Article

JIS

Journal of Information Science
2021, Vol. 47(1) 118–128
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0165551519871822
journals.sagepub.com/home/jis
SAGE

Clickbait detection using multiple categorisation techniques

Abinash Pujahari
Computer Science & Engineering, National Institute of Technology, Raipur, India

Dilip Singh Sisodia
Computer Science & Engineering, National Institute of Technology, Raipur, India

Abstract

Clickbaits are online articles with deliberately designed misleading titles for luring more and more readers to open the intended web page. Clickbaits are used to tempt visitors to click on a particular link either to monetise the landing page or to spread the false news for sensationalisation. The presence of clickbaits on any news aggregator portal may lead to unpleasant experience to readers. Automatic detection of clickbait headlines from news headlines has been a challenging issue for the machine learning community. A lot of methods have been proposed for preventing clickbait articles in recent past. However, the recent techniques available in detecting clickbaits are not much robust. This article proposes a hybrid categorisation technique for separating clickbait and non-clickbait articles by integrating different features, sentence structure and clustering. During preliminary categorisation, the headlines are separated using 11 features. After that, the headlines are recategorised using sentence formality and syntactic similarity measures. In the last phase, the headlines are again recategorised by applying clustering using word vector similarity based on *t*-stochastic neighbourhood embedding (*t*-SNE) approach. After categorisation of these headlines, machine learning models are applied to the dataset to evaluate machine learning algorithms. The obtained experimental results indicate that the proposed hybrid model is more robust, reliable and efficient than any individual categorisation techniques for the dataset we have used.

Keywords

Classification; clickbait; clustering; sentence structure; word vector

1. Introduction

The use of online news media has increased these days rapidly due to the excessive use of the Internet. These are very useful for users in gaining knowledge and information at any time; however, sometimes these websites create frustration and waste the time of users by providing altered content than the news headlines. These days, most of the websites are using unwanted 'Advertisements/News' kind of things to make money out of it. One of the simplest examples is the usage of 'clickbait' [1–3] headlines. These are the headlines that frequently appear on news websites that attract users and force them to click on those headlines so that the website can earn money from users' clicks [4,5]. The information present in these headlines creates suspense and can tease users by containing exaggerate information than the actual content [6,7]. The main aim of these (clickbait) headlines is to lure users to click on the headlines. Finally, it causes a lot of frustrations for the users. Some of the common clickbait headline examples are given in Table 1.

A lot of work has been done to combat the clickbait titles. Some tools are available in different leading media sites, which automatically block such articles [8,9]. Bauhaus-Universität Weimar organised a clickbait challenge[1] to detect clickbait by providing their datasets, which draws a lot of attraction in this domain of research. However, the problem is that the structure of these headlines is quite similar to non-clickbait headlines, which causes a problem. This article aims to provide an efficient method to categorise the clickbait and non-clickbait articles using semantic analysis and validate using machine learning classification methods. The main contributions of this article are as follows:

Corresponding author:
Dilip Singh Sisodia, Computer Science & Engineering, National Institute of Technology, Raipur, GE Road, Raipur 492010, Chhattisgarh, India.
Email: dssisodia.cs@nitrr.ac.in

Table 1. Example of clickbait headlines.

Headlines	Description
'Man tries to hug a wild lion; you won't believe what happens next'.	These kinds of headlines seem to be shocking, amazing and unbelievable which generates curiosity among users.
'Remember the girl played the role of "Nikita" in the movie "Koi Mil Gaya?"' This is how she looks now! Absolutely hot!	These kinds of 'celebrity gossip' headlines are teasing contents which force users to click on the headlines.
'Only the people with an IQ above 160 can solve these questions. Are you one of them? Click to find out ...'	These headlines make a challenge to our IQ, which creates anxiety to explore, but the content may be different.

IQ: Intelligence Quotient.

