big data and cognitive computing

MDPI

Article

Empowering Propaganda Detection in Resource-Restraint Languages: A Transformer-Based Framework for Classifying Hindi News Articles

Deptii Chaudhari [1,*] and Ambika Vishal Pawar [1,2,*]

[1] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune 412115, India
[2] Persistent University, Persistent Systems Limited, Ramanujan, Blue Ridge, Pune 4011057, India
* Correspondence: phdgrad.deepti.chaudhari@siu.edu.in (D.C.); ambika_pawar@persistent.com (A.V.P.)

Abstract: Misinformation, fake news, and various propaganda techniques are increasingly used in digital media. It becomes challenging to uncover propaganda as it works with the systematic goal of influencing other individuals for the determined ends. While significant research has been reported on propaganda identification and classification in resource-rich languages such as English, much less effort has been made in resource-deprived languages like Hindi. The spread of propaganda in the Hindi news media has induced our attempt to devise an approach for the propaganda categorization of Hindi news articles. The unavailability of the necessary language tools makes propaganda classification in Hindi more challenging. This study proposes the effective use of deep learning and transformer-based approaches for Hindi computational propaganda classification. To address the lack of pretrained word embeddings in Hindi, Hindi Word2vec embeddings were created using the H-Prop-News corpus for feature extraction. Subsequently, three deep learning models, i.e., CNN (convolutional neural network), LSTM (long short-term memory), Bi-LSTM (bidirectional long short-term memory); and four transformer-based models, i.e., multi-lingual BERT, Distil-BERT, Hindi-BERT, and Hindi-TPU-Electra, were experimented with. The experimental outcomes indicate that the multi-lingual BERT and Hindi-BERT models provide the best performance, with the highest F1 score of 84% on the test data. These results strongly support the efficacy of the proposed solution and indicate its appropriateness for propaganda classification.

Keywords: propaganda classification; Hindi text processing; news article analysis; natural language processing; deep learning; low-resource language processing; transformer-based models

check for updates

Citation: Chaudhari, D.; Pawar, A.V. Empowering Propaganda Detection in Resource-Restraint Languages: A Transformer-Based Framework for Classifying Hindi News Articles. *Big Data Cogn. Comput.* **2023**, *7*, 175. https://doi.org/10.3390/bdcc7040175

Academic Editors: Zuchao Li and Min Peng

Received: 27 September 2023
Revised: 3 November 2023
Accepted: 13 November 2023
Published: 15 November 2023

1. Introduction

The dictionary defines propaganda as “information, ideas, opinions, or images that give one part of an argument, which are broadcast, published to influence people’s opinions”. Propaganda is a well-studied sociological field. Propaganda is ponderously devised to influence the opinions and actions of people with some predefined goals. In 1937, social scientists, opinion leaders, historians, educators, and journalists founded an organization called the Institute for Propaganda Analysis (IPA). This organization was established to spread awareness among American citizens about political propaganda. IPA defined propaganda as “an expression of opinion or action by individuals or groups deliberately designed to influence opinions or actions of other individuals or groups with reference to predetermined ends”. One of the foundational classifications of propaganda techniques originated in the 1930s. The 1936 United States presidential campaign, including the reigning President Franklin D. Roosevelt and Alf Landon, garnered scholarly interest in the linguistic strategies employed by both candidates. In 1937, Clyde R. Miller put up a significant classification of propaganda. The set of seven techniques, initially introduced by the Institute for Propaganda Analysis in 1938, continues to be widely acknowledged and embraced in current discourse. The seven features identified by the Institute of Propaganda

Analysis indicating the use of propaganda are name-calling, glittering generalities, transfer, testimonials, plain folks, and card stacking [1].

In recent times, propaganda has been used by terrorist organizations for recruitment, by political parties during elections, and even by marketing agencies. Propaganda spreads through various techniques and is loaded with emotional appeals, falsification, and misinformation. Propaganda techniques are seen to be influencing the political discourse [2,3] and the spread of extremism [4–7]. During the COVID-19 pandemic, many misinformation and propaganda articles were also observed. According to [8], any biased message, intentional or unintentional, can be termed propaganda. The social ramifications of propaganda are manifold, ranging from the manipulation of elections [9,10] to health [11]. It is also observed that the propaganda phenomenon is not limited to specific demographic regions or languages.

In recent years, digital news media has grown tremendously and public opinion manipulation through propaganda has reached new levels. News articles in digital media have a propaganda continuum ranging from neutral to biased [1]. Every reader of a news piece should be conscious that it invariably reflects the bias of the news organization that published it, as well as the author, at least partially. It is challenging to determine the precise nature of prejudice, though. The author may be unaware of his/her prejudice. Alternatively, the article could be an instrument in the author's toolkit to convince readers of a particular viewpoint. Such a situation is an example of propaganda. The efficacy of propaganda is maximized when it operates covertly. If an individual engages with a journalistic piece, regardless of whether it is presented through a formal or informal news channel such as a blog or social media platform, it is expected that they would not readily discern it as propagandistic. Under such circumstances, the reader becomes unwittingly exposed to propagandistic content, potentially leading to a modification of the reader's viewpoints. In light of the diverse range of news sources available, comprising tabloids, print media, and digital platforms, as well as varying degrees of objectivity and bias, this study proposes that the development of an automated tool capable of identifying propagandistic fragments could prove advantageous for both news readers and news agencies. Propaganda spreads through various techniques and is loaded with emotional appeals, falsification, and misinformation. Such hidden techniques require extensive and in-depth analysis. To address this infodemic of disinformation, it is necessary to develop techniques for propaganda detection and analysis. Although the propaganda detection and analysis problem has caught the attention of researchers in recent years, major work has been reported in the English language. Less work in this area has been reported for low-resource languages [12,13]. In our effort to tackle propaganda detection, a framework for propaganda classification in Hindi was proposed. Hindi is India's primary language and the fourth most spoken language worldwide.

Propaganda detection research in English has produced excellent results, whereas research in Hindi propaganda is still promising. Hindi is a language short of various linguistic resources; hence, it is termed a resource-restraint language (RRL). Many Hindi news websites have come up in the last decade and have accepted Hindi as their content language, like amarujala.com, bhaskar.com, jagran.com, zeenews.india.com/hindi.com, lokmatnews.in, etc. Hindi has become a noteworthy web content language. It has also become vital to analyze Hindi news articles for propaganda and gain valuable insights and relevant information. The technological advances and abundant availability of online news articles in Hindi Devanagari script makes it a stimulating research area.

1.1. Background

The term "low-resource text processing" pertains to the utilization of natural language processing (NLP) methodologies, including text categorization, automated translators, sentiment detection, and information extraction, within contexts characterized by limited resources, such as insufficient training data, computational capacity, or domain-specific expertise. In low-resource situations, the availability of computational resources is often constrained, hence affecting the efficiency of running complex natural language processing (NLP) models. The utilization of resource-restraint languages presents unique challenges in domain-specific situations, such as propaganda news, characterized by highly specialized language. The field of low-resource text processing is currently a subject of extensive research due to its profound implications for enhancing the accessibility and utility of natural language processing (NLP) technology across various languages and domains.

The research in text mining for Asian low-resource languages has gained attention over the last decade. The availability of native language keyboards and one's affinity to indigenous languages has increased the use of various languages on the Web. With the advent of natural language processing techniques, researchers are increasingly addressing problems related to indigenous languages. However, Hindi is a resource-restrained language for which significantly less research has been conducted. The nuances of the Hindi language pose significant challenges to researchers dealing with Hindi text processing. Some of the challenges are as follows:

1. Lack of Hindi text corpora: Text corpora for the Hindi language are scarce and lack gold standards. The currently available resources are either in the developmental stage or lack authentication. Hence, obtaining a high quantity and quality annotated corpus is challenging.

2. Lack of language resources: Linguistic resources such as parsers, morphological analyzers, part-of-speech taggers, lexicons, and WordNets for Hindi are inadequate and need improvement.

3. Code-mixing: Often multi-lingual users use the mixing of two languages or code-mixing. Hinglish is the combination of Hindi and English and is seen prevalently while writing in Hindi digital platforms, for example, "स्टेट बैंक ऑफ इंडिया देश का सबसे बड़ा बैंक है। जो समय-समय पर अपने 40 करोड़ से ज्यादा अकाउंट होल्डर्स के लिए स्कीम लेकर आता रहता है। इस बार एसबीआई ने अपने योनो ऐप के थ्रू फ्री में इनकम टैक्स फाइलिंग करने की सुविधा दी है।" (State Bank of India is the largest bank in the country. This keeps coming up with schemes for its more-than-40-crore account holders from time to time. This time, the SBI has given the facility of filing income tax for free through its YONO app.) The above example was taken from the H-Prop-News dataset used in this study and shows the use of Hinglish in Hindi news articles. Generating a pure Hindi dataset is time-consuming and dreary, which remains a significant challenge.

4. Spelling and morphological variations: In Hindi, a lot of morphological information is infused in words that carry information such as gender, tense, and person. Dealing with these morphological variations is a major challenge. In addition, the same word can be spelled differently in Hindi, making it very complex to integrate these terms into lexical resources.

