Expert Systems With Applications 151 (2020) 113350

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model

Sawinder Kaur[a,*], Parteek Kumar[a], Ponnurangam Kumaraguru[b]

[a] Thapar Institute of Engineering and Technology, Patiala, India
[b] Indraprastha Institute of Information Technology, Delhi, India

ARTICLE INFO

Article history:
Received 27 August 2019
Revised 27 February 2020
Accepted 27 February 2020
Available online 28 February 2020

Keywords:
Clickbait
News
Classifier
Features
Social media

ABSTRACT

Clickbait indicates the type of content with an intending goal to attract the attention of readers. It has grown to become a nuisance to social media users. The purpose of clickbait is to bring an appealing link in front of users. Clickbaits seen in the form of headlines influence people to get attracted and curious to read the inside content. The content seen in the form of text on clickbait posts is very short to identify its features as clickbait. In this paper, a novel approach (two-phase hybrid CNN-LSTM Biterm model) has been proposed for modeling short topic content. The hybrid CNN-LSTM model when implemented with pre-trained *GloVe* embedding yields the best results based on accuracy, recall, precision, and F1-score performance metrics. The proposed model achieves 91.24%, 95.64%, 95.87% precision values for Dataset 1, Dataset 2 and Dataset 3, respectively. Eight types of clickbait such as *Reasoning, Number, Reaction, Revealing, Shocking/Unbelievable, Hypothesis/Guess, Questionable, Forward referencing* are classified in this work using the Biterm Topic Model (BTM). It has been shown that the clickbaits such as *Shocking/Unbelievable, Hypothesis/Guess* and *Reaction* are the highest in numbers among rest of the clickbait headlines published online. Also, a ground dataset of non-textual (image-based) data using multiple social media platforms has been created in this paper. The textual information has been retrieved from the images with the help of OCR tool. A comparative study is performed to show the effectiveness of our proposed model which helps to identify the various categories of clickbait headlines that are spread on social media platforms.

1. Introduction

Clickbait refers to the content with a purpose to encourage visitors and attract attention by creating curiosity among users to click on a link to a particular web page (Gardiner, 2015; Tan & Ang, 2017). The maximum of the revenue generation for publishers of the online content revenue model is through advertising. Online content (Chen, Conroy, & Rubin, 2015) attracts users through click links known as clickbaits by creating curiosity among visitors. The content shared on social media for advertising is in the form of short text, attachments including videos, audios, images, sharing links, *etc.* Clickbait is an umbrella term that encompasses all kinds of content capable of instigating an increased click-through (Biyani, Tsioutsiouliklis, & Blackmer, 2016). Most of the clickbait headlines use the first form of a person like 'I', 'We', 'You'. Some of the clickbait headlines are given below.

What we found was really shocking!
You can never guess what Happened!
What happens next will surprise you!
Just click to see this!
Did you know this!
OMG You won't believe your Eyes!

The primary danger posed by clickbaits is not only that the news topics such as economics, science, politics are replaced by business, sports, entertainment, but also the focus has shifted on attention-grabbing shareable contents. Often misleading and unverified headlines in the form of clickbaits are major contributors to the spread of fake news (Horne & Adali, 2017) on the internet. In spite of all these effects of clickbaits, no comprehensive solution has been devised to remove the clickbait stories from the news feeds generated on user's accounts (Elyashar, Bendahan, & Puzis, 2017). Currently, some solutions for automatic clickbait detection (Potthast, Köpsel, Stein, & Hagen, 2016) are based on machine learning models (Cao, Le et al., 2017), among which many of them yield low accuracy rates (Rony, Hassan, & Yousuf, 2017).

In this paper, a novel approach is proposed which is capable to identify the difference between legitimate and clickbait posts by directly analyzing the headings for both textual and non-textual (text embedded in images) posts. To evaluate our model, three datasets have been collected. The first one is provided by Chakraborty, Paranjape, Kakarla, and Ganguly (2016), second is provided by Khater, Al-sahlee, Daoud, and El-Seoud (2018) and the

* Corresponding author.
E-mail addresses: skaur_phd17@thapar.edu (S. Kaur), parteek.bhatia@thapar.edu (P. Kumar), pk@iiitd.ac.in (P. Kumaraguru).

https://doi.org/10.1016/j.eswa.2020.113350
0957-4174/

third has been collected manually with the help of human annotations. The main contributions of this paper are as follows.

- A ground dataset has been prepared from the Facebook page and Reddit website with the help of human annotations.
- Text extraction from the non-textual (image-based) data with the help of OCR tool using a pre-processing approach has been performed in this paper.
- Automatic identification of the eight types of clickbait headlines into *reasoning, number, reaction, revealing, shocking/unbelievable, hypothesis/guess, questionable, forward referencing* is done.
- A novel approach is proposed which works under a two-phase structure. In the first phase, the headlines of textual and non-textual data are fed to the CNN-LSTM model with the use of pre-trained vectors to identify whether the chosen post is legitimate or a clickbait. In the second phase, the classified clickbaits are fed to the Biterm Topic Model (BTM) which is a type of short text classifier. It uses a biterm co-occurrence of words to cluster similar clickbait headlines to identify the types of clickbait.
- A comparative study is performed with the existing systems to show the effectiveness of our proposed two-phase hybrid CNN-LSTM Biterm model.

The structure of the paper is organized as follows. Section 2 gives a brief overview of the related work done in the field of clickbait detection. The problem statement is discussed in Section 3. Section 4 discusses the corpus used for the experiment. Section 5 covers the architecture of our proposed system. The experimental results achieved after implementing our proposed model is presented in Section 6. A comparative analysis is performed with various existing systems in Section 7. The paper is concluded in Section 8 along with its future scope.

2. Related work

Clickbait has become an imperative subject for research purposes, as it is used by computer research and linguists team. An overview of the related approaches for clickbait detection has been discussed in this section.

Vijgen et al. (2014) studied listicles which are one of the vital components of clickbaits. The authors deliberated approximately 7000 clickbaits by Buzzfeed and the shred like “16 Cancer making food you eat every day” or “Do you know 39 celebrities who passed away this year” are some compiled examples from their collected dataset.

A clickbait type known as forward reference was studied by Blom and Hansen (2015) from a Danish news website in the form of headlines and is used for two purposes either to attract the user to click the title or to make the information gap by giving a headline. According to the authors, clickbaits mainly composed of adverbs, articles, demonstrative and personal pronouns.

Ferro et al. (2016) brought together approximately 3,000 tweets from the top twenty publishers on twitter among other computer scientists and created a model that used handcrafted features from three fields which are, the meta information, the linked web page, and the teaser message or title. Among these three, the linked web page comprised of readability and text features, and the teaser message had some basic dictionary and text features, while the meta information had tweets related features. These features were classified into a supervised mechanism and achieved 0.79 ROC-AUC for both precision and recall at 0.76. They also revealed that features retrieved from the first category outperformed rest of the categories taken into consideration, with n-gram and uni-gram features contributing the maximum as they are known for capturing the writing styles.

Eight types of clickbait were expounded by Biyani et al. (2016). The authors used the definitions of such types to collect non-clickbait (2,724) and clickbait (1,349) webpages, respectively from the Yahoo homepage. The handcrafted features like presences of quotes, questions, exclamations, *etc.*, are taken along with traditional features like uni-grams and bi-grams to develop the machine learning model for identifying the clickbait. Comparable features calculated the resemblance between the first five lines of the body and title of the article individually, and non-formality features were used to calculate the quality and formality of pages. Forward reference features were created after (Blom & Hansen, 2015) had given four types of forward references for clickbait class. They achieved 0.712 precision and 0.548 recall.

A Machine Learning classifier based browser plug-in called ‘Stop Clickbait’ has been proposed by Chakraborty et al. (2016). The classifier has been trained by extracting the clickbaits (8069 articles) which they crawled from web domains such as ViralStories, ScoopwHoop, ViralNova, Buzzfeed and 18,513 Wikinews articles as legitimate posts. They used 14 features spanning to train their SVM classifier and reported the accuracy of 89%, 93% in blocking and detecting the clickbait.

In contrary, the methods given by authors are both painstaking and time-consuming tasks as they used handcrafted features. To the best of our knowledge, no work has been done to automatically (Tacchini, Ballarin, Della Vedova, Moret, & de Alfaro, 2017) identify the type of clickbait on the basis of short text. So, in this paper, a novel two-phase approach is proposed. The model works in two phases. In the first phase, features are extracted from the clickbait headlines using the embedding *(GloVe)* model, which are fed to Convolutional Neural Network (CNN) - Long Short-Term Memory (LSTM) model to classify the headlines as legitimate or clickbait. In the second phase, the retrieved headlines labeled as clickbait from the first phase are fed to the BTM to analyze the type of cluster (topic) for clickbait headlines. The problem statement to identify the type of clickbait has been discussed in the next section.