- Categorising titles into 'clickbait' and 'non-clickbait' using document formality measures like *F*-score [10] and Coh-Metrix [11], because most of the clickbait articles have poor sentence structures [12];
- Using the word-to-vector scheme for finding the similarity among the texts between both the categories (clickbait and non-clickbait) for proper categorisation;
- Building and validating a hybrid model using the above categorisation techniques for the detection of clickbait headlines using different machine learning algorithms.

This article is organised as follows. In Section 2, we will discuss some related works/methodologies available for detecting clickbait headlines. Section 3 contains the proposed methodology used in this article to detect the clickbait article. The experimental results and discussion of the proposed methodology are given in Section 4 along with the performance of different machine learning algorithms for detecting clickbait headlines. Finally, Section 5 discusses the general outcome of this article along with the future scope of research in this field.

2. Related works

Some of the related works in the domain of clickbait detection are described in detail in this section as well as their limitations and possible extensions.

Chakraborty et al. [13] proposed a method for detecting clickbait articles and also built a browser add-on for detecting clickbaits. They have collected non-clickbait articles from 'Wikinews', and for the clickbait article, they followed several domains like BuzzFeed and ViralNova. They carried out a linguistic analysis on the dataset collected using the 'Stanford CoreNLP' [14] tool. They primarily focused on the sentence structure (i.e. length of the headline/words, hyperbolic words, Internet slangs, common phrases, determiners) to categorise the clickbait and non-clickbait headlines. They have used four feature selection techniques: sentence structure, word patterns, clickbait language and *n*-gram features. Finally, they have compared the classification of articles using three machine learning algorithms: support vector machine (SVM), decision tree and random forest.

Another work was done by Biyani et al. [15] to detect clickbait, in which they first categorised the clickbait headlines into eight different categories (exaggeration, teasing, inflammatory, formatting, graphic, bait-and-switch, ambiguous, wrong). According to them, these eight are the most common categories of clickbait headlines. They have used document informality measures and reading difficulty of the text to determine whether a headline is a 'clickbait' or not. They applied the 'gradient boosted decision trees' [16] classification algorithm to the dataset using four features (similarity, URL, content and informality and forward reference) and tested them individually using test data. They have also made the comparison of the density of clickbait and non-clickbait headlines. The performance is acceptable in this article for most of the dataset, but the individual features used are not enough to obtain the desired outcome.

Rony et al. [17] extended the prevention of 'clickbait' to a social media platform. They have used the skip-gram model [18] to use word embeddings, which is further used to find out the similarity between the texts. They have made the comparison of their pre-trained vectors with the Google News dataset. They compared the results with earlier findings in the field of clickbait detection with and without pre-trained vectors. They have also categorised the percentage of different media in terms of clickbait and non-clickbait. They have used the headline–body similarity for categorising the clickbait article.

The main problem of clickbait detection is the categorisation of headlines, because earlier research suggests that sentences having the similar kind of structure and context can fall into either of the categories (clickbait and non-clickbait). The next section describes the proposed method for detecting clickbait headlines.