This research aims to address the historical lack of NLP technologies in Hindi language and particularly the subject of propaganda classification that has traditionally received limited attention.

1.2. Motivation

The propagation of propaganda techniques in mainstream media has inspired us to devise an approach to propaganda detection in Hindi news articles. From a technical perspective, there is varied accessibility to tools to take advantage of deep learning and transfer learning techniques. However, there are substantial limitations owing to the language of the analysis or the available datasets. It is often difficult to obtain satisfactory performance for different languages, usually low-resource languages, such as Hindi. Much of the earlier work on propaganda classification is centered on resource-rich languages such as English and, recently, Arabic. As per our knowledge, minimal work on propaganda detection in Hindi is reported, and hence this problem is worthy of research. The two substantial research gaps found in the literature survey are as follows:

- Limited analysis of techniques effective in striving against propaganda;
- Constrained research on identifying propaganda in Hindi news articles.

This research studies, analyzes, and proposes a deep learning and transformer-based approach to classify Hindi news articles with opposing polarities, such as propaganda and non-propaganda.

1.3. Contributions

To create a propaganda classification tool in Hindi, the Hindi computational propaganda dataset H-Prop-News [14] was used. The main objective of this study was to present a simple, yet compelling, solution for propaganda classification using deep learning and a transformer-based approach. The significant contributions of this study are as follows:

- Experiments with three deep learning models using Word2vec embeddings trained on our corpus;
- Experimentation with four variations of bidirectional encoder representation from transformer embeddings;
- Analysis of all models by evaluating them on test data;
- Fine-tuned multi-lingual BERT model for propaganda classification.

2. Related Work

The research community has recently shown an interest in exploring textual propaganda detection and classification. The rise of fake news has drawn interest in its use for propaganda. Prominent areas in which propaganda has been studied include terrorism and politics [1]. Various systems were developed for propaganda detection, such as Proppy [15], Prta [16], and PROpaganda Text dEteCTion (PROTECT) [17] for propaganda detection in news texts. In another study, true news was identified from a corpus of trusted satire, hoax, and propaganda (TSHP) [18] using n-gram features and a maximum-entropy classifier.

In their recent work, ref. [19] addressed the issue of propaganda detection in code-switched social media text. The authors created a corpus of English and Roman Urdu code-switched text and obtained the best results using a fine-tuned XLM-RoBERTa model. The authors [20] compared two approaches, i.e., BERT and SVM, to detect Pro-Kremlin propaganda in social media. The authors conclude that both the approaches turn out to be insufficient toward the automatically spawned news. The authors [21] have created a HQP dataset by manual annotations of 30,000 English tweets related to the Russia–Ukraine war. The authors indicate that the pretrained models may include biases in the downstream tasks and should be deployed with caution in practice. In their work, ref. [22] worked on bi-lingual dataset of tweets related to Smart City in Urdu and English. The authors have reported the best results using a fine-tuned RoBERTa model.

2.1. Deep Learning Models

Deep learning models have been effectively used for text classification tasks in the last few years. For the task of propaganda classification, acquiring significant features and representations of textual data is crucial. Deep learning models have proficiency in acquiring such intricate hierarchical representations of data. Deep learning methods can make use of pre-trained representations, such as word embeddings (e.g., Word2Vec, GloVe), to extract generic linguistic knowledge from massive text corpora. This pre-training considerably improves the model performance by providing them with a strong starting point for the propaganda text classification task. Context is vital in the propaganda classification task for resolving ambiguity, understanding nuances in language, and recalling prior words or sentences. Deep learning models can efficiently gather and leverage such context-sensitive data. Hence, deep learning models have shown promising results in propaganda classification. Convolutional networks have been widely used in various studies [3,23–25]. LSTM and Bi-LSTM models were used in [6,23,24,26]. In their work, Gavrilenko et al. [3] used CNN, LSTM, and Hierarchical-LSTM (H-LSTM) to identify propaganda in the text files from the corpus of the Internet Research Agency (IRA). They achieved 88.2% accuracy using the CNN model. Nizzoli et al. [6] worked on ISIS propaganda tweets using CNN and an RCNN, and reported an F1 score of 0.9. The authors of [23,24,26] performed sentence-level propaganda detection on a Proppy corpus using an ensemble of deep learning models. Hashemi and Hall [25] studied visual propaganda in images shared by violent extremist organizations (VEOs) on Twitter. They used an eight-layered deep CNN architecture, known as AlexNet, and reported an accuracy of 97.02%.

2.2. Transformer-Based Models

In recent years, transformer-based models have shown encouraging results in terms of text classification and natural language processing. In NLP4IF'19, the best-performing approach was reported using the BERT model with hyperparameter tuning [27]. Other studies have used variations of BERT-based models for propaganda classification tasks, such as the context-dependent BERT model [28] and cost-sensitive BERT [29]. Researchers have also utilized several variations of features for propaganda classification, such as EmoFeat emotion word features [25], linguistic, layout, and topical features [24], linguistic inquiry, and word count (LIWC) features [28], and linguistic style and complexity features [26]. The details of various deep learning and transfer learning models used for propaganda classification by various authors are listed in Table 1.

Table 1. Deep learning and transfer learning models used for propaganda classification.

Source	Models Used	Dataset	Techniques/Word Embeddings	Evaluation Parameters
[3]	CNN, LSTM, H-LSTM	Internet Research Agency (IRA) corpus	word2vec, GloVe	CNN + word2vec: Accuracy—0.877, F-measure—0.83
[6]	CNN, RCNN	ISIS propaganda tweets	character-based CNN, RCNN with pre-trained FastText word embeddings	RCNN: F1 score −0.9
[19]	Ensemble of BiLSTM, XGBoost, and BERT	Proppy corpus	Glove embedding	BERT-cased + BERT-uncased + BiLSTM + XGBoost: F1—0.67, Precision—0.62, Recall—0.73
[20]	CNN, LSTM-CRF, and BERT	Proppy corpus	Linguistic, layout, and topical features	CNN, LSTM-CRF, BERT Ensemble: F1—0.62, Precision—0.57 Recall-0.6819
[21]	CNN—AlexNet	Violent extremist organizations (VEOs) propaganda images collected from Twitter	Five convolutional layers and three fully connected layers	CNN AlexNet: Accuracy—97.02% F1—97.89%
[22]	BiLSTM, attention-based bidirectional LSTM	Proppy corpus	Dense word representations using ELMO models	ABL-Balanced-Ens: F1—59.5, Precision—50.9 Recall—71.6
[23]	BERT-based attention transformer model	Proppy corpus	Attention heads and transformer blocks with a softmax layer	BERT: Precision—60.1, Recall—66.5 and F1—63.2
[24]	BERT variations	Proppy corpus	BERT with logits	Precision—0.57, Recall—0.79, F1 score—0.66
[25]	BERT	Propaganda techniques corpus	Cost-sensitive BERT	Precision—0.56, Recall—0.70, F1 score—0.62

2.3. Propaganda Classification in Low-Resource Languages

Propaganda is a global phenomenon; however, most propaganda classification systems are centered on English, and little advancement has been seen in low-resource languages. The details of different propaganda related datasets used previously are as shown in Table 2. In the recent work by Kausar et al. [2], ProSOUL, a framework for propaganda detection in online Urdu content, was proposed. The authors used Urdu news article dataset and generated a linguistic inquiry and word count (LIWC) dictionary in Urdu language. The BERT model, along with NELA, word n-gram, and character n-gram features, outperformed it with an accuracy of 0.91. Recently, a significant contribution toward detecting propaganda techniques in Arabic was generated by the shared task in the WANLP workshop in 2022. The dataset developed by the authors [30] contained 3200 Arabic tweets from various news sources and was labeled with 20 propaganda techniques. Sameer et al. [31] obtained the best results using the AraBERT and Marefa-NER models. Mittal and Nakov [32] used XLM-R and a multigranularity network with an mBERT encoder, whereas [33] relied on AraBERT to show the best results. Other studies on the shared task [34–37] have also shown promising results.

Table 2. Propaganda-related datasets.

Source	Dataset Name	Language	Data Type	Source of Data	Classes/Labels
[18]	TSHP-17	English	News article	English gigaword corpus and other News websites	Trusted, satire, hoax, propaganda
[15]	QProp Corpus	English	News article	News sources published by MBFC (media bias/fact check)	Propagandist, non-propagandist
[16]	Proppy	English	News article	13 propaganda news outlets and 36 non-propaganda news outlets	18 propaganda technique labels
[12]	The propaganda benchmark dataset currently	Czech	News article	4 Czech digital news servers	15 propaganda technique labels
[13]	ProSOUL	Urdu	News article	Qprop [15] translated in Urdu	Propaganda, non-propaganda
[13]	Humkinar-Web	Urdu	Urdu content contents from a variety of domains including news, sports, health, religion, entertainment, and books, etc.	Urdu content on the WWW	Propaganda, non-propaganda
[13]	Humkinar-News	Urdu	News article	News webpages from 35 manually selected propaganda-free websites	Propaganda, non-propaganda
[27]	Arabic Propaganda Dataset (WANLP Shared Task)	Arabic	Tweets	Tweets from top news sources in Arabic countries	20 propaganda techniques
[14]	H-Prop	Hindi	News article	QProp corpus [15] translated in Hindi	Propaganda, non-propaganda
[14]	H-Prop-News	Hindi	News article	32 prominent Hindi News websites	Propaganda, non-propaganda

Some researchers translated the English news text dataset into low-resource languages to address resource scarcity. For example, Mapes et al. [27] translated English news into Bulgarian to create a news toxicity detector. In their work, Alam et al. [30] developed a fake news dataset by translating an English news text. Other studies have used Google Translate to generate the LIWC dictionary in Dutch [38] and Filipino [39]. A similar translation approach was used by Chaudhari et al. [14] to generate the H-Prop dataset in Hindi by translating the Proppy corpus. However, translation introduces errors that significantly affect classification tasks.