3. Problem statement

The work proposed in this paper addresses two types of issues.

- To identify clickbait and non-clickbait headlines seen in the form of short text.
- To categorize the identified clickbait headlines into various clusters (topics).

Thus, the problem statement has been formulated as follows. To identify the headline as a binary classification problem, at the first phase the binary set of classes, $C = \{non\_clickbait, clickbait\}$ are taken into consideration. Consider H as a set of all headlines and a training set $T \subseteq H \times C$ of labeled sentences to train a model to learn a function F. The function F maps the headlines H to $\{non\_clickbait, clickbait\}$ such as F: $H \rightarrow C$. At the second phase, the clickbait headlines are classified into 8 categories, $C_t = \{Re, Nm, Rea, Rev, Sh, Hy, Ques, Fr\}$, where $Re \in$ *Reasoning*, $Nm \in$ *Number*, $Rea \in$ *Reaction*, $Rev \in$ *Revealing*, $Sh \in$ *Shocking/Unbelievable*, $Hy \in$ *Hypothesis/Guess*, $Ques \in$ *Questionable*, $Fr \in$ *Forward referencing* using unsupervised model.

4. Data collection

Many sources are utilized for clickbait generation such as Twitter (Potthast et al., 2018), LinkedIn (Chakraborty, Sarkar, Mrigen, & Ganguly, 2017), Facebook (Chakraborty et al., 2017), *etc.*, which are used by publishers as a trading platform. Such platforms are used to disseminate inappropriate information in the form of clickbait posts.

Table 1
Statistics of collected datasets

Datasets	Total headlines	Clickbait headlines	Non-clickbait headlines	Vocabulary length	Year of creation
Dataset 1	32,000	15,999	16,001	18,966	2015
Dataset 2	12,000	5,637	6,080	13,232	2017
Dataset 3	1,800	1,200	600	4,596	2019

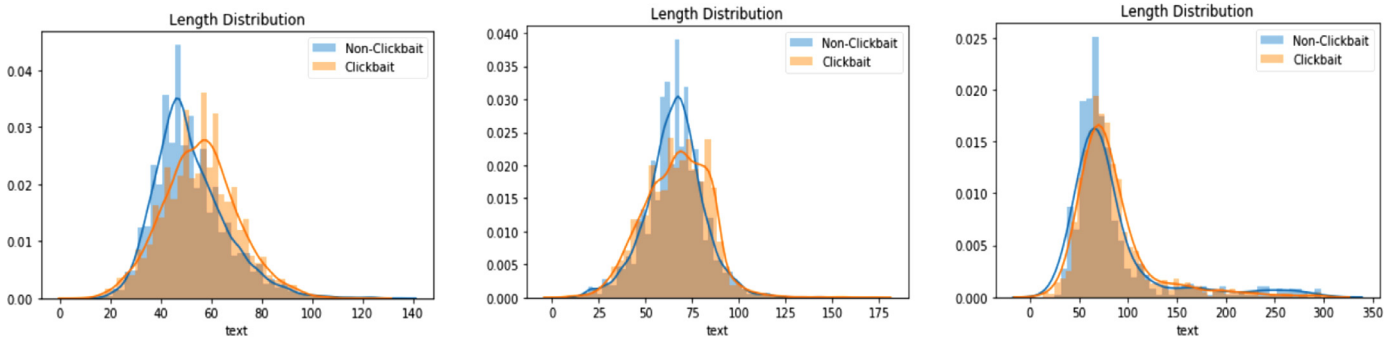

Fig. 1. Length distribution of clickbait and non-clickbait headlines on (a) Dataset 1, (b) Dataset 2 and (c) Dataset 3.

Various datasets are taken into consideration for evaluating our proposed model. The first dataset, taken from Chakraborty et al. (2016) is named as 'Dataset 1' in this paper. It contains 32,000 headlines of news articles from several web domains such as 'ViralStories', 'Scoopwhoop', 'Thatscoop', 'ViralNova', 'Upworthy', 'Buzzfeed', 'The Hindu', 'The Guardian', 'New York Times', 'Wikinews'. Among these headlines, a total of 15,999 were identified as clickbait headlines whereas 16,001 as non-clickbait headlines to develop a two-phase classification model.

The Second dataset, taken from Khater et al. (2018) is named as 'Dataset 2', which consists of 12,000 headlines. Among which 5,637 are the clickbaits and 6,080 are the non-clickbait headlines retrieved after pre-processing phase.'The Huffington Post', 'The Times Of India', 'NewsWeek' and 'BuzzFeed' web domains were chosen to fetch the data for clickbait headlines whereas 'The Indian Express', 'National Geographic', 'The wall street journal', 'The Economist', 'Thr Guardian'and 'The Hindu' web domains were chosen to fetch the non-clickbait headlines. The data was collected and got published online in Jan-2017.

A ground dataset has been created and is named as 'Dataset 3' in this paper. The dataset was collected from two sources, *i.e.*, Reddit[1] website (Agrawal, 2016) using *Octoparse*[2] (a web-scraping tool) and Facebook[3] page with the help of human annotations. Facebook and Reddit are the most popular social networking sites. The dataset was collected to analyze the distribution of both clickbait and non-clickbait headlines in terms of shares, likes, comments, domains, time (available from 1-DEC-2016 to 21-JUN-2019). The non-clickbait headlines were collected from subreddits (worldnews,[4] news[5]), which do not allow the clickbaits to creep in.

The overall statistics of our three collected datasets are discussed in Table 1, where the ratio of clickbait headlines for Dataset 1, Dataset 2, and Dataset 3 are 49.99%, 46.97% and 40%, respectively. The vocabulary length to classify the headlines for Dataset 1 are 18,966, for Dataset 2 are 13,232 and 4,596 for Dataset 3.

The length distribution of clickbait and non-clickbait headlines is represented by graphs shown in Fig. 1. The X-axis labeled as *'text'* represents the number of terms used in clickbait headlines whereas Y-axis represents the corresponding number of headlines having the same length distribution. It has been observed from Fig. 1(a) and (b) that the non-clickbait headlines have longer length distribution than clickbait headlines. Whereas from Fig. 1(c), no appropriate difference is noted between clickbait and non-clickbait headlines.

The mean distribution of the clickbait datasets as given in Table 2 for Dataset 1, Dataset 2 and Dataset 3 is 55.74, 66.72 and 84.27, respectively. The mean values of clickbait headlines are greater than the mean values of non-clickbait headlines for the same datasets. The difference between the mean values is not so significant when evaluated on the basis of the Z-test. Hence, it can be concluded that the length distribution of headlines for clickbait news is often longer than the non-clickbait headlines published on social networking sites (Song, Lee, & Kim, 2015).

5. Architecture of the proposed system

The overview of the proposed two-phase model, where the hybrid CNN-LSTM model (Wang, Jiang, & Luo, 2016) is embedded at the first phase and the BTM (Yan, Guo, Lan, & Cheng, 2013) at the second phase to identify the type of online clickbaits is shown in Fig. 2. To evaluate the proposed system, various datasets have been collected from three different sources in the first phase. During the pre-processing phase of textual data (Kaur, Kumar, & Kumaraguru, 2019), extra whitespaces are removed and the conversion of all characters from upper to lower case is processed (Collobert et al., 2011). Whereas the non-textual data is processed using the Optical character recognition (OCR) tool (Mulfari, Celesti, Fazio, Villari, & Puliafito, 2016) to retrieve the textual content and then the pre-processing step is applied to it. Once completed with the pre-processing pipeline, the next step is to extract features (Sboev, Litvinova, Gudovskikh, Rybka, & Moloshnikov, 2016) from the collected data. The extraction of features from the given data is called as feature engineering. It is done with the help of pre-trained embedding models. During the embedding process, various words are represented using dense vector representation. Word embedding is an improvement over the Bag of Words (BOW) approach (Kaur et al., 2019) where each word is represented using sparse vectors to represent the entire vocabulary. The representations are sparse because every single word is represented by a

[1] https://www.reddit.com/r/SavedYouAClick.
[2] https://www.octoparse.com/download.
[3] https://www.facebook.com/StopClickBaitOfficial.
[4] https://www.reddit.com/r/worldnews.
[5] https://www.reddit.com/r/news.