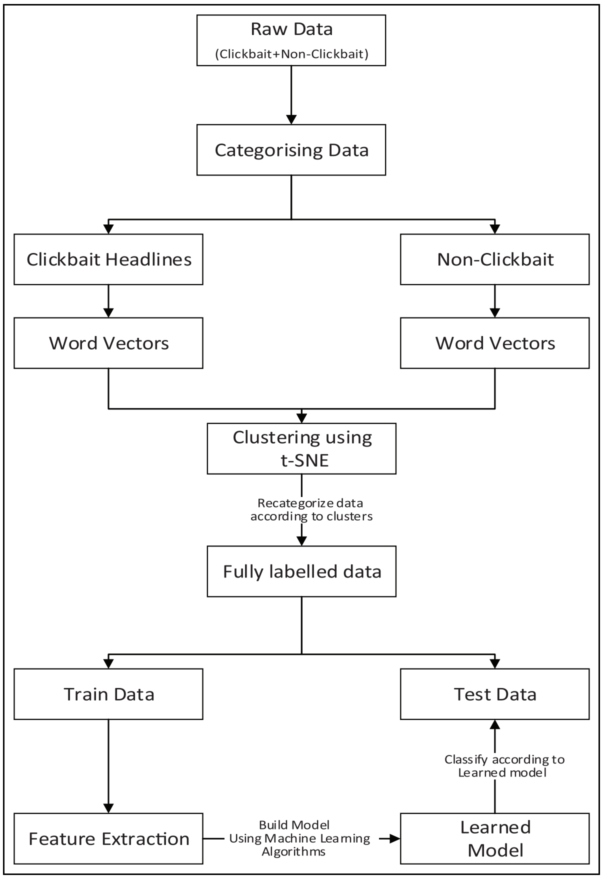

Figure 1. The proposed model for clickbait classification.

Table 2. Categories of clickbait headlines and their examples.

Category	Definition	Example
Incomplete	The title is incomplete in conveying the message	'Click here, and you will get ...'
Headline cloning	Copying of actual headline for different content	Headlines have a different structure but the same text as normal headlines
URL redirection	The headlines land on a false page than promised	Invalid URLs having false domain information (e.g. http://xyz.by)

3. Methods

To improve the quality of the clickbait detection techniques discussed earlier in Section 2, we proposed a model as presented in Figure 1 for clickbait prevention. The dataset is obtained from Chakraborty et al. [13] which contains news headlines of both kinds (clickbait and non-clickbait). The dataset is thoroughly evaluated using the proposed method. After categorisation of the data into two classes, they are evaluated using the machine learning algorithms, which are described later.

3.1. Categorisation of headlines

The proposed model in this article extends the categorisation of clickbait headlines by Biyani et al. [15], which adds more constraints to categorisation. The proposed constraints are given in Table 2. This helps in identifying the clickbait and non-clickbait articles more precisely. Furthermore, we would like to mention that the categories mentioned in Table 2 are not the subsets of any of the categories mentioned by Biyani et al. These are the constraints we applied in addition to the eight constraints available in Biyani et al. [15]. The purpose of this addition is to classify a headline to which category it belongs more correctly. After adding these constraints, we are getting almost 10% of the data from the dataset which were recategorised, which is analysed later in this article, which improves the performance of categorisation. Some of the headlines are falling into more than one category, which is trivial. The major category out of the three given in Table 2 is 'incomplete' because it always creates suspense among users and forces them to click on the headline. Out of these three, the headline cloning is very difficult to detect, because the normal headlines and the 'clickbait' are almost the same. To detect this kind of clickbaits, we parsed the body of the landing document and compared the keywords with the headlines in order to determine whether it is a clickbait or not.

3.2. Sentence formality and structure

To find out the sentence formality of the headlines, we have used two measures: the first one is *F*-score [10] which is also used by Biyani et al. [15] and the second one is Coh-Metrix. *F*-score, which is calculated using equation (1), has a value of 0%–100%, that is, the higher the value, the more formal the language is. The threshold percentage for our experiment is 60% of the *F*-score value. We have taken the intersection part between these two schemes to determine the well-formed headlines, because the 'clickbait' headlines are poorly formed. To find out the terms listed in equation (1) and analysis of words in the text, we have used the Stanford CoreNLP [14] language tool

$$F-\text{score} = \frac{\begin{pmatrix} \text{noun freq.} + \text{adjective freq.} + \text{preposition freq.} + \text{article freq.} \\ -\text{pronoun freq.} - \text{verb freq} - \text{adverb freq} - \text{interjection freq} + 100 \end{pmatrix}}{2} \tag{1}$$

Coh-Metrix [11] is used to find out the ambiguous words, syntactic complexity, word ratio, readability, co-reference cohesion and so on. After using these two measures, two refined sets of headlines are created (clickbait and non-clickbait). Coh-Metrix^2 is an online tool for assessing the features mentioned above from a piece of document. The headlines we collected are evaluated online using this tool to separate clickbait headlines and normal (non-clickbait) headlines. This feature is used along with *F*-score because it assesses the document in multiple levels.