2.4. Propaganda-Related Datasets

Low-resource text processing is a major hurdle owing to the lack of standard and public datasets. TSHP-17 [18] is the earliest dataset seen to address propaganda classification tasks. In [18], propaganda was considered a class of fake news analysis. The QProp corpus in [15] contains 51,246 articles from news sources labeled using the distant supervision technique. The Proppy dataset [16] was collected from 13 propaganda news outlets and 36 non-propaganda news feeds. This dataset was labeled manually to obtain the text fragments from news articles using 18 propagandistic techniques. The dataset was further utilized for the MNLP shared task for fine-grained propaganda classification. In their work, S. Kausar et al. [13] created three datasets and a framework called ProSOUL for propaganda detection in Urdu. The first dataset was created by translating the QProp corpus into Urdu, and two other corpuses, Humkinar-Web and Humkinar-News, were obtained from the Urdu webpages and Urdu news articles. The Humkinar-Web and Humkinar-News datasets were labeled using the ProSOUL framework. An Arabic propaganda dataset was developed by [30] and released as part of the WANLP shared task for fine-grained propaganda classification in Arabic. This dataset contains 3200 tweets from leading Arabic news sources. Chaudhari et al. [14] created H-Prop and H-Prop-News datasets in their work. The H-Prop dataset was created by translating QProp. We can conclude from the previous work discussion that significant work has been carried out on propaganda detection and classification in English news texts and, recently, Arabic texts. However, little work has been conducted on propaganda classification tasks in low-resource languages. The propaganda classification research in Hindi language is limited as compared to English and Arabic. The peculiarities of Hindi and the limited linguistic resources may be the substantial reasons. Notably, no significant results have been reported for propaganda classification in Hindi other than in [14]. Taking advantage of previous work [14], this research proposes classifying propaganda news text in Hindi using deep learning and transformer-based models. In addition, Word2vec embeddings developed using Hindi content were utilized.

3. Materials and Methods

3.1. Problem Statement

Given a paragraph of a news article, the system performs binary classification to label the news article as propagandist or not. For simplicity, the propaganda classification can be considered a text classification problem where we map the input sequence of tokens T in news articles to one of the n labels. First, a composition function f was applied to the sequence of tokens to create word embeddings vw such that $w \in T$. The output of this function f is the vector v that serves as input to deep learning models. The sigmoid activation function in the output layer generates estimated probabilities for the output label y.

$$\boldsymbol{y} \ = \ sigmoid\ (vw) = \frac{1}{1 + e^{vw}}$$

The transformer-based models have encoder with multi-head attention capable of learning multiple representation features from the given text. Based on the pretrained parameters, the proposed models were fine-tuned to Hindi news text. The output probability of the models can be calculated as follows:

$$y = \boldsymbol{W}\,\boldsymbol{V} + b$$

$$P_x = \frac{e^{yx}}{\sum_{i=1}^{n} e^{yi}}$$

where V is the classification vector output by the transformer models W and b, i.e., weights and biases. Px is the output probability. Furthermore, the classification loss can be calculated using the cross-entropy function as defined below.

$$Cross\ Entropy(y,\ P)\ =\ -\sum_{i=1}^{N} yi\ log(Pi) = -log(Px)$$

The proposed approach of this study encompasses three main steps as preprocessing, word embedding generation, and classification using deep learning and transformer-based models. Figure 1 shows an overview of the steps involved in the methodology. The details of the dataset, preprocessing, and model architectures are elaborated in the following sections.

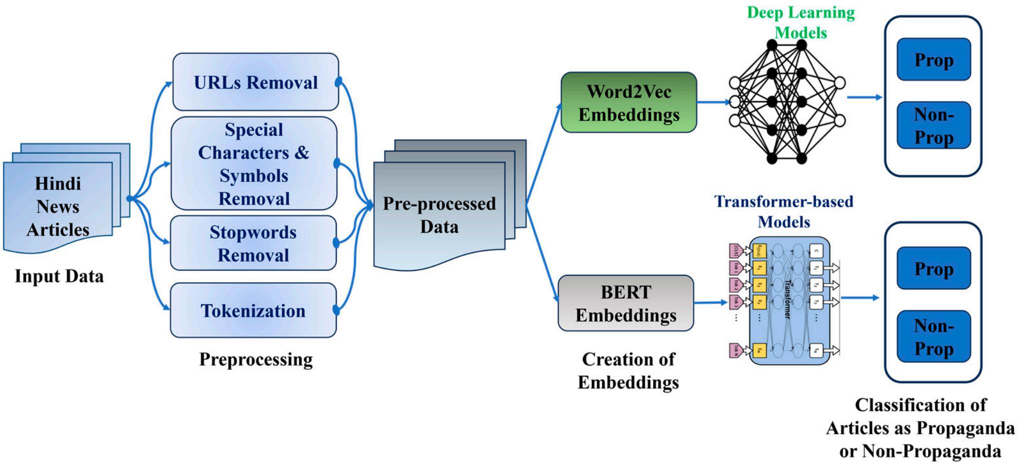


Figure 1. Overview of the methodology.

3.2. Dataset

For this study, our previously introduced Hindi news article dataset H-Prop-News was used. This dataset was generated by collecting news articles from notable Hindi news websites. The dataset was released for the binary propaganda classification task. Each news article was labeled propaganda (1) or non-propaganda (−1). The dataset contains 5500 Hindi news articles, the news website source, URL, and headlines. The H-Prop News dataset is available in three partitions: development, training, and testing. The details of the H-Prop-News dataset are shown in Table 3.

Table 3. Details of H-Prop news dataset.

Data Partition	Propaganda Articles	Non-Propaganda Articles
Development	275	275
Training	1850	2000
Testing	505	595
Total	2630	2870

The H-Prop-News dataset was analyzed to understand the distribution of news article sources. Figure 2 shows the news source-wise article distribution in the H-Prop-News dataset. The dataset contained a maximum of 604 articles from Amar Ujala (amarujala.com) and the fewest from India TV. The category-wise article distribution shown in Figure 3 indicates that most propaganda articles were sourced from patrika.com, and most non-propaganda articles come from amarujala.com. A word cloud of the most frequently appearing keywords, as shown in Figure 4, was created to find the top words appearing in the dataset.

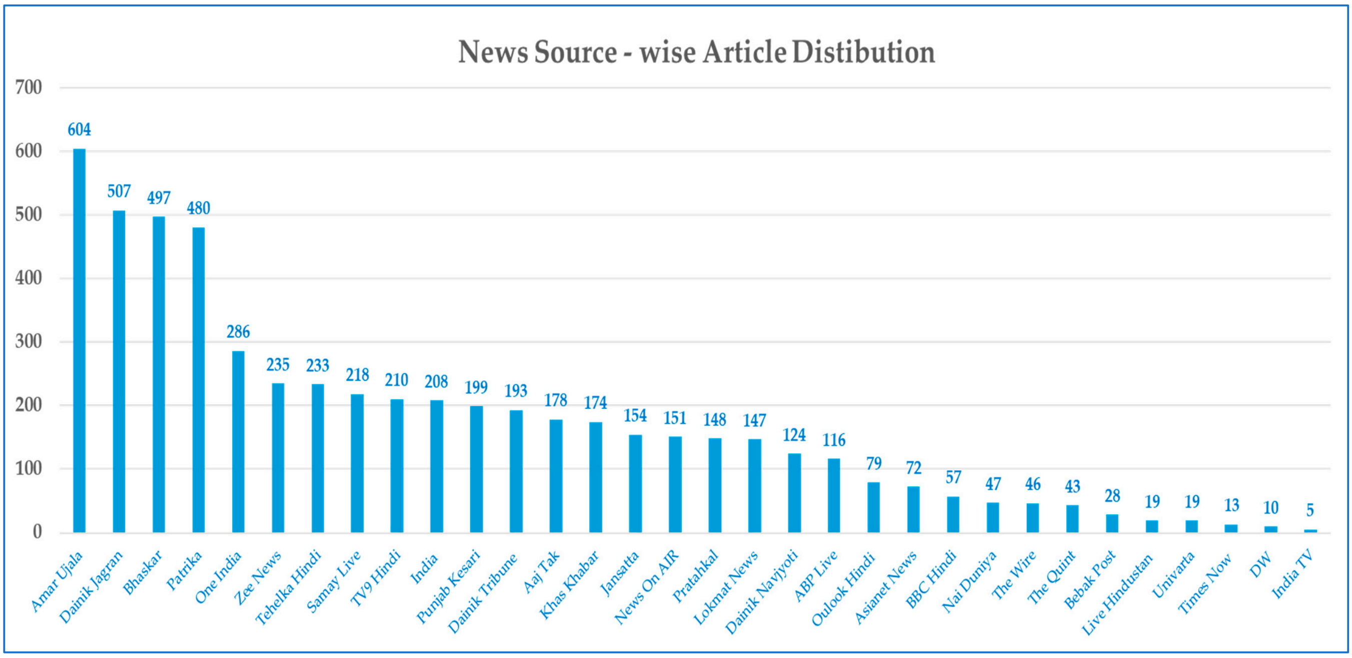


Figure 2. News source-wise article distribution in the H-Prop-News dataset.