Table 2
Mean distribution of collected datasets.

Datasets	Mean distribution of labeled datasets	Standard deviation of labeled datasets	Mean distribution of headlines labeled as clickbait	Mean distribution of headlines labeled as non-clickbait
Dataset 1	0.50	0.50	55.74	51.85
Dataset 2	0.51	0.49	66.72	65.65
Dataset 3	0.32	0.46	84.27	83.86

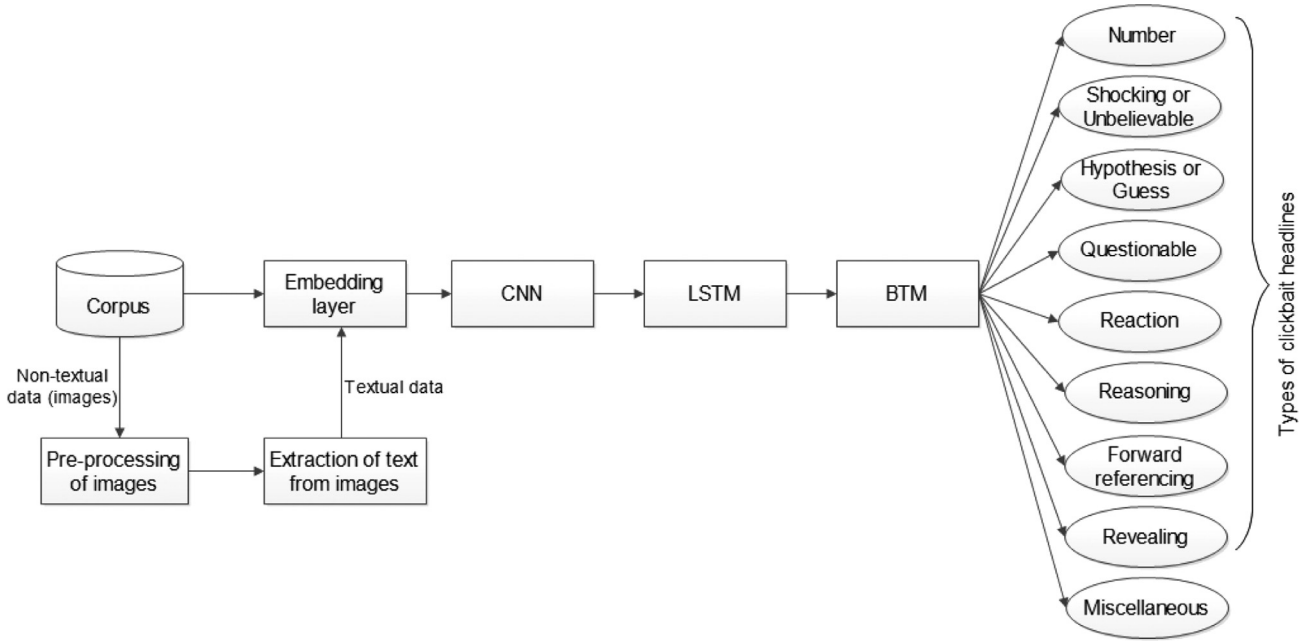

Fig. 2. Overview of our proposed two-phase hybrid CNN-LSTM Biterm model for automatic detection and classification of various types of clickbait headlines.

row comprised of zero values. Whereas in an embedding layer every single word is represented by dense vectors. The dense vectors represent the projection of a word in a continuous vector space. The position of a word is represented within these vector spaces learned from the short headline text and is called as its embedding. In the next step, the retrieved vectors are fed to the hybrid CNN-LSTM model to classify the features of textual content. The model classifies the content as clickbait or non-clickbait. Further, the identified clickbait headlines are fed to the BTM (a short text classifier) to identify the type of clickbait (Yan et al., 2013). The detailed description of each phase has been further discussed in this section.

5.1. Pre-processing

Our system can take two types of input, *i.e.*, in the form of textual and non-textual data as shown in Fig. 3.

To process textual data there is no requirement of OCR tool. Whereas non-textual data cannot be directly processed and fed to the proposed model. So, a pre-processing step is required to extract the textual information from non-textual (image-based) data. During the pre-processing of non-textual data, following operations are performed:

- Converting to grayscale: The quality of the image is improved by removing the color variation by converting the original image (as shown in Fig. 4(a)) into a grayscale image (as shown in Fig. 4(b)) for accurate text detection.
- Noise removal: Non-local Means Denoising algorithm (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) has been used for noise removal from an image. The denoised image retrieved can be seen in Fig. 4(c).

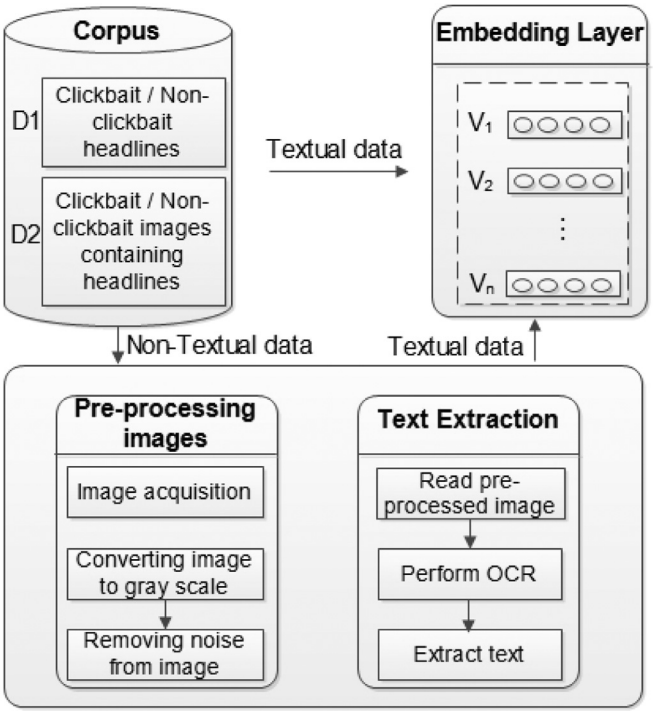

Fig. 3. Pre-processing of non-textual (image-based) data to extract textual content

- Extraction: The Google cloud vision (an OCR tool) is used to extract text from images. It requires Google cloud access and the creation of an API key. The *google-cloud-vision* and *google-cloud*

(a) (b) (c)

Fig. 4. Pre-processing on (a) Original image, (b) Grayscale image and (c) Denoised image.

Table 3
Embedding layer specifications of the proposed model.

Arguments	Description	Size
Input Dimension	It specifies the size of vocabulary in our collected corpus	20,000
Output Dimension	It specifies the size of vector space where words are embedded	128
Input length	It specifies the length of the input sequences, *i.e.*, the maximum limit of words for each headline.	100

libraries have been used for reading the images for further processing (Mulfari et al., 2016).

- Auto-correction: The python libraries like *autocorrect* and *language_check* are used in the experiment to correct the spelling errors (Kulkarni & Shivananda, 2019).

Whereas during the pre-processing phase of textual headlines, all the missing values are removed from the document using *dropna* method in python dataframe, *lower* and *sub* methods are used to convert the upper characters to lower cases and remove the whitespaces, respectively.

The formal representation of features chosen from textual and non-textual data has been discussed below.

Let K be the collection of posts published online over social media platforms and H is defined as a collection of headlines extracted from posts K. $H(i)$ is defined as the extracted headline from the ith post as represented by Eq. (1),

$$H(i) = \begin{cases} K(i), & \text{if } K(i) = Textual\ data \\ OCR(K(i)), & \text{if } K(i) = Non-Textual\ data \end{cases} \quad (1)$$

where i is the index which is repeated for the collection of posts with range $[0 : (|K| - 1)]$. $OCR(K(i))$ is considered as a function that extracts text from the image if the textual content is present else stores *NULL*.

The embedding layer used in the proposed model is discussed in the following section.

5.2. Embedding layer

In this paper, word embeddings are used to provide a better vector feature representation of words. *Gensim* (NLP library) provides an amazing wrapper to adopt different pre-trained word embedding models which include *GloVe* (by Stanford) (Sisodia, 2019) and *Word2vec* (by Google) (Mikolov, Chen, Corrado, & Dean, 2013a).

The embedding layer specifies three arguments that are defined for the first hidden layer of the hybrid CNN-LSTM model as shown in Table 3.

