. (2019). "Tintin at SemEval-2019 task 4: Detecting hyperpartisan news article with only simple tokens," pp. 1062–1066, doi:10.18653/v1/s19-2186

Caramancion, K. M. (2020). Understanding the impact of contextual clues in misinformation detection. In *Proceedings of the IEMTRONICS 2020 international IOT, electronics and mechatronics conference*. 10.1109/IEMTRONICS51293.2020.9216394.

Cavalcanti, R. D., Lima, P. M. V., De Gregorio, M., & Menasche, D. S. (2017). Evaluating weightless neural networks for bias identification on news. In *Proceedings of the 2017 IEEE 14th international conference on networking, sensing and control (ICNSC 2017)* (pp. 257–262). 10.1109/ICNSC.2017.8000101.

Celena, C., Park, C., Dwyer, J., & Medero, J. (2019). Harvey mudd college at SemEval-2019 Task 4: The carl kolchak hyperpartisan news detector. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 957–961). Association for Computational Linguistics. pages.

Chawda, S., Patil, A., Singh, A., & Save, A. (2019). A novel approach for clickbait detection. In *Proceedings of the 2nd international conference on trends in electronics and informatics, ICOEI 2019* (pp. 1318–1321). no. Icoei. 10.1109/ICOEI.2019.8862781.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the HLT-NAACL 2019: Conference of the North American chapter of the association for computational linguistics: Human language technologies (HLT-NAACL)* (pp. 4171–4186). no. Mlm.

Dey, A., Rafi, R. Z., Hasan Parash, S., Arko, S. K., & Chakrabarty, A. (2019). Fake news pattern recognition using linguistic analysis. In *Proceedings of the 2018 joint 7th international conference on informatics, electronics & vision and 2018 2nd international conference on imaging, vision & pattern recognition (ICIEV)* (pp. 305–309). 10.1109/ICIEV.2018.8641018.

Drissi, M., Sandoval P., Ojha, V., & Medero, J. (2019). "Harvey mudd college at SemEval-2019 task 4: The clint buchanan hyperpartisan news detector," arXiv, pp. 962–966, doi:10.18653/v1/s19-2165.

Dukic, D., Keca, D., & Stipic, D. (2020). Are you human? *Detecting Bots on Twitter Using BERT*, 631–636. 10.1109/dsaa49011.2020.00089.

Fan, X., Feng, Z., Yang, X., Xu, T., Tian, J., & Lv, N. (2018). Haze weather recognition based on multiple features and random forest. In *Proceedings of the 2018 international conference on security, pattern analysis, and cybernetics (SPAC)* (pp. 485–488). 10.1109/SPAC46244.2018.8965544.

Gaglani, J., Gandhi, Y., Gogate, S., & Halbe, A. (2020). Unsupervised WhatsApp fake news detection using semantic search. In *Proceedings of the international conference on intelligent computing and control systems, ICICCS 2020* (pp. 285–289). no. Iciccs. 10.1109/ICICCS48265.2020.9120902.

Genç, Ş. (2019). "Sosyal medyada ' clickbait ' haber başliklarinin makine öğrenmesi algoritmalari ile tespit edilmesi detecting ' clickbait ' news on social media using machine learning algorithms," pp. 0–3, George, J., Skariah, S. M., & Xavier, T. A (2020). Role of contextual features in fake news detection: A review. In *Proceedings of the international conference on innovative trends in information technology (ICITIIT)* (pp. 1–6). 10.1109/ICITIIT49094.2020.9071524.

Ghosh, I., & Sanyal, M. K. (2021). *Introspecting predictability of market fear in Indian context during COVID-19 pandemic: An integrated approach of applied predictive modelling and explainable AI*: 1.

Giachanou, A., Zhang, G., & Rosso, P. (2020). "Multimodal multi-image fake news detection," pp. 647–654, doi:10.1109/dsaa49011.2020.00091.

Hitesh, M. S. R., Vaibhav, V., Kalki, Y. J. A., Kamtam, S. H., & Kumari, S. (2019). Real-time sentiment analysis of 2019 election tweets using word2vec and random forest model. In *Proceedings of the 2nd international conference on intelligent communication and computational techniques (ICCT)* (pp. 146–151). 10.1109/ICCT46177.2019.8969049.

Hlaing, M. M. M., & Kham, N. S. M. (2020). Defining news authenticity on social media using machine learning approach. In *Proceedings of the IEEE conference on computer applications(ICCA)*. 10.1109/ICCA49400.2020.9022837.

Huang, G. K. W., & Lee, J. C. (2019). Hyperpartisan news and articles detection using BERT and ELMo. In *Proceedings of the international conference on computer and drone applications (IConDA)* (pp. 29–32). 10.1109/IConDA47345.2019.9034917.

Ji, S., Satish, N., Li, S., & Dubey, P. K. (2019). Parallelizing Word2Vec in shared and distributed memory. *IEEE Transactions on Parallel and Distributed Systems, 30*(9), 2090–2100. 10.1109/tpds.2019.2904058.

Jiang, Y., Petrak, J., Song, X., Bontcheva, K., & Maynard, D. (2019). "Team Bertha von Suttner at SemEval-2019 Task 4: Hyperpartisan news detection using ELMo sentence representation convolutional network," pp. 840–844, doi:10.18653/v1/s19-2146.

Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. In *Proceedings of the IEEE international conference on data mining (ICDM)* (pp. 1103–1108). 10.1109/ICDM.2013.61.

Madhusudhan, S., Mahurkar, S., & Nagarajan, S.K. (2020). "Attributional analysis of multi-modal fake news detection models (Grand Challenge)," pp. 451–455, doi:10.1109/bigmm50055.2020.00074.

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights, 1*(1), Article 100007.

Pal, K., & Patel, B. V. (2020). Automatic multiclass document classification of Hindi poems using machine learning techniques. In *Proceedings of the international conference for emerging technology (INCET)* (pp. 1–5). 10.1109/INCET49848.2020.9154001.

Palić, N., Vladika, J., Čubelić, D., Lovrenčić, I., Buljan, M., & Šnajder, J. (2019). "TakeLab at SemEval-2019 task 4: Hyperpartisan news detection," pp. 995–998, doi:10.18653/v1/s19-2172.

Patankar, A.A., Bose, J., & Khanna, H. (2018). "A bias aware news recommendation system," arXiv, doi:10.1109/ICSC.2019.00051.

Rajendran, D. P. D., & Sundarraj, R. P. (2021). Using topic models with browsing history in hybrid collaborative filtering recommender system: Experiments with user ratings. *International Journal of Information Management Data Insights, 1*(2), Article 100027.

Rathore, A. K., Kar, A. K., & Ilavarasan, P. V. (2017). Social media analytics: Literature review and directions for future research. *Decision Analysis, 14*(4), 229–249.

Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P., & Satoh, S. (2019). SpotFake: A multi-modal framework for fake news detection. In *Proceedings of the IEEE fifth international conference on multimedia big data (BigMM)* (pp. 39–47). 10.1109/BigMM.2019.00-44.

Wang, W. Y. (2017). 'Liar, liar pants on fire': A new benchmark dataset for fake news detection. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 422–426). 10.18653/v1/P17-2067.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A. et al. (2019). Transformers: State-of-the-art natural language processing. arXiv.