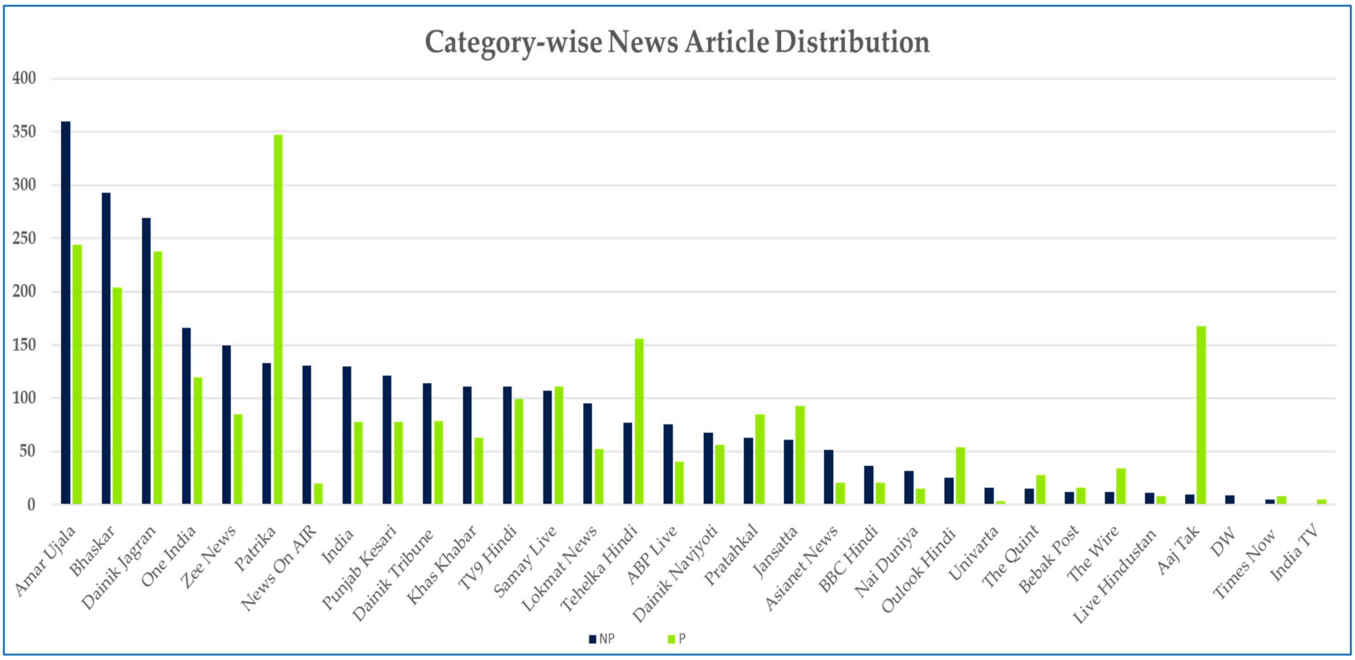


Figure 3. Category-wise news articles distribution in the H-Prop-News dataset.

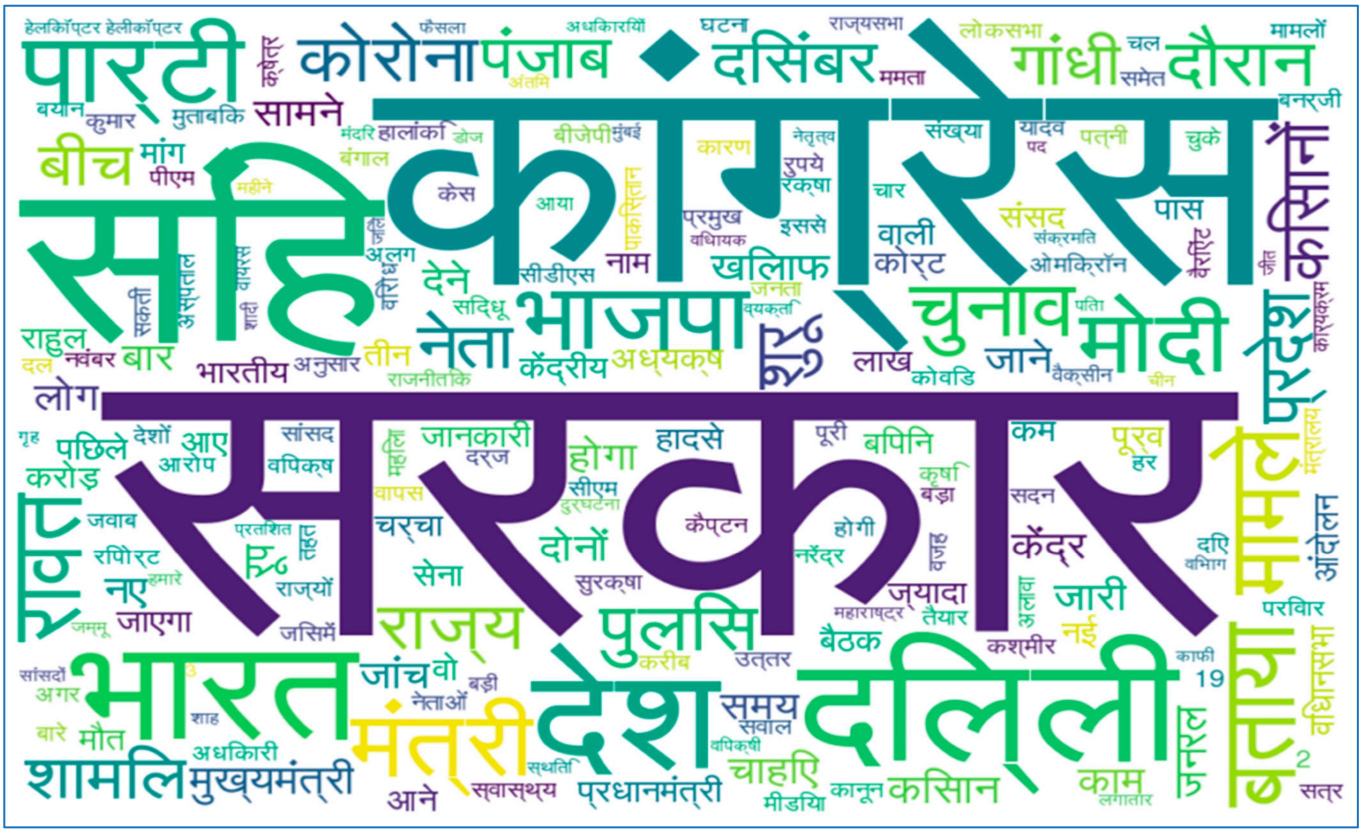


Figure 4. Word cloud of most frequent words appearing in the dataset.

3.3. Data Preprocessing

The news articles in the corpus need the following preprocessing steps before further utilization.

1. **URL and mentions removal:** Some news articles contain URLs and tweet mentions. These were identified using regular expressions and replaced by a space. The URLs were removed using regular expression patterns like "http://" or "https://". A simple regular expression pattern "@\w+" was used to find the mentions. This expression identifies tweet mentions by finding the strings starting with "@" followed by one or more characters.
2. **Stop word and non-Hindi word removal:** Stop words are frequently occurring words in the text that do not hold any meaning. Some examples of Hindi stop words are "और" (and), "की" (of), "है" (is), and so on. A predefined list of Hindi stop words was used to remove the stop words from the text. Removing these words helps in reducing the noise and focusing on the content-carrying words. To remove non-Hindi words, a language detection library, langdetect, was used and the words belonging to English were removed. Additionally, all punctuation marks were removed.
3. **Tokenization:** Tokenization was performed using the indicnlp tokenizer, which is a specialized tokenizer for Indian languages. It is tailored to handle the linguistic characteristics and requirements of languages spoken in India, which may differ from tokenizers used for languages with different linguistic structures. The "indicnlp tokenizer" tokenizes text based on punctuation boundaries. This means it breaks text into tokens wherever it finds punctuation marks like periods, commas, question marks, exclamation marks, and so on. Punctuation marks act as usual boundaries between words or subword units in Hindi. Tokenization based on punctuation boundaries is a direct approach and works well for Hindi.