This article also uses the Flesch–Kincaid grade level [19] which is used for readability test for different text reading applications. In general practice, the 'clickbaits' are more difficult to be read, due to their structure, whereas the normal headlines are easier to be read. The formula for the Flesch reading ease score (FRES) [19] test is given in equation (2)

$$Score = 206.835 - 1.015\left(\frac{total\,words}{total\,sentences}\right) - 84.6\left(\frac{total\,syllables}{total\,words}\right) \tag{2}$$

The value of the score is in between 0 and 100, where the higher value indicates the easier readability of the text. Hence, the headlines having high scores are generally normal headlines, and low-score headlines can be 'clickbait'. The threshold value for the normal headlines using equation (2) is above 60 in our experiment to distinguish between clickbait and non-clickbait.

3.3. Recategorisation of headlines using clustering

Using the processes mentioned above, the raw data obtained are divided into two categories. However, still, the data are noisy since the volume of the data is very large. The next step in this work is to recategorise the headlines. For this, we use the word representation in the vector space [20] using the word-to-vector scheme. After converting the words into the vector format, we have used the *t*-stochastic neighbourhood embedding (*t*-SNE) [21] algorithm to create two clusters, where similar words are grouped together. This is a dimension reduction approach to categorise high-dimensional data. The *t*-SNE method is proposed by Van Der Maaten et al., which is an extension to stochastic neighbourhood embedding (SNE), proposed by Hinton and Roweis [22]. The SNE method uses probability distribution over pairs of high-dimensional objects such that similar objects have a high probability of being grouped, while dissimilar objects have a very small probability of being grouped. For calculating the similarity between two objects, they calculated the conditional probability using equation (3) [22], where $P(x(j)|x(i))$ is the conditional probability that $x(i)$ and $x(j)$ are treated as neighbours

Table 3. Precision of different clickbait categories.

S. No.	Category	No. of examples classified as clickbait	Precision (%)
1	Ambiguous	645	47.81
2	Exaggeration	4954	45.86
3	Inflammatory	1023	52.34
4	Bait-and-switch	536	65.82
5	Teasing	5278	59.81
6	Formatting	789	49.63
7	Wrong	152	41.23
8	Graphic	365	40.78
9	Incomplete	678	48.64
10	Headline cloning	831	51.56
11	URL redirection	1345	53.96

$$P(x(j)|x(i)) = \frac{\exp\left(\frac{-\|x(i)-x(j)\|^2}{2\sigma_i^2}\right)}{\sum\limits_{k\neq i}\exp\left(\frac{-\|x(i)-x(k)\|^2}{2\sigma_i^2}\right)} \tag{3}$$

where σ_i is known as Gaussian variance which is centred on the object $x(i)$. t-SNE is a different method for embedding objects by overcoming the 'crowding problem' and uses t-distribution rather than Gaussian distribution as proposed by Van Der Maaten et al. Before applying these procedures, we have converted the data into word vectors using the word vector embedding technique. The results of the neighbourhood embedding are discussed in the next section.

4. Experimental results and discussion

In this section, we first described the dataset used for the experimental verification of the proposed model followed by the detailed results obtained using different classifiers. We have also compared the results obtained using the proposed method with those obtained by another model.

4.1. Dataset

We have used the same dataset[3] as that used by Chakraborty et al. [13]. The dataset contains two files: one contains clickbait headlines and the other contains non-clickbait headlines. Each file consists of around 16,000 articles. Chakraborty et al. extracted non-clickbait headlines from trusted sources like 'Wikinews', 'The New York Times', 'The Guardian' and 'The Hindu' and clickbait headlines from 'BuzzFeed', 'Upworthy' and so on. The dataset is balanced, that is, the sizes of clickbait and non-clickbait articles are almost the same. We have also used the source code provided by Chakraborty et al. to compare their results with our results.