5.2.1. GloVe

Global Vectors for Word Representation *(GloVe)* is provided by the Stanford NLP team. The team also provides various versions of the *GloVe* pre-trained vectors. Among various available versions, *GloVe 1.2* is used to perform the experiment. It consists of four files with four different embedding representations such as *glove.6B.50d.txt* with *6* billion tokens and *50* features, glove.6B.100d.txt with 6 billion tokens and 100 features, *glove.6B.200d.txt* with *6* billion tokens and *200* features, and *glove.6B.300d.txt* with *6* billion tokens and *300* features (Sisodia, 2019). The script uses *glove.6B.100d.txt* (with 400,000 vocabulary size) file[6] to train the hybrid CNN-LSTM model. The *GloVe* is an unsupervised learning model that is used to generate word to word co-occurrence matrix from a corpus. Python library named as *glove_python* is used to implement *GloVe* embeddings.

5.2.2. Word2vec

It is a type of predictive model which is used to predict the destination word from the context of its neighboring words. The word vectors are trained on Google news data using skip-gram to build the model provided by Mikolov, Chen, Corrado, and Dean (2013b) in 2013. Each word is encoded using a one-hot encoding scheme and then is fed to the hidden layer of the hybrid CNN-LSTM model using a matrix of weights. Our experiments also used a pre-trained vector with a vocabulary size of *3* million words trained from *100* billion words of Google News data.The pre-trained vector file[7] for 300-dimensional word vectors was downloaded to perform the experiment.

5.3. Deep-learning based hybrid classifier

The architecture of the hybrid classifier mainly consists of two components, *i.e.*, Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). CNN (Kim, 2014) is applied to extract high-level sequences of word features. Whereas the LSTM (Eidnes, 2015) is embedded to capture long term sequences and also reduces the training time of the proposed model. The processed data (textual and non-textual) is fed to the hybrid CNN-LSTM model (Wang et al., 2016) where hyper-parameters are tuned in a standard split.

5.3.1. CNN component

The data extracted from the embedding layer is fed to the CNN module. Where the features are detected at different regions with the help of a sliding filter vector evolved in the convolution layer.

[6] https://nlp.stanford.edu/projects/glove/.
[7] https://docs.google.com/file/d/0B7XkCwpI5KDYeFdmcVltWkhtbmM/edit.

Consider $u_j \in \mathbb{R}^n$ as the n-dimensional word vectors of a particular word in a headline for j^{th} position. Let $u \in \mathbb{R}^{L\times n}$ be the input headline, where L denotes the length of the headline. Consider f as the length of the filter and vector $v \in \mathbb{R}^{f\times n}$ be the filter for performing convolution operation. A window vector w_i with f consecutive word vectors, where i is the position of a word in a headline is given by Eq. (2),

$$w_i = [u_i, u_{i+1}, ..., u_{i+f-1}] \tag{2}$$

where commas denote the row vector concatenation. The filter v revolves at each position with the window vectors to generate a feature map $m \in \mathbb{R}^{L-f+1}$, where each element m_i of feature map for window vector w_i is represented by Eq. (3),

$$m_i = t(w_i \odot v + b) \tag{3}$$

where b is the biased term that belongs to $\mathbb{R}$ and t is the non-linear transformation function and $\odot$ is the element-wise multiplication. In our experiment, *ReLu* (Nair & Hinton, 2010) is chosen as a non-linear function. To generate multiple feature maps, the hybrid CNN-LSTM model uses multiple filters. For each row W_i of $W \in \mathbb{R}^{(L-f+1)\times z}$ is a new feature representation generated from z filters for window vector at position i. Max-pooling is applied to feature maps after the convolution layer which helps to select the top most important features. Further, the successive window representation is fed into the LSTM model.

5.3.2. LSTM component

In LSTM, the output of the module is controlled by a set of gates (forget gate (f_s), input gate (i_s), output gate (o_s)). In our experiment, all vectors used in LSTM architecture share the same memory dimensions. The working of i_s, f_s and o_s is denoted by Eqs. (4)–(6),

$$i_s = \sigma(W_i \cdot [h_{s-1}, x_s] + b_i) \tag{4}$$

where σ is the logistic sigmoid function that gives output between [0,1], x_s is the input at current time step s and h_{s-1} is the previous output of the hidden state.

$$f_s = \sigma(W_f \cdot [h_{s-1}, x_s] + b_f) \tag{5}$$

Here, f_s is the function to control the extent of information required from the old memory cell to be thrown away.

$$o_s = \sigma(W_o \cdot [h_{s-1}, x_s] + b_o) \tag{6}$$

Here, o_s is the control of output on the memory cell. Current input and previous output of the hidden state are taken into consideration to update the current hidden state h_s and current memory cell c_s through various transition functions as shown by Eqs. (7)–(9),

$$q_s = tanh(W_q \cdot [h_{s-1}, x_s] + b_q) \tag{7}$$

where *tanh* is the hyperbolic function that gives output between [-1,1].

$$c_s = f_s \odot c_{s-1} + i_s \odot q_s \tag{8}$$

Here, c_s represents the current memory cell and $\odot$ denotes the element-wise multiplication,

$$h_s = o_s \odot tanh(c_s) \tag{9}$$

where h_s represents the current hidden state.

LSTM model is chosen and embedded on the CNN model to learn the high-level sequence features. To get the final output from the hidden layer, sigmoid layer is used.

5.3.3. Implementation of hybrid classifier

The number of filters used in the hybrid CNN-LSTM model for our experiment is 128. The size of 1-dimensional convolution window is considered as 3 and the 'ReLu' activation function is used for the hidden layers. The value of the pool size is selected as 3 for max-pooling layers used in the hybrid CNN-LSTM model (Wang et al., 2016). The early stopping method (Zhang, Bengio, Hardt, Recht, & Vinyals, 2016) is used to decide the number of training epochs. Initially, 50 epochs were chosen to analyze the model performance. It was observed that the model performance stops showing any significant improvement after 15 epochs as the accuracy rate remains nearly the same till 50 epochs. So, during the learning process, our model is trained for 15 epochs. The architecture of the hybrid CNN-LSTM model is shown in Fig. 5.

The input size in this architecture is the maximum sequence length chosen for clickbait headlines,*i.e.*, 100. The input is fed to the embedding layer. The output dimensions (100 × 100) are retrieved from the embedding layer which is further fed to the convolution layer, where the number of parameters used is 38,528. The output dimensions of the first convolution layer are 98 × 128, which is fed as an input to the first max-pooling layer giving an output of 32 × 128. The second convolution layer takes 32 x 128 dimensions as input and trains the vector with 49,280 parameters. In the next step, the second max-pooling layer takes 30 x 128 input dimensions from the second convolution layer and gives 10 x 128 as an output shape. The output from max-pooling layer is fed to LSTM (flattening) layer which gives 100 values with 91,600 parameters. The output is then fed to the dense layer with 256 (values) x 128 (number of neurons) + 128 (bias values for neurons) along with 25,856 number of parameters, which is fed to the sigmoid layer to identify the clickbait headlines.

The total parameters, trainable and non-trainable parameters taken by the hybrid CNN-LSTM model for Dataset 1, Dataset 2 and Dataset 3 are shown in Table 4.

It has been observed that the total parameters of our collected dataset are 1,268,501, whereas only 268,501 parameters are used by the model to find the optimal weights to reduce the cost function of the model.

The classified clickbait headlines retrieved from the hybrid CNN-LSTM model are fed to BTM to identify the types of clickbait.

5.4. Biterm topic model (BTM)

Discovering topics from short texts has genuinely become an intractable problem. It was hard for traditional topic models such as *k-Means, Probabilistic latent semantic analysis (PLSA), Latent discriminant analysis (LDA) (Hoffman, Bach, & Blei, 2010)* to identify short texts as they suffered from severe data sparsity problem when implemented on short texts. So, the BTM came into existence to overcome this limitation.

BTM is a word co-occurrence biterm based model as shown in Fig. 6. Where a biterm consists of two words co-occurrence in the same headline. It is a type of generative model which generates a biterm by making a two word pattern from the same topic t. The number of topics (t) chosen to apply BTM was decided after analyzing the elbow curve (Kodinariya & Makwana, 2013). A plot was drawn by taking into consideration the appropriate number of topics. The point of a bend (known as knee) was tested by varying the value of t from 4 to 12. The best compactness of clustering was considered when the value of t was set to 9, where an appropriate bend of the curve was seen. The first eight clusters (topics) define the types of clickbait and the last cluster (9^{th}) defines the miscellaneous clickbait headlines as it was a loosely packed cluster covering mixed topics. Some of the generative processes for the collected corpus in BTM where β and γ are considered as the Dirichlet priors (Dp) (Yan et al., 2013) is discussed as follows.