3.4. Hindi Word Embedding Creation

Natural language data are typically unstructured, and deep learning models cannot handle them in their raw format. Hence, natural language data must be converted into internal vector representations. To create word embeddings, Word2vec, an unsupervised algorithm for learning the vector representations of words, was used. The Word2vec model transformed the words into vectors by placing the words with a shared context in vicinity of the vector space [40]. Taking a large corpus as the input, the Word2vec algorithm creates a vector space and assigns a distinctive vector value to each word in the vector space.

Currently, the primarily available Word2vec embeddings are in English. Custom Word2vec embeddings were created which are trained on our corpus of all the news articles to obtain the desired embeddings. After preprocessing the data, the entire dataset was divided into sentences and obtained a vector representation of Hindi words using Gensim [37]. To create the model, a dimension of 300 and skip-gram window of 10 were chosen. The corpus had 43,423 raw words with 10,310 word types. After retaining the unique words and down-sampling most common words, a corpus of 29,933 words was obtained. The trained Word2Vec model on the corpus produced the vector of dimensions 10,310 × 300. The vocabulary of all unique words was built, and the maximum length of the news text was found. Each news text was padded to a maximum length of 1725 words. Using the trained Word2vec model, an embedding matrix was created, where the index of each word in the vocabulary was its index in the matrix, and the corresponding entry was its Word2vec representation. The generated word embeddings served as an input to all deep learning models to predict the news text labels.

3.5. Deep Learning Models

As seen in related work performed by other researchers, the deep learning models showed good results in the propaganda classification task in other languages. Hence, this research proposes three deep learning-based models for the computational propaganda classification task for the Hindi language. The models presented are convolutional neural networks (CNNs), long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM). These deep learning models were selected based on the following points:

- Convolutional neural networks (CNNs) perform well at identifying local characteristics in text, which is useful for tasks like classifying propaganda. CNNs also have the capability to detect patterns in limited text segments and can offer computational advantages for smaller datasets and less intricate natural language processing (NLP) assignments.
- Long short-term memory (LSTM) networks are specifically built to effectively process sequential data, enabling them to capture and represent long-range dependencies within textual information. These models have the capability to handle the input sequences of varying lengths, a characteristic frequently seen in news articles. Long short-term memory (LSTM) models have demonstrated a notable ability to properly preserve context and conversation history, which is required for the propaganda classification task.
- Bidirectional long short-term memory (Bi-LSTM) models are capable of capturing contextual information from both preceding and succeeding tokens within a sequence. This characteristic is of utmost significance, as it enables a comprehensive analysis of the complete context surrounding a given word or phrase. The inclusion of contextual information can be advantageous in the task of classifying propaganda. Bi-LSTM models also have the capability to construct sentence embeddings by effectively encoding information from both forward and backward directions.

3.5.1. Convolutional Neural Network (CNN)

The adopted CNN architecture is shown in Figure 5. This architecture is inspired by the work described in [41]. In our case, the embedding layer serves as an input to the model. As explained in [41], classification quality can be affected by the filter size configuration.

Four filter sizes, 2, 3, 4, and 5, were used, which allowed us to focus on smaller and larger sections of the news text. After filtering, a max-pooling layer was applied with a dropout probability of 0.2. The maximum values obtained from each convolutional layer were concatenated to obtain a single vector. This single vector was then processed using a fully connected layer of size 30 and dropout probability of 0.2. The dropout layers help achieve better convergence and avoid overfitting.

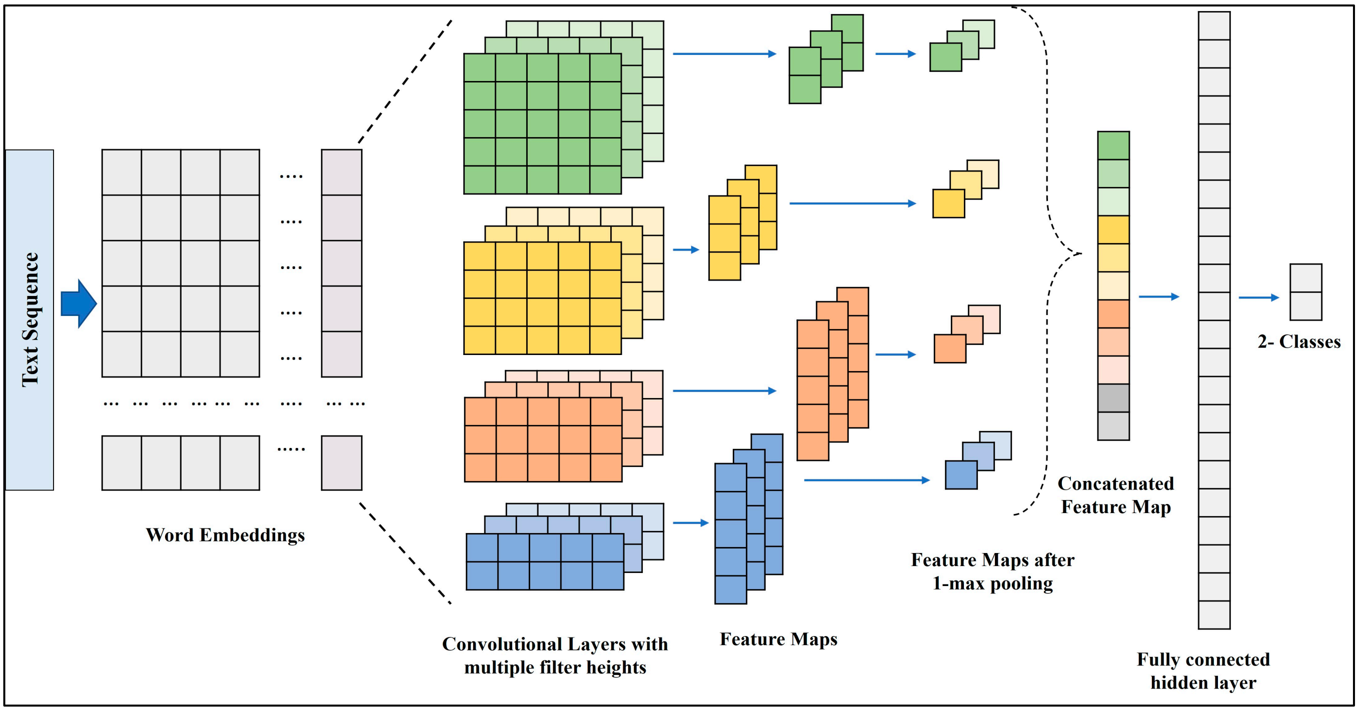

Text Sequence
Word Embeddings
Convolutional Layers with multiple filter heights
Feature Maps
Feature Maps after 1-max pooling
Concatenated Feature Map
Fully connected hidden layer
2- Classes

Figure 5. CNN architecture.

3.5.2. Long Short-Term Memory (LSTM)

The context of a word can be determined by using the words preceding it. It has been proven that LSTMs are capable of capturing the relevant context of the word by using memory cells in the network. These memory cells record the meaning of previously occurring words. To model the contextual information, an LSTM model was constructed. As explained earlier, an LSTM classifier was implemented that used the Word2vec embeddings obtained. In the model architecture, the embedding layer is followed by the LSTM layer and then the fully connected layer. The last layer with two neurons is responsible for the news text classification. Also, a dropout layer with a probability of 0.5 was used to counterbalance the overfitting. In the output layer, a sigmoid activation function was used because the classification problem was binary.

3.5.3. Bidirectional Long Short-Term Memory (BILSTM)

Unlike a unidirectional LSTM, which relies only on past words in the sequence, a bidirectional LSTM leverages the context of the word's past and future sequences. In bidirectional LSTM, the memory cells are present in both directions to preserve the information of the words surrounding a particular word. Our bidirectional LSTM token embeddings were fed as inputs to the input layer. A rectified linear unit (ReLU) in the hidden layers was used as an activation function. The average pool and max pool in the pooling layers to reduce the data dimensions were used. The probability of 0.1 was used in the dropout layer. The fully connected layer was the last layer of the model and provided the output for the two classes.

3.6. Transformer-Based Models

Although convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and bidirectional LSTMs have their respective advantages, the emergence of transformer-based models such as BERT and its variations has resulted in notable enhancements across multiple natural language processing (NLP) tasks. Transformers have demonstrated exceptional proficiency in capturing intricate interdependencies within textual data, hence establishing themselves as the prevailing preference for numerous natural language processing (NLP) applications. Owing to their unparalleled performance, this research work focused on fine-tuning the mono-lingual and multi-lingual transformer-based models. The general architecture of transformer-based models is as shown in Figure 6.