4.2. Results

The dataset discussed in Section 4.1 is categorised between normal text and texts having marks like punctuation, exclamation and question because these are often present in clickbait headlines. After doing this, the documents are categorised into clickbait and non-clickbait using the features discussed in Section 3.1 by comparing the features of the headlines. The clickbait articles are grouped into a total of 11 categories. The numbers of headlines falling into these categories are given in Table 3. One headline can fall into more than one category due to the sentence structure and text format. Hence, the total number of headlines falling into each category exceeds the total number of clickbait headlines. The categories with serial numbers 1–8 were taken from Biyani et al. [15], and the rest are proposed in this article. The precision of each category is also mentioned in Table 3. The headlines not falling into these categories are grouped as non-clickbait headlines during this phase of categorisation.

After categorising the headlines into clickbait and non-clickbait using the features mentioned in Table 3, we refined the grouping using the document formality measures as discussed in Section 3.2 using F-score and score measures using equations (1) and (2), respectively.

After grouping the headlines into two categories, word embedding vector is generated for both types of headlines. Then the words are converted to word vectors using word2vec using MATLAB Text Analytics Toolbox.[4] After

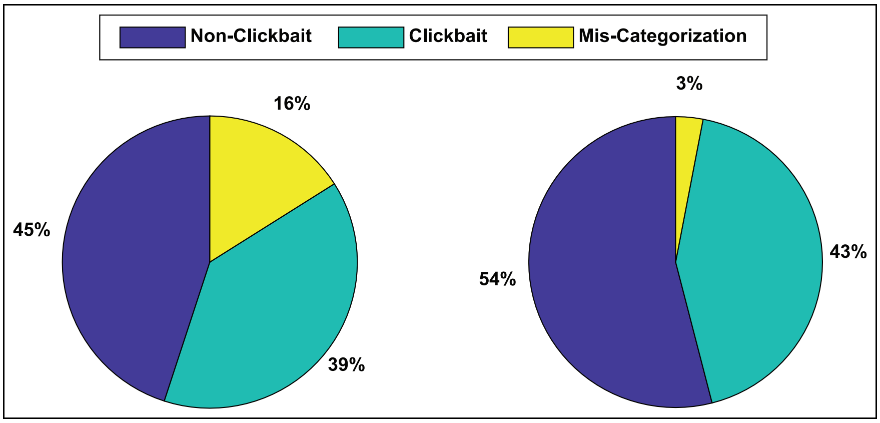

Figure 2. Percentage of clickbait and non-clickbait articles after clustering.

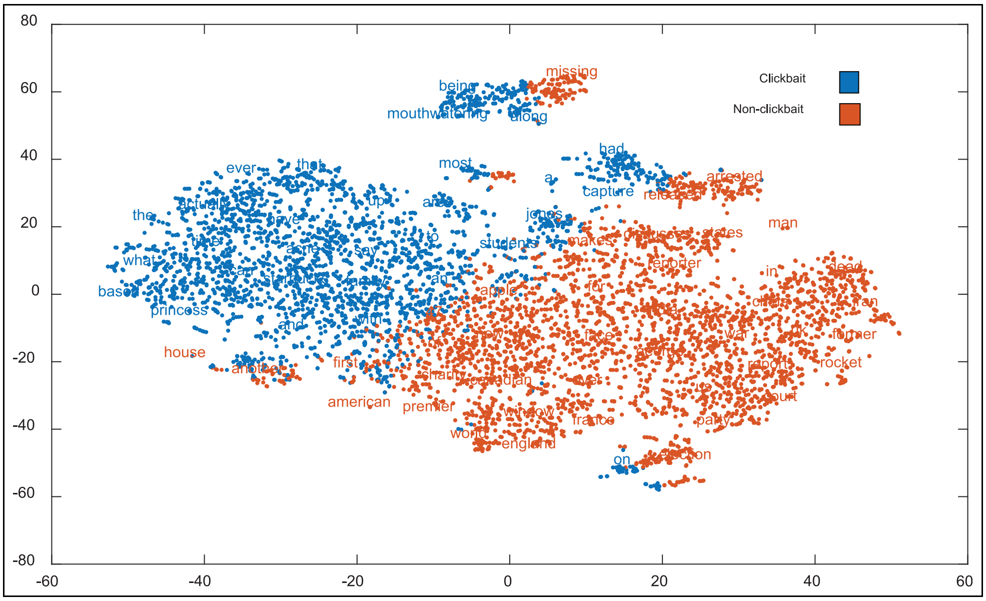

Figure 3. Clustering of headlines using *t*-SNE.