Table 4
Number of parameters in the hybrid CNN-LSTM model using *GloVe* embeddings.

Corpus	Total Parameters	Trainable parameters	Non-trainable parameters
Dataset 1	2,287,701	287,701	2,000,000
Dataset 2	219,422	217,722	17,000
Dataset 3	1,268,501	268,501	1,000,000

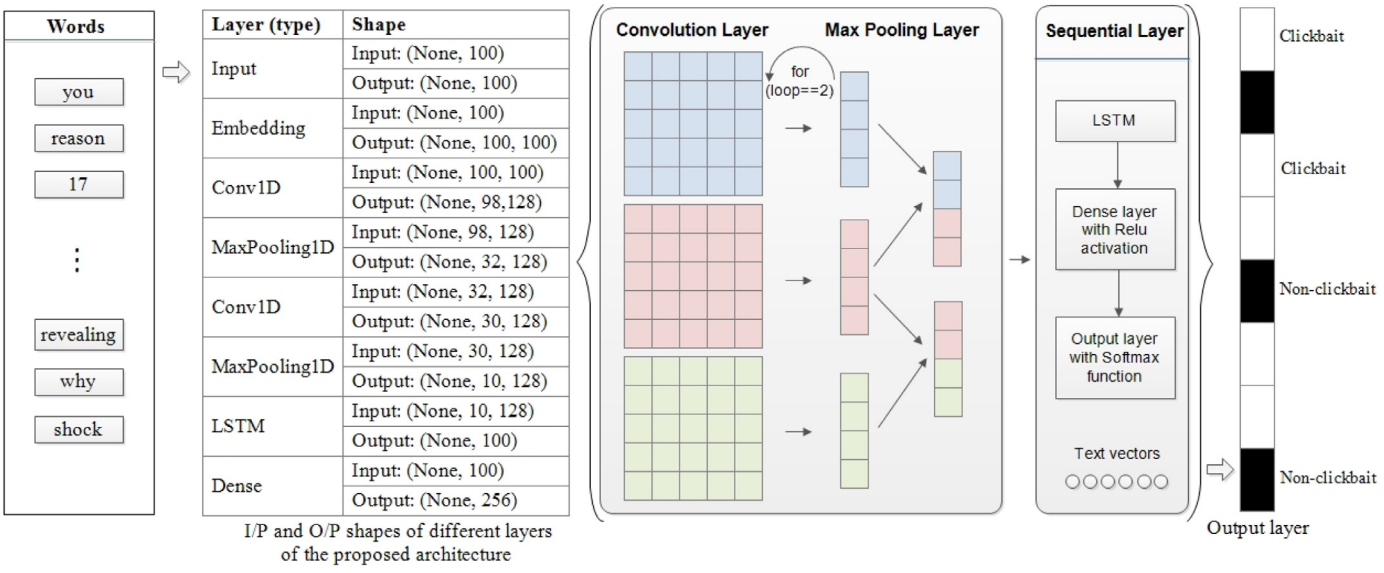

Fig. 5. Architecture of the hybrid CNN-LSTM model.

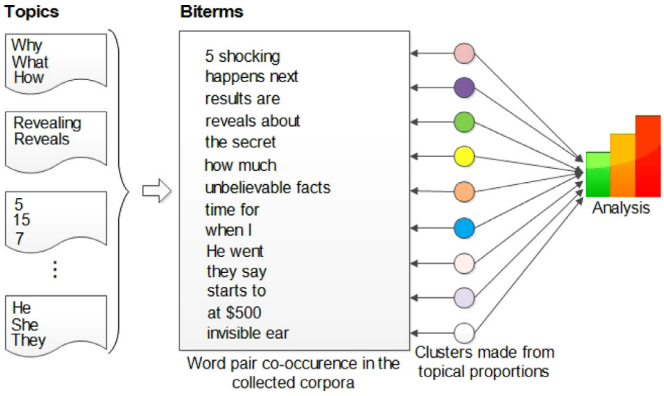

Fig. 6. Architecture of the Biterm Topic Model (BTM).

- For each topic t,
  - a topic-specific word distribution is drawn $\delta_t \sim Dp(\gamma)$
- Draw a topic distribution $\vartheta \sim Dp(\beta)$ for the chosen dataset
- For every biterm (b) in biterm set (B)
  - draw a topic assignment $t \sim Multinomial(\vartheta)$
  - draw b (*i.e., two word occurrence from the same headline*): $(w_x, w_y) \sim Multinomial(\delta_t)$

The joint probability of b can be represented by Eq. (10).

$$P(b) = \sum_t P(t)P(w_x|t)P(w_y|t) = \sum_t \vartheta_t \delta_{x|t} \delta_{y|t} \tag{10}$$

Hence, the likelihood of the whole chosen dataset is represented by Eq. (11).

$$P(B) = \prod_{(x,y)} \sum_t \vartheta_t \delta_{x|t} \delta_{y|t} \tag{11}$$

The argument t represents the number of topics chosen to classify the types of clickbait. The symmetric Dirichlet prior probability of a topic is $P(t)$ and is kept 1 whereas the symmetric Dirichlet prior probability of a word with given topic $P(w|t)$ is kept to 0.01. The Gibbs algorithm (Finkel, Grenager, & Manning, 2005) is used for categorizing the type of clickbait, the number of iterations for Gibbs sampling is kept 10. The default window size (15) is considered for performing the second phase experiment.

Thus, the BTM helps to convey the semantic features of related topics by making use of co-occurrence patterns based on biterms.

6. Experimental results

The configuration of the computing environment includes: Processor is Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz 2.60GHz, DDR3 16GB RAM and Nvidia Titan Xp GPU with 3840 Cores running at 1404MHz. Theano 1.0.3 (Bergstra et al., 2010) is used with Python version 3.6.3 on Windows 10 Pro Operating System. To evaluate the proposed model, the performance analysis of each phase has been discussed in this section.

6.1. OCR performance

It has been observed from the results obtained in Table 5 that the Google cloud vision tool (Mulfari et al., 2016) gives an accuracy of 87.60% while extracting headlines from the non-textual (image-based) posts for our Dataset 3. After pre-processing the images, it was analyzed that the accuracy to recognize correct text from images increased by 1.34%.

Also, it has been observed that the Google cloud vision tool performs best on *.bmp,.tff,.png,.jpg and.jpeg* to extract text from images (Vithlani & Kumbharana, 2015). The collected Dataset 3 used for the experiment has only *.jpg* and *.png* formats.

6.1.1. Tag clouds

Tag clouds of top 12 words are retrieved after pre-processing the headlines as shown in Figs. 7–9.

Table 5
Results obtained after performing OCR and pre-processing on Dataset 3.

Tool	Total images	Correctly identified	Incorrectly identified	Accuracy	Precision	Recall	F1-score
Google cloud vision	1,800	1,578	222	87.60	91.34	86.56	88.88
Google cloud vision after pre-processing	1,800	1,601	199	88.94	92.7	89.6	91.12

Fig. 7. Top 12 (a) Clickbait and (b) Non-clickbait word clouds generated from Dataset 1.

Fig. 8. Top 12 (a) Clickbait and (b) Non-clickbait word clouds generated from Dataset 2.

Fig. 9. Top 12 (a) Clickbait and (b) Non-clickbait word clouds generated from Dataset 3.

Fig. 7(a) shows the top words like *guess, questions, reasons, you, identify* retrieved from the clickbait headlines, whereas top words retrieved from non-clickbait headlines like *inauguration, iraq, advantage, government, supports* are shown in Fig. 7(b) for Dataset 1. Top words like *Twitter, celebs, actually, quote, most*, retrieved from the clickbait headlines are depicted in Fig. 8(a) and words like *delhi, family, mumbai, england, uk* are seen among top words, as shown in Fig. 8(b) for Dataset 2. For our collected Dataset 3, the tag clouds includes *why, don't, guess, what, reason, secret, worse* top words for clickbait headlines, as depicted in Fig. 9(a) and words like *saudi, agreement, sumbit, bomb, israel, india, pakistan, russia* are retrieved from non-clickbait headlines, as demonstrated in Fig. 9(b).