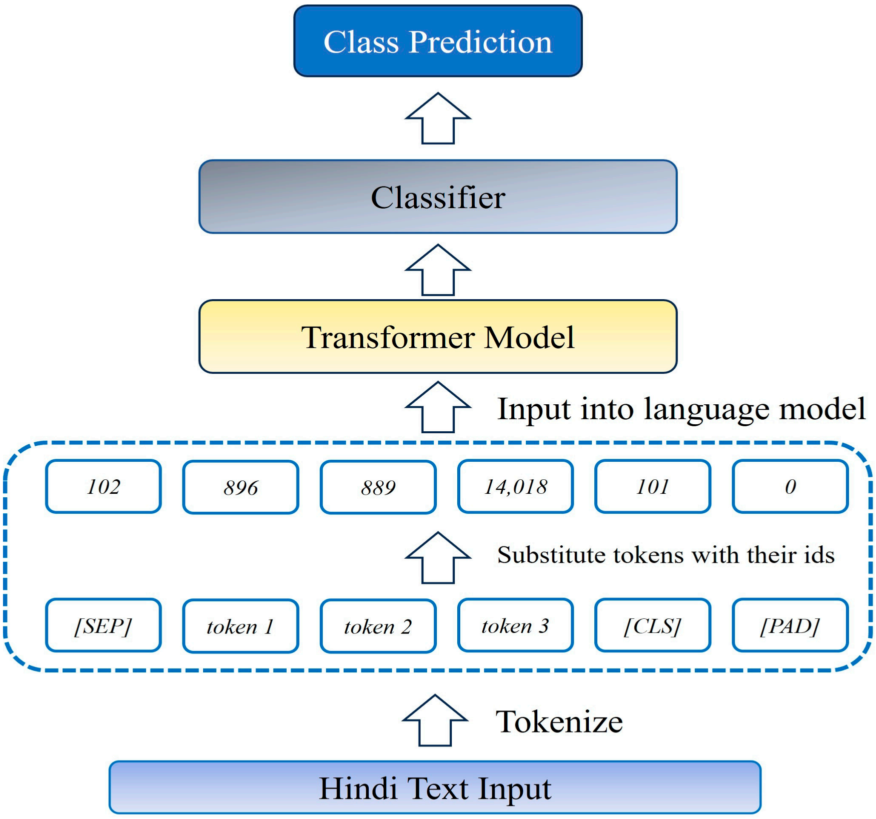


Figure 6. General architecture of transformer-based models.

The news article input text was tokenized into a list of tokens using a model-specific tokenizer. For example, the sentence “आप कैसे है” (How are you) will result in six tokens using the mBERT tokenizer as [‘आ’, ‘##प’, ‘क’, ‘##ै’, ‘##से’, ‘है’]. The pre-trained language models use fixed-length inputs. Hence, paddings were added, or extra tokens were removed if required to match the sequence length. Figure 7 shows the distribution of the token counts in the dataset. Padded tokens were not forwarded during the training phase. Hence, the padded tokens were masked. After padding and masking, the generated tokens were inputted into the transformer models and passed through multiple self-attention layers. For each token, a final hidden embedding was generated. The first token related to the beginning of the sequence was considered to generate the probability distribution over the two classes. Our study used four models: multilingual BERT, distill-multi-lingual BERT, Hindi BERT, and Hindi-TPU-Electra.

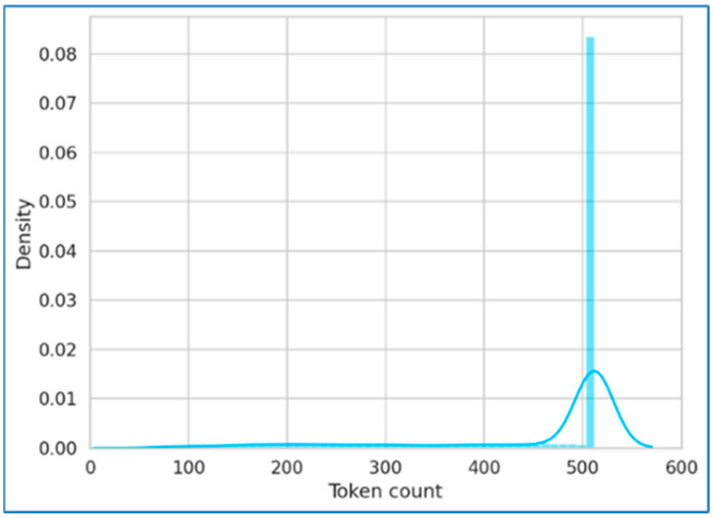


Figure 7. Distribution of token counts.

3.6.1. Multi-Lingual Bert (Bert-Base-Multi-Lingual-Cased)

The multi-lingual BERT is a multi-lingual variant of BERT [42] trained on 104 languages using a Wikipedia corpus. The number of entries for the different languages in the Wikipedia corpus was different. To have a sufficient number of words in the vocabulary, high-resource languages such as English were under-sampled, and low-resource languages were over-sampled. An exponentially smoothed weighing factor was used for the sampling.

3.6.2. Distil-Multi-lingual Bert (Distilbert-Base-Multi-Lingual-Cased)

The DistilBERT base multi-lingual model [43] is the distilled version of the BERT-based multi-lingual model. Compared with the multi-lingual BERT base model, the distil-multi-lingual BERT model has only 6 layers and 768 dimensions, resulting in 133 M parameters. In contrast, the mBERT model produces 177 M parameters. This makes the model twice as fast as the mBERT base model.

3.6.3. Hindi Bert and Hindi-Tpu-Electra

Hindi-BERT was trained on Google Research's ELECTRA [44] in the Hindi language. The model was trained using the Hindi CommonCrawl and Hindi Wikipedia corpora. Hindi-TPU-Electra is a pre-trained Hindi language model trained with an Electra base size. These models were trained using ktrain and TensorFlow to achieve better accuracy. Both models were fine-tuned for our propaganda classification task.

4. Experimental Settings

All the experiments were performed on an Nvidia DGX-Server with four Nvidia Tesla V-100 GPUs and 32 GB of memory. The details of the environmental setup for experimentation is as mentioned in Table 4. For the deep learning models, we split the dataset into 80% and 20% of examples in the training and testing sets, respectively. As explained in Section 4, we generated the word embeddings for our dataset using the Word2vec skip-gram approach. For all deep learning model experiments, we experimented with loss functions, activation functions, and optimizers to obtain the best results. The ReLU activation function was employed for all layers except the output layer. The binary cross-entropy loss function and Adam optimizer were used. The models were trained for 50 epochs early stopping on patience 10.

This research used Hugging Face transformers for transformer-based models, a Python library that provides pre-trained models for research and end-users. Pytorch, TensorFlow, and ktrain were used to train the models described in Section 4. The pre-trained models such as BERT-base-multi-lingual-cased, distil-mBERT, Hindi-BERT, and Hindi-TPU-Electra were fine-tuned to our dataset and for the propaganda classification task. The AdamW optimizer, cross-entropy loss function, a learning rate 2 × 10^{-5}, batch size of 16, and 15 epochs were used.

Table 4. Environment setup for experimentation.

Environment		Configuration Details
Hardware	GPU	Nvidia DGX-Server with 4 Nvidia Tesla V-100 GPUs
	CPU	Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz 1.99 GHz
	Memory	20 GB
Software	Operating System	Windows 10
	Programming Environment	Anaconda, Google Colab
	Libraries	TensorFlow, Keras, PyTorch, Scikit-Learn, nltk, GenSim

5. Results and Analysis

This section summarizes and compares the results obtained using deep learning and transformer-based models. The evaluation metrics and analysis of the obtained results are explained in the subsections.

5.1. Evaluation Metrics

The models were evaluated based on evaluation metrics such as precision, recall, F1 score, and accuracy. The evaluation metrics can be stated as follows:

$$Precision = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + False\ Positives\ (FP)}$$

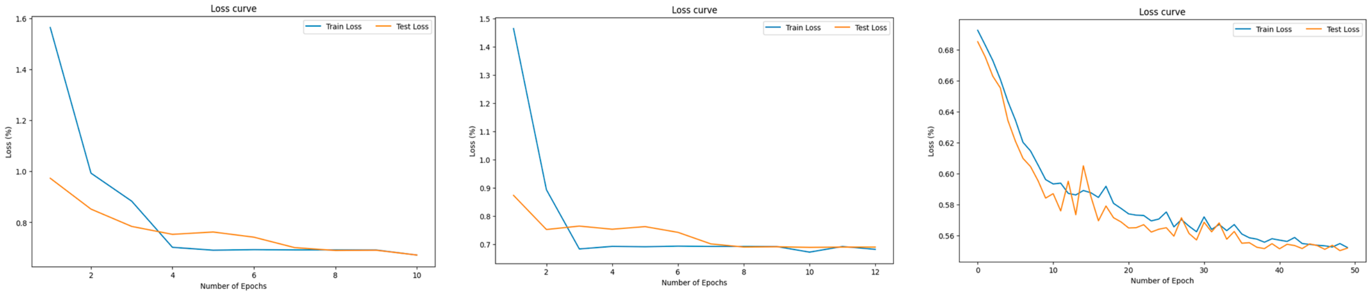
$$Recall = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + False\ Negatives}$$

$$F1 - Score = 2 \times \frac{\text{Precisionx Recall}}{\text{Precision} + \text{Recall}}$$

All the evaluation metrics can be macro-averaged or weight-averaged for calculating the values across different classes. As our classification problem is binary, to treat the classes equally, the macro-averaged scores were employed for precision, recall, and the F score.