converting to the vector format, the dataset is applied through clustering using the *t*-distributed SNE technique as discussed in Section 3. We can clearly see from Figure 2 that more numbers of examples are grouped into either of the categories (i.e. clickbait or non-clickbait). This makes the two classes (clickbait and non-clickbait headlines) more robust and less noisy. Furthermore, it can be seen from the figure that the wrong categorisation of headlines is less after using word vector clustering. A wrong categorisation is that if a headline is clickbait and it is categorised as non-clickbait and vice versa. The two different clusters (clickbait and non-clickbait headlines) are represented in text scattered plots in Figure 3. From this figure, we can observe that there is very less noise in the data after categorisation and word vector clustering. These data are now ready for classification and model learning procedures.

4.3. Classification of dataset using learning algorithms

After going through rigorous categorisation of headlines, now a model is built using the learning algorithm so that it can predict the unseen examples in the future. We have used the word vectors as our features for training purposes. The classification algorithms used in this article are 'SVM' [23], 'decision tree (C4.5)' [24] and 'random forest' [25]. More description about these classifiers is given below. We have used the 10-fold cross-validation scheme to test the efficiency

Table 4. Performance of different classifiers with respect to features.

Features	Decision tree			SVM			Random forest		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Based on categories	0.86	0.89	0.87	0.92	0.94	0.90	0.88	0.92	0.89
Based on structures	0.84	0.89	0.81	0.90	0.92	0.89	0.89	0.91	0.88
Based on categories + word vector clustering	0.91	0.92	0.89	0.95	0.95	0.95	0.93	0.94	0.91
Based on structures + word vector clustering	0.90	0.91	0.88	0.93	0.84	0.93	0.91	0.91	0.90
All features (proposed)	**0.92**	**0.93**	**0.92**	**0.97**	**0.97**	**0.96**	**0.94**	0.94	**0.93**
Chakraborty et al. [13]	0.90	0.91	0.89	0.93	0.95	0.90	0.92	0.94	0.91

Note: Bold represents best results.
SVM: support vector machine.

of the classifiers. The results were first evaluated individually using different features and then evaluated after integrating all the features. The evaluation criteria used for the classifiers are accuracy, precision and recall. The receiver operating characteristic (ROC) curves for the respective classifiers are also generated to determine the relationship between true-positive rates and false-positive rates.

4.3.1. SVM. It comes under the supervised learning techniques, where each example is treated as a point in space and separated by a boundary (known as hyperplane) [23]. The boundary tries to separate different categories (classes) of examples. New examples are then mapped in the same space to determine in which side they fall, and the corresponding category will be assigned to them. The hyperplane is chosen so that the distance from the nearest data point of any class to the hyperplane is maximum. SVM uses some mathematical functions which are known as 'kernels'. The job of the kernel is to take data as input and convert them into the required form. Several kernel functions are linear, non-linear, polynomial, Gaussian, radial basis function (RBF) and sigmoid. We have used the linear kernel during classification.

4.3.2. Decision tree (C4.5). It is an algorithm to build a decision tree [24], which can be used for the classification. C4.5 builds a classification tree in a similar way as ID3, but has several improvements, namely, it is capable of handling continuous and discrete attributes, can handle training data with missing attribute values and can prune tree after generation. The splitting criterion used in C4.5 is *gain ratio*, that is, whichever attribute value has the highest *gain ratio* value will be selected as the splitting node.

4.3.3. Random forest (C4.5). It comes under the ensemble learning methods [26,27] for classification/regression. Random forest [25], as the name suggests, builds multiple decision trees during the training process. During prediction, we will select the class label that is mostly suggested by those decision trees. The main advantage of this approach is to avoid overfitting. Here the learning of each tree inside the forest can be any decision tree classifier algorithm (e.g. ID3, C4.5, CART, etc.).