It can be concluded that the top words extracted from the clickbait headlines create the curiosity among users and urge them to click to view such type of headlines seen on various social media platforms in form of posts. The words retrieved from the non-clickbait headlines are genuine and do not create any type of confusing statements among users.

6.2. Classification performance

In this experiment, classification has been performed on Dataset 1, Dataset 2 and Dataset 3. To validate our proposed model various results are discussed in this section. It has been observed from Fig. 10 that the validation loss value for Dataset 1, Dataset 2 and Dataset 3 is 0.18, 0.43 and 0.27, respectively.

The performance analysis of three models (hybrid CNN-LSTM without pre-trained vectors, hybrid CNN-LSTM with *Word2vec* and *GloVe* pre-trained vectors) is evaluated in terms of accuracy, precision, recall, F1-score and ROC-AUC as depicted in Table 6. It has been observed that the hybrid CNN-LSTM model with pre-trained embeddings (*Word2vec and GloVe*) performs better than the hybrid CNN-LSTM model without pre-trained embeddings for all three datasets. The accuracy metric of the hybrid CNN-LSTM model using *GloVe* embeddings achieves an increase of 2.38%, 4.57% and 4.25% for Dataset 1, Dataset 2 and Dataset 3, respectively, when compared with the hybrid CNN-LSTM model using *Word2vec* embeddings. Also, an increase of 3.9% in precision and 5.74% in recall values is seen for our collected Dataset 3. It can be observed from Table 6 that the hybrid CNN-LSTM model when pre-trained with the *GloVe* embeddings outperforms the hybrid CNN-LSTM model when pre-trained with *Word2vec* embeddings in terms of accuracy, recall, F1-score, ROC-AUC performance metrics. Hence, the hybrid CNN-LSTM model with pre-trained *GloVe* embeddings is used to further classify clickbait and non-clickbait headlines.

The classified clickbait headlines are fed to BTM with specified arguments considered in our experiment.

6.3. Cluster analysis to identify the types of clickbait headlines

The approach followed in this work offers a preliminary discussion on misleading clickbait headlines. Various characteristics of misleading clickbait headlines have been analyzed in this section. To decide the appropriate number of topics (t) to apply the BTM, an elbow method (Kodinariya & Makwana, 2013) was applied. After running the elbow algorithm by varying the size of t from the range 4–12, it was analyzed that the best compactness of clustering was seen at 9. Thus the number of topics selected to apply BTM was 9.

To evaluate the topic quality of clusters, an automated metric, *i.e.,* coherence score has been used (Mimno, Wallach, Talley, Leenders, & McCallum, 2011). The coherence score of 9 clusters for Dataset 1, Dataset 2 and Dataset 3 is shown in Table 7. Three samples of top words (5, 10 and 20) are used to analyze the coherence score of each topic for Dataset 1, Dataset 2 and Dataset 3. It was observed that for each topic, some specific common words for all three collected samples of top words (5, 10 and 20) were extracted. Some frequent occurrence of similar words like *anger, guilt, react, reply* were seen in $Cluster_3$ and words like *unbelievable, impossible, shock, wow* were seen in $Cluster_5$, where the coherence score signifies the measure of the occurrence of frequent words in a topic. It was observed from Table 7 that the quality of $Cluster_9$ was very poor for our collected Dataset 1, Dataset 2 and Dataset

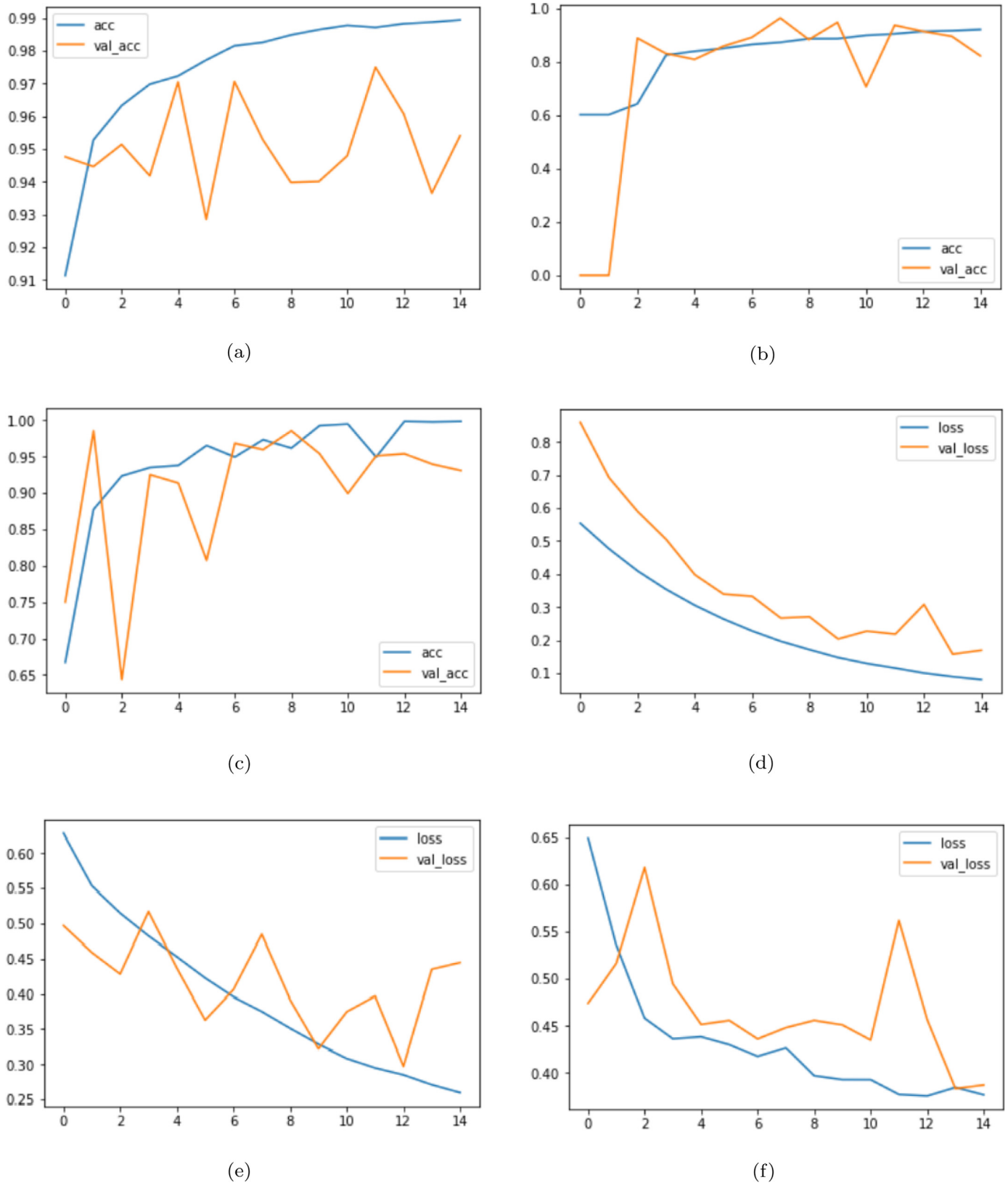

Fig. 10. Training accuracy vs validation accuracy for (a) Dataset 1, (b) Dataset 2, (c) Dataset 3, and Training loss vs validation loss for (d) Dataset 1, (e) Dataset 2, (f) Dataset 3 at different epochs.

Table 6
Comparative analysis of the hybrid CNN-LSTM architecture with and without pre-trained vectors using performance metrics.

Methods	Dataset	Accuracy	Precision	Recall	F1-score	ROC-AUC
Without Pre-trained vectors	Dataset 1	81.85	82.89	79.76	81.29	0.87
	Dataset 2	78.48	80.25	78.12	79.17	0.81
	Dataset 3	84.61	84.7	80.53	82.56	0.87
With *Word2vec* pre-trained vectors	Dataset 1	93.42	92.34	80.01	85.73	0.97
	Dataset 2	84.87	90.00	81.33	85.44	0.89
	Dataset 3	89.96	91.97	85.66	88.70	0.94
With *GloVe* pre-trained vectors	Dataset 1	95.8	91.24	85.43	88.23	0.99
	Dataset 2	89.44	95.64	89.69	92.56	0.94
	Dataset 3	94.21	95.87	91.4	93.58	0.98

Table 7
Coherence score of top 5, 10 and 20 words for Dataset 1, Dataset 2 and Dataset 3 among 9 clusters retrieved using BTM to analyze the quality of clusters.