5.2. Model Performance

The performance of deep learning architectures is shown in Table 5. To avoid overfitting, an early stopping with patience of 10 was used. The best results were recorded after 11, 12, and 50 epochs for the CNN, LSTM, and BiLSTM models, respectively. The validation loss plots are shown in Figure 8. Model loss plots play a vital role in the training and evaluation processes of deep learning models. The loss curves of deep learning models were developed for the purpose of monitoring and analyzing the training process, as well as evaluating the performance of the models. The observed loss curves throughout the epochs demonstrate a favorable convergence of the models, indicating an absence of overfitting. The BiLSTM loss plot indicates the best model performance without overfitting. The macro-averaged metrics for all the models were considered and the best accuracy, precision, recall, and F1 score were obtained using the BiLSTM model.



(**a**) (**b**) (**c**)

Figure 8. Deep learning model loss plots. (**a**) Loss curve—CNN; (**b**) Loss curve—LSTM; (**c**) Loss curve—BiLSTM.

Table 5. Results of the deep learning models.

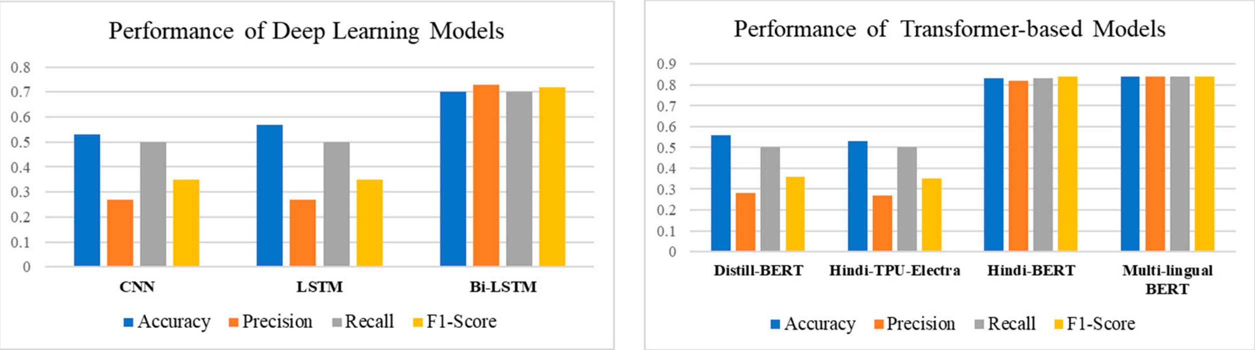
Model	Accuracy	Precision	Recall	F1 Score
CNN	0.53	0.27	0.50	0.35
LSTM	0.57	0.27	0.50	0.35
Bi-LSTM	0.70	0.73	0.70	0.72

Table 6 presents the results for the transformer-based models. This work explored the multi-lingual and mono-lingual BERT models. It was observed that the transfer learning approach yielded better results in smaller epochs. However, more time was required to train these models. By examining the results, it can be seen that the multi-lingual BERT outperformed the other models during testing. The distill-BERT model is a distilled version of the mBERT, whereas the Hindi-TPU-Electra is a larger version of the Hindi-BERT model. The multi-lingual BERT model achieved an 84% score on our main evaluation measure (macro-averaged F1 score), which is just 1% higher than the second best score obtained via Hindi-BERT. Both models were trained with large Hindi texts and hence showed the best performance for our task of propaganda classification on Hindi texts.

Table 6. Results of the transformer-based models.

Model	Accuracy	Precision	Recall	F1 Score
Multi-lingual BERT	0.84	0.84	0.84	0.84
Distill-BERT	0.56	0.28	0.50	0.36
Hindi-BERT	0.83	0.82	0.83	0.84
Hindi-TPU-Electra	0.53	0.27	0.5	0.35

Figure 9 shows the comparative analysis of deep learning and transformer-based models. From the comparative analysis across all evaluation metrics, it is evident that among the deep learning models, the Bi-LSTM model performed the best, and among transformer-based models, the multi-lingual BERT showed the best results. After analyzing the obtained results, it can be concluded that the classification performances of the multi-lingual BERT and Hindi-BERT models were almost on par with each other. Achieving an 84% accuracy, Bi-LSTM models can capture contextual information, thus showing superior results to the CNN. Multi-lingual BERT performed slightly more effectively than mono-lingual Hindi-BERT.



(**a**) (**b**)

Figure 9. Performance analysis of deep learning and transformer-based models. (**a**) Performance of deep learning models. (**b**) Performance of transformer-based models.

6. Conclusions

The propagation of propaganda techniques in mainstream media has inspired us to devise an approach to propaganda detection in Hindi news articles. From a technical perspective, there is varied accessibility to tools to take advantage of deep learning and transfer learning techniques. However, there are substantial limitations owing to the language of the analysis or the available datasets. It is often difficult to obtain satisfactory performance for different languages, usually low-resource languages, such as Hindi. Much of the earlier work on propaganda classification is centered on resource-rich languages such as English and, recently, Arabic. As per our knowledge, minimal work in propaganda detection in Hindi is reported, and hence this problem is worthy of research. In this study, we investigated deep learning and transformer-based models for the Hindi computational propaganda classification task. The H-Prop-News dataset of Hindi propaganda and non-propaganda news text is used for the classification task. Owing to the lack of pre-trained word embeddings for Hindi, we trained a Word2vec model for word representations. These word embeddings are used as inputs to all the deep learning models. The loss curves for the deep learning models also indicate that the models are converging without overfitting. The comparative analysis of all experimented models indicates that the Bi-LSTM and multi-lingual BERT model performance is best across all performance metrics. After analyzing the obtained results, we concluded that the classification performances of the Bi-LSTM and multi-lingual BERT models were almost on par with each other, achieving approximately 84% accuracy. The multi-lingual BERT model achieved an 84% score of macro-averaged F1 score, which is just 1% higher than the second best score obtained via Hindi-BERT. The Bi-LSTM model can capture contextual information, thus showing superior results to the CNN. Multi-lingual BERT performed slightly more effectively than mono-lingual Hindi-BERT did. This study highlights the significance of contextual factors in the classification of propaganda in the Hindi language. A minimal work on low-resource language such as Hindi was reported for propaganda detection and this research contributed to the field by proposing an approach to address this.

Limitations and Future Work

As with other works, ours is not free of limitations. One of the limitations of this research is the absence of linguistic features which will be addressed in the future work. This work also does not consider data augmentation, and this can be explored in future work. We intend to extend our work to the detection of fine-grained propaganda. We also aim to further fine-tune the BERT model and improve it using the semantic features of Hindi texts.

Our work will encourage and help further research on low-resource languages. Our work can be extended to study other Hindi texts available on the web such as blogs, social media posts, and also other regional Indian languages. Our proposed solution is also deployed as a web application, thereby providing a pedestal for future research. In the future, we plan to investigate the impact of different word representations on propaganda classification tasks.

Author Contributions: Conceptualization, D.C. and A.V.P.; methodology, D.C.; software, D.C.; validation, D.C. and A.V.P.; formal analysis, D.C.; investigation, D.C.; resources, A.V.P.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, A.V.P.; visualization, D.C.; supervision, A.V.P.; project administration, A.V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available at https://zenodo.org/records/5828240 (accessed on 26 September 2023).

Acknowledgments: The authors would like to express their gratitude to the Symbiosis Institute of Technology (SIT) and Symbiosis International University (SIU) for their support and assistance with this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chaudhari, D.D.; Pawar, A.V. Propaganda analysis in social media: A bibliometric review. *Inf. Discov. Deliv.* **2021**, *49*, 57–70. [CrossRef]
2. Kellner, A.; Rangosch, L.; Wressnegger, C.; Rieck, K. *Political Elections Under (Social) Fire? Analysis and Detection of Propaganda on Twitter*; Technische Universität Braunschweig: Braunschweig, Germany, 2019; pp. 1–20. Available online: http://arxiv.org/abs/1912.04143 (accessed on 22 June 2020).
3. Gavrilenko, O.; Oliinyk, Y.; Khanko, H. *Analysis of Propaganda Elements Detecting Algorithms in Text Data*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 938.
4. Heidarysafa, M.; Kowsari, K.; Odukoya, T.; Potter, P.; Barnes, L.E.; Brown, D.E. Women in ISIS Propaganda: A Natural Language Processing Analysis of Topics and Emotions in a Comparison with Mainstream Religious Group. 2019. Available online: http://arxiv.org/abs/1912.03804 (accessed on 21 June 2020).
5. Johnston, A.H.; Weiss, G.M. Identifying sunni extremist propaganda with deep learning. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence, Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–6. [CrossRef]
6. Nizzoli, L.; Avvenuti, M.; Cresci, S.; Tesconi, M. Extremist propaganda tweet classification with deep learning in realistic scenarios. In Proceedings of the WebSci 2019—Proceedings of the 11th ACM Conference on Web Science, Boston, MA, USA, 30 June–3 July 2019; pp. 203–204. [CrossRef]
7. Relations, E. *An Analysis of Online Terrorist Recruiting and Propaganda Strategies*; E International Relations: Bristol, UK, 2017; pp. 1–11.
8. Ellul, J. Propaganda: The Formation Of Men's Attitudes By Jacques Ellul. In *United States: Vintage Books*; Vintage: Tokyo, Japan, 1965.
9. Stukal, D.; Sanovich, S.; Tucker, J.A.; Bonneau, R. For Whom the Bot Tolls: A Neural Networks Approach to Measuring Political Orientation of Twitter Bots in Russia. *SAGE Open* **2019**, *9*, 2158244019827715. [CrossRef]
10. Beğenilmiş, E.; Uskudarli, S. Supervised Learning Methods in Classifying Organized Behavior in Tweet Collections. *Int. J. Artif. Intell. Tools* **2019**, *28*, 1960001. [CrossRef]
11. Ahmed, W.; Seguí, F.L.; Vidal-Alaball, J.; Katz, M.S. COVID-19 and the 'Film Your Hospital' conspiracy theory: Social network analysis of Twitter data. *J. Med. Internet Res.* **2020**, *22*, e22374. [CrossRef] [PubMed]
12. Baisa, V.; Herman, O.; Horák, A. Benchmark dataset for propaganda detection in Czech newspaper texts. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 2–4 September 2019; pp. 77–83. [CrossRef]
13. Kausar, S.; Tahir, B.; Mehmood, M.A. Prosoul: A framework to identify propaganda from online urdu content. *IEEE Access* **2020**, *8*, 186039–186054. [CrossRef]
14. Chaudhari, D.; Pawar, A.V.; Cedeño, A.B. H-Prop and H-Prop-News: Computational Propaganda Datasets in Hindi. *Data* **2022**, *7*, 29. [CrossRef]