Clusters	Top words								
	Dataset 1			Dataset 2			Dataset 3		
	5	10	20	5	10	20	5	10	20
$Cluster_1$	−16.31	−93.89	−302.17	−15.71	−109.21	−428.83	−5.67	−60.67	−371.41
$Cluster_2$	−19.01	−97.28	−425.67	−20.78	−102.9	−432.01	−14.55	−89.07	−330.38
$Cluster_3$	−20.28	−76.09	−287.97	−20.05	−94.11	−445.36	−16.48	−81.62	−336.96
$Cluster_4$	−20.03	−98.64	−353.91	−22.7	−93.19	−444.3	−17.34	−63.33	−376.81
$Cluster_5$	−20.63	−105.92	−362.89	−24.67	−92.23	−455.63	−19.69	−85.37	−348.38
$Cluster_6$	−21.55	−103.23	−379.1	−21.91	−118.02	−459.39	−17.54	−63.38	−360.41
$Cluster_7$	−22.13	−107.39	−389.15	−21.8	−116.13	−469.85	−18.55	−69.06	−380.7
$Cluster_8$	−22.46	−85.1	−430.3	−27.11	−109.21	−428.83	−22.44	−83.63	−349.88
$Cluster_9$	−25.53	−108.56	−446.45	−29.81	−121.45	−474.4	−28.18	−91.09	−376.84

Table 8
Description of identified topics.

Topics	Priority	Definition	Example
Reasoning	5	Deducing the fact out of the statement	Here's why chelseamperetti left "brooklyn nine-nine".
Number	8	Creating curiosity by a particular quantum/figure	29 pictures that will ruin your whole dang day.
Reaction	3	A process to act in response to a situation	Rumour: next DLC fighter for smash bros. ultimate seemingly teased by game supervisor.
Revealing	7	Making interesting by allowing a look at hidden content	Star trek: william shatner reveals if he'd return as kirk for a new series.
Shocking/Unbelievable	2	Causing a sudden violent impact or blow in mind	Can you shoot a feature film on an iPhone?
Hypothesis/Guess	1	Committing oneself to an opinion with little knowledge	You'll never guess what language kendall jenner can speak.
Questionable	4	Trying to invite inquiry	cCan you shoot a feature film on an iPhone?
Forward Referencing	6	Use of demonstratives, pronouns to create a reference to entities	Firefighters rescued litter of puppies. then they realized they weren't actually dogs.

Table 9
Distribution of headlines.

Types of clickbait	Dataset 1			Dataset 2			Dataset 3		
	Non-clickbait	Clickbait	Clickbait %	Non-clickbait	Clickbait	Clickbait %	Non-clickbait	Clickbait	Clickbait %
$Cluster_1$ (Reasoning)	18	257	93.45	12	66	84.61	4	50	92.5
$Cluster_2$ (Number)	2,981	7,057	70.3	3,875	3,248	45.59	28	482	94.50
$Cluster_3$ (Reaction)	8	43	84.31	1	24	96	0	5	100
$Cluster_4$ (Revealing)	45	94	67.6	20	27	57.44	7	63	90
$Cluster_5$ (Shocking or Unbelievable)	1	130	99.23	9	39	81.25	0	16	100
$Cluster_6$ (Hypothesis or Guess)	10	226	95.76	2	32	94.11	0	29	100
$Cluster_7$ (Questionable)	290	3,544	92.43	190	1,238	86.69	18	196	91.58
$Cluster_8$ (Forward referencing)	200	3,191	91.96	250	530	67.94	102	330	76.38
Sub-total	3,553	14,542	–	4,359	5,204	–	159	1,171	-
$Cluster_9$ (Miscellaneous)	12,448	1,457	10.47	1,721	433	20.1	441	29	6.17
Total	16,001	15,999	–	6,080	5,637	–	600	1,200	-

3. The coherence score for top words (5, 10 and 20) for Dataset 3 is −28.18, −91.09 and −376.84, respectively. The more positive coherence score depicts that the topics are more coherent and have better quality.

Various types of clickbait headlines are defined with an appropriate example in Table 8.

The distribution of clickbait headlines in each categorized cluster is shown in Table 9. It can be observed that 14,542 out of 15,999 clickbait headlines from Dataset 1 have been categorized among 8 types of clusters while the remaining headlines are clustered as a 'Miscellaneous' set (9^{th} cluster) of clickbait headlines.

Table 10
Comparison of the experimental results of our proposed model with the existing systems.

Author	Proposed approach	Feature selection	Models	Dataset	Results
López-Sánchez et al. (2017)	The proposed approach uses metric learning and deep learning algorithms by integrating them with Case-Based Reasoning methodology.	TF-IDF, n-gram, 300 dimensional *Word2vec*	CBR + CNN	Largest collected publicly available dataset (Chakraborty et al., 2016) having 32,000 (clickbait and non-clickbait) headlines.	The proposed approach achieved 0.994, 0.95, 0.90 average area under the ROC curve using *Word2vec, TF-IDF, n-gram count*.
Agrawal (2016)	A convolution neural network based model is proposed to detect clickbaits.	*Click-Word2vec, Click-scratch*	CNN	Created their own corpus from Reddit, Facebook, and Twitter social media platforms	*Click-scratch* - 89% accuracy with 0.87 ROC-AUC score; *Click-Word2vec* -90% accuracy with 0.90 ROC-AUC score
Chakraborty et al. (2016)	Build a browser extension to automatically detect the clickbait headlines.	Sentence Structure, Clickbait Language, Word patterns and n-gram features	SVM, Decision Tree, Random Forest	Collected 30,000 headlines (clickbait and non-clickbait) from ViralStories, Upworthy, BuzzFeed, Wikinews, Scoopwhoop, and ViralNova.15,000 headlines both in the clickbait and non-clickbait categories	SVM-93% accuracy rate with 0.95 precision, 0.90 recall, 0.93 F1-score and 0.97 ROC-AUC values; Decision Tree-90% accuracy rate with 0.91 precision, 0.89 recall, 0.90 F1-score and 0.90 ROC-AUC values; Random Forest-92% accuracy rate, 0.94 precision, 0.91 recall, 0.92 F1-score and 0.97 ROC-AUC values using a combination of all extracted features.
Khater et al. (2018)	A supervised machine learning algorithm was applied to classify the posts into clickbait and non-clickbait headlines.	Extracted 28 features. The most common are Bag of Words (BOW), noun extraction, similarity, readability, and formality.	Logistic regression, Linear SVM	The dataset used was provided by Bauhaus-Universitat Weimar at the time of clickbait detection challenge.	Logistic regression and Linear SVM gave 0.79, 0.78 precision and 0.79, 0.79 recall values, respectively.
Our proposed approach	A two-phase hybrid CNN-LSTM Biterm model is proposed to detect short topics (clickbait headlines) and its types.	Without pre-trained vectors, with *Word2vec* and *GloVe* (100 dimensional) pre-trained vectors	CNN + LSTM	First dataset (Chakraborty et al., 2016) consists of 32,000 (clickbait and non-clickbait) headlines. Second dataset (Khater et al., 2018) consists of 12,000 (clickbait and non-clickbait) headlines. Our created corpus (both textual and non-textual) from Reddit and Facebook social media platforms consists of 1,800 (clickbait and non-clickbait) headlines.	The accuracy achieved by our proposed hybrid CNN-LSTM model for Dataset 1, Dataset 2 and Dataset 3 using *GloVe* embeddings is 95.8%, 89.44% and 94.21%, respectively. Whereas the accuracy achieved by the proposed hybrid CNN-LSTM model using *Word2vec* embeddings for Dataset1, Dataset 2 and Dataset 3 is 93.42, 84.87 and 89.96, respectively.

The count of clickbait headlines from Dataset 2 is 5637, from which 5204 have been categorized into 9 clusters using BTM.

Dataset 3 has 1,200 clickbait headlines, among which 1171 have been categorized into 8 clusters. It has been observed from Table 9 that the clickbaits such as *Shocking/Unbelievable, Hypothesis/Guess* and *Reaction* are the highest in numbers among rest of the clickbait headlines published on social media platforms. Based on these findings, a priority number has been assigned to the identified clusters in Table 8. For the priority attribute, value 1 signifies the highest priority whereas 8 signifies the lowest priority.

The repeatable code for our proposed model has been uploaded to Github.[8] It will help the research community to regenerate the results.

7. Comparison with existing systems

Our proposed model outperforms other existing clickbait detection systems as shown in Table 10.