15. Barrón-Cedeño, A.; Jaradat, I.; Da San Martino, G.; Nakov, P. Proppy: Organizing the news based on their propagandistic content. *Inf. Process. Manag.* **2019**, *56*, 1849–1864. [CrossRef]
16. da San Martino, G.; Yu, S.; Barrón-Cedeño, A.; Petrov, R.; Nakov, P. Fine-grained analysis of propaganda in news articles. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5636–5646.
17. Vorakitphan, V.; Cabrio, E.; Villata, S. PROTECT—A Pipeline for Propaganda Detection and Classification To cite this version: HAL Id: Hal-03417019 A Pipeline for Propaganda Detection and Classification. In Proceedings of the CLiC-it 2021-Italian Conference on Computational Linguistics, Milan, Italy, 26–28 January 2022.
18. HRashkin; Choi, E.; Jang, J.Y.; Volkova, S.; Choi, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2931–2937. [CrossRef]
19. Salman, M.U.; Hanif, A.; Shehata, S.; Nakov, P. Detecting Propaganda Techniques in Code-Switched Social Media Text. 2023. Available online: http://arxiv.org/abs/2305.14534 (accessed on 27 September 2023).
20. Solopova, V.; Popescu, O.-I.; Benzmüller, C.; Landgraf, T. Automated Multilingual Detection of Pro-Kremlin Propaganda in Newspapers and Telegram Posts. *Datenbank-Spektrum* **2023**, *23*, 5–14. [CrossRef]
21. Maarouf, A.; Bär, D.; Geissler, D.; Feuerriegel, S. HQP: A Human-Annotated Dataset for Detecting Online Propaganda. No. Mcml. 2023. Available online: https://arxiv.org/abs/2304.14931v1 (accessed on 27 September 2023).
22. Ahmad, P.N.; Khan, K. Propaganda Detection And Challenges Managing Smart Cities Information On Social Media. *EAI Endorsed Trans. Smart Cities* **2023**, *7*, e2. [CrossRef]
23. Al-Omari, H.; Abdullah, M.; AlTiti, O.; Shaikh, S. JUSTDeep at NLP4IF 2019 Task 1: Propaganda Detection using Ensemble Deep Learning Models. In Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Hong Kong, China, 3 November 2019; pp. 113–118. [CrossRef]
24. Gupta, P.; Saxena, K.; Yaseen, U.; Runkler, T.; Schütze, H. Neural Architectures for Fine-Grained Propaganda Detection in News. *arXiv* **2019**, arXiv:1909.06162. [CrossRef]
25. Hashemi, M.; Hall, M. Detecting and classifying online dark visual propaganda. *Image Vis. Comput.* **2019**, *89*, 95–105. [CrossRef]
26. Cruz, A.F.; Rocha, G.; Cardoso, H.L. On Sentence Representations for Propaganda Detection: From Handcrafted Features to Word Embeddings. In Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Hong Kong, China, 3 November 2019; pp. 107–112. [CrossRef]
27. Mapes, N.; White, A.; Medury, R.; Dua, S. Divisive Language and Propaganda Detection using Multi-head Attention Transformers with Deep Learning BERT-based Language Models for Binary Classification. In Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Hong Kong, China, 3 November 2019; pp. 103–106. [CrossRef]
28. Alhindi, T.; Pfeiffer, J.; Muresan, S. Fine-Tuned Neural Models for Propaganda Detection at the Sentence and Fragment levels. *arXiv* **2019**, arXiv:1910.09702. [CrossRef]
29. Madabushi, H.T.; Kochkina, E.; Castelle, M. Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data. *arXiv* **2020**, arXiv:2003.11563. [CrossRef]
30. Firoj, P.N.A.; Mubarak, H.; Wajdi, Z.; Martino, G.D.S. Overview of the WANLP 2022 Shared Task on Propaganda Detection in Arabic. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; Available online: https://aclanthology.org/2022.wanlp-1.11 (accessed on 27 September 2023).
31. Samir, A. NGU_CNLP at WANLP 2022 Shared Task: Propaganda Detection in Arabic. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 545–550.
32. Mittal, S.; Nakov, P. IITD at WANLP 2022 Shared Task: Multilingual Multi-Granularity Network for Propaganda Detection. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 529–533. Available online: https://aclanthology.org/2022.wanlp-1.63 (accessed on 27 September 2023).
33. Laskar, S.R.; Singh, R.; Khilji, A.F.U.R.; Manna, R.; Pakray, P.; Bandyopadhyay, S. CNLP-NITS-PP at WANLP 2022 Shared Task: Propaganda Detection in Arabic using Data Augmentation and AraBERT Pre-trained Model. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 541–544. Available online: https://aclanthology.org/2022.wanlp-1.65 (accessed on 27 September 2023).
34. Refaee, E.A.; Ahmed, B.; Saad, M. AraBEM at WANLP 2022 Shared Task: Propaganda Detection in Arabic Tweets. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 524–528. Available online: https://aclanthology.org/2022.wanlp-1.62 (accessed on 27 September 2023).
35. Attieh, J.; Hassan, F. Pythoneers at WANLP 2022 Shared Task: Monolingual AraBERT for Arabic Propaganda Detection and Span Extraction. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 534–540. Available online: https://aclanthology.org/2022.wanlp-1.64 (accessed on 27 September 2023).
36. Singh, G. AraProp at WANLP 2022 Shared Task: Leveraging Pre-Trained Language Models for Arabic Propaganda Detection. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 496–500. Available online: https://aclanthology.org/2022.wanlp-1.56 (accessed on 26 September 2023).

37. Taboubi, B.; Brahem, B.; Haddad, H. iCompass at WANLP 2022 Shared Task: ARBERT and MARBERT for Multilabel Propaganda Classification of Arabic Tweets. In Proceedings of the Seventh Arabic Natural Language Processing Workshop (Wanlp), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 511–514. Available online: https://aclanthology.org/2022.wanlp-1.59 (accessed on 26 September 2023).
38. van Wissen, L.; Boot, P. An Electronic Translation of the LIWC Dictionary into Dutch. In Proceedings of the eLex 2017: Lexicography from Scratch, Leiden, The Netherlands, 19–21 September 2017; pp. 703–715. Available online: https://pure.knaw.nl/portal/en/publications/an-electronic-translation-of-the-liwc-dictionary-into-dutch(de9c8272-0df1-4c92-bcb3-d789ad793603)/export.html (accessed on 22 June 2020).
39. Cruz, J.C.B.; Cheng, C. Establishing Baselines for Text Classification in Low-Resource Languages. 2020. Available online: http://arxiv.org/abs/2005.02068 (accessed on 22 June 2020).
40. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, USA, 2–4 May 2013; pp. 1–12.
41. Smetanin, S.; Komarov, M. Sentiment analysis of product reviews in Russian using convolutional neural networks. In Proceedings of the Proceedings—21st IEEE Conference on Business Informatics, CBI 2019, Moscow, Russia, 15–17 July 2019; Volume 1, pp. 482–486. [CrossRef]
42. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MI, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
43. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108. Available online: http://arxiv.org/abs/1910.01108 (accessed on 22 June 2020).
44. Manning, C.D. Electra: P Re—Training T Ext E Ncoders As D Iscriminators R Ather T Han G Enerators(ICLR2020). *arXiv* **2020**, arXiv:2003.10555. Available online: https://github.com/google-research/ (accessed on 22 June 2020).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.