The dataset created for clickbait and non-clickbait headlines in this work has been collected from various social media platforms (Reddit, Facebook). The collected dataset consists of textual and non-textual (images with clickbait headlines) data. Agrawal (2016) also created the ground corpus (only textual data) for detecting clickbaits online using Reddit, Facebook, and Twitter platforms. The author used CNN with one layer convolution for detecting clickbaits using two variants of embeddings. The first variant includes *click-scratch* features which give 89% accuracy with 0.88 precision, 0.80 recall and 0.84 F1-score values. The second variant includes *click-Word2vec* features which give 90% accuracy with 0.85 precision, 0.88 recall and 0.86 F1-score values. It was observed that our proposed hybrid CNN-LSTM model with *Word2vec* embeddings outperforms (Agrawal, 2016) using *click-Word2vec* variant in terms of accuracy, precision and F1-score rates by 0.96%, 3.97% and 2.70%, respectively.

López-Sánchez, Herrero, Arrieta, and Corchado (2017) used metric and deep learning algorithms by integrating them with CBR (Case-Based Reasoning) methodology using *n-gram, Word2vec, TF-IDF* feature extraction techniques. The proposed model achieved 0.99, 0.95, 0.90 average area under ROC curves using *Word2vec, TF-IDF, n-gram count* on Chakraborty et al. (2016) dataset. Whereas our proposed approach achieves 0.99 ROC-AUC score using *GloVe* embeddings for the same dataset by precisely identifying the type of clickbait headlines that are spread on social media. Automatic identification of classified clickbait headlines is not seen in the case of López-Sánchez et al. (2017). Khater et al. (2018) used the dataset created by Bauhaus-Universitat Weimar giving 0.79, 0.78 precision, 0.79, 0.79 recall and 0.79, 0.79 F1-score values using Logistic regression, and Linear SVM models. Whereas, our proposed model gives 0.95 precision, 0.89 recall, 0.92 F1-score values using *GloVe* pre-trained vectors on the same dataset. Chakraborty et al. (2016) build a browser extension to automatically detect the clickbait headlines using *sentence structure, clickbait language, word patterns*, and *n-gram* features. Our proposed hybrid CNN-LSTM model using *GloVe* embeddings outperforms SVM, Decision Tree, Random Forest in Chakraborty et al. (2016) by 2.8%, 5.7%, 3.8% accuracy rates and 0.02, 0.09 and 0.02 ROC-AUC scores, respectively.

8. Conclusion and future scope

In this paper, interesting characteristic differences between clickbait and non-clickbait headlines have been highlighted. Such characteristics are used as features to classify clickbait headlines. A two-phase model has been proposed in this paper. In the proposed approach, the hybrid CNN-LSTM model is implemented in the first phase to identify clickbaits, which are further fed to the second phase where the BTM is implemented to analyze the category of clickbait headlines. The proposed approach outperforms when compared to the existing clickbait systems. It was surprising to note that the *Shocking/Unbelievable, Hypothesis/Guess* and *Reaction* types of clickbait are published maximum on social media platforms to attract the readers. To the best of our knowledge, no work has been done to detect the type of clickbait on subsequent visits by the users of social media.

In future, the end user will be able to identify the type of clickbait headline through a browser extension. The proposed system has the potential to provide an impulse to analyze the type of clickbait spreading at the time of political elections, natural disasters, terrorist attacks, *etc*. As it will become easy for the researchers to classify the category of clickbait spread at maximum during natural events. Such type of application can help in controlling the various types of clickbaits from growing over social media by activating a pop-up message indicating the warning in form of clickbait categories (Number, Questionable, Reasoning, Reaction, Revealing, Shocking/Unbelievable, Hypothesis/Guess, Forward referencing).

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.

Credit authorship contribution statement

Sawinder Kaur: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Validation, Writing - review & editing. **Parteek Kumar:** Validation, Supervision, Writing - review & editing. **Ponnurangam Kumaraguru:** Writing - review & editing.

Acknowledgement

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Agrawal, A. (2016). Clickbait detection using deep learning. In *2016 2nd international conference on next generation computing technologies (NGCT)* (pp. 268–272). IEEE.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., … Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th python in science conf: 1* (pp. 3–10).

Biyani, P., Tsioutsiouliklis, K., & Blackmer, J. (2016). "8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. *Thirtieth AAAI conference on artificial intelligence*.

Blom, J. N., & Hansen, K. R. (2015). Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics, 76*, 87–100.

Cao, X., Le, T. et al. (2017). Machine learning based detection of clickbait posts in social media. arXiv:1710.01977.

Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 9–16). IEEE.

Chakraborty, A., Sarkar, R., Mrigen, A., & Ganguly, N. (2017). Tabloids in the era of social media?: Understanding the production and consumption of clickbaits in twitter. *Proceedings of the ACM on Human-Computer Interaction, 1*(CSCW), 30.

Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). Misleading online content: Recognizing clickbait as false news. In *Proceedings of the 2015 ACM on workshop on multimodal deception detection* (pp. 15–19). ACM.

[8] https://github.com/sawinderkaurvohra/Clickbait-Detection.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research, 12*(Aug), 2493–2537.

Eidnes, L. (2015). Auto-generating clickbait with recurrent neural networks.

Elyashar, A., Bendahan, J., & Puzis, R. (2017). Detecting clickbait in online social media: You won't believe how we did it. arXiv:1710.06699.

Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., et al. (2016). *Advances in Information Retrieval: 38th European Conference on IR research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings: 9626*. Springer.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 363–370). Association for Computational Linguistics.

Gardiner, B. (2015). You'll be outraged at how easy it was to get you to click on this headline. *Wired Magazine*.

Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems* (pp. 856–864).

Horne, B. D., & Adali, S. (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *Eleventh international aaai conference on web and social media*.

Kaur, S., Kumar, P., & Kumaraguru, P. (2019). Automating fake news detection system using multi-level voting model. *Soft Computing*, 1–21.

Khater, S. R., Al-sahlee, O. H., Daoud, D. M., & El-Seoud, M. (2018). Clickbait detection. In *Proceedings of the 7th international conference on software and information engineering* (pp. 111–115). ACM.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a special interest group of the ACL* (pp. 1746–1751).

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal, 1*(6), 90–95.

Kulkarni, A., & Shivananda, A. (2019). Exploring and processing text data. In *Natural language processing recipes* (pp. 37–65). Springer.

López-Sánchez, D., Herrero, J. R., Arrieta, A. G., & Corchado, J. M. (2017). Hybridizing metric learning and case-based reasoning for adaptable clickbait detection. *Applied Intelligence*, 1–16.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 262–272). Association for Computational Linguistics.

Mulfari, D., Celesti, A., Fazio, M., Villari, M., & Puliafito, A. (2016). Using google cloud vision in assistive technology scenarios. In *2016 IEEE symposium on computers and communication (ISCC)* (pp. 214–219). IEEE.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814).

Potthast, M., Gollub, T., Komlossy, K., Schuster, S., Wiegmann, M., Fernandez, E. P. G., et al. (2018). Crowdsourcing a large corpus of clickbait on twitter. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1498–1507).

Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016). Clickbait detection. In *European conference on information retrieval* (pp. 810–817). Springer.

Rony, M. M. U., Hassan, N., & Yousuf, M. (2017). Diving deep into clickbaits: Who use them to what extents in which topics with what effects? In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017* In *ASONAM '17* (pp. 232–239). New York, NY, USA: ACM.

Sboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., & Moloshnikov, I. (2016). Machine learning models of text categorization by author gender using topic-independent features. *Procedia Computer Science, 101*, 135–142.

Sisodia, D. S. (2019). Ensemble learning approach for clickbait detection using article headline features. *Informing Science: The International Journal of an Emerging Transdiscipline, 22*, 031–044.

Song, J., Lee, S., & Kim, J. (2015). Crowdtarget: Target-based detection of crowdturfing in online social networks. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 793–804). ACM.

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv:1704.07506.

Tan, E. E. G., & Ang, B. (2017). Clickbait: Fake news and role of the state,.

Vijgen, B., et al. (2014). The listicle: An exploring research on an interesting shareable new media phenomenon. *Studia Universitatis Babes-Bolyai-Ephemerides, 59*(1), 103–122.

Vithlani, P., & Kumbharana, C. (2015). Comparative study of character recognition tools. *International Journal of Computer Applications, 118*(9).

Wang, X., Jiang, W., & Luo, Z. (2016). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 2428–2437).

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. In *Proceedings of the 22nd international conference on world wide web* (pp. 1445–1456). ACM.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv:1611.03530.