The performance of the aforementioned classifiers is given in Table 4. It can be seen clearly from the table that the performance of individual features is not good enough, but when integrating all the features the results are quite acceptable for the data used for clickbait headline detection. The performance results of the individual features are comparably less due to the inconsistencies in the data obtained from different online media, because the same headlines can have different structures, different semantics and different sentence formation. Hence, by integrating all the features, we can obtain good results. The confusion matrices of the respective classifiers are shown in Figure 4, where '0' represents the clickbait class and '1' represents the 'non-clickbait' class. Also, from the confusion matrices, we can observe that the classifiers are able to detect clickbait and non-clickbait headlines equally, due to the multiple categorisation techniques. The ROC curves obtained for the 'clickbait' class using different classifiers are also shown in Figure 5. We obtained the area under the curve (AUC) values of 0.94, 0.96 and 0.99 for decision tree, random forest and SVM, respectively, which outperformed the results reported by Chakraborty et al. [13] in their work. The comparisons given in Table 4 were obtained under the same evaluation conditions (i.e. the same set of training data and test data). We can see that the performance of the proposed method is better for all the classifiers as compared with the method proposed by Chakraborty et al. Figure 6 presents the AUC graphs for the respective classifiers. When comparing the classifiers, SVM is

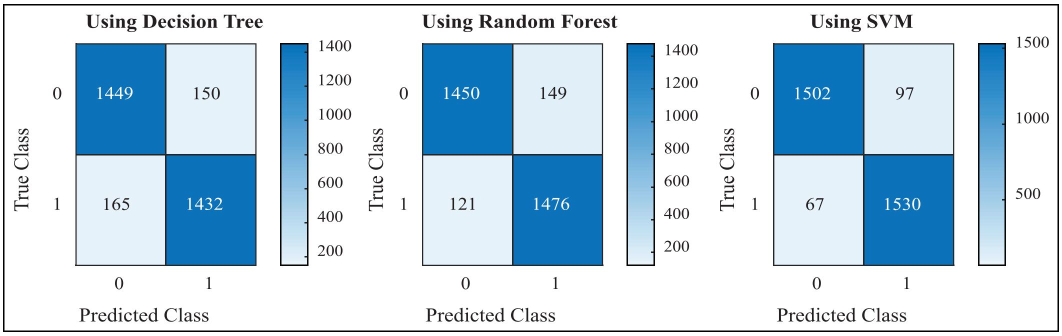

Figure 4. Confusion matrices along with heatmap for different classifiers.

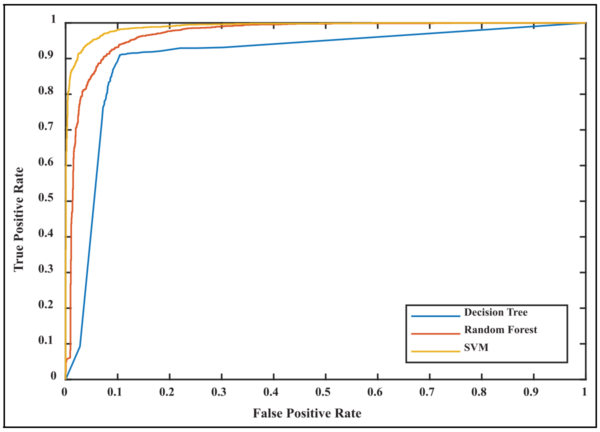

Figure 5. ROC curves using different classifiers for the 'clickbait' class.

performing better than the other two classifiers in clickbait headline detection, which can be clearly seen from the confusion matrices as well as ROC curves. SVM performs well on text classification using the linear kernel because text data contain a lot of features and are linearly separable most of the times. Since the dataset we have used is a two-class problem, SVM performs better than the other classifiers used in the experiment.

4.4. Reliability test of the model for detecting clickbait

Alexandru and Caruana [28] in their paper proposed a model for predicting good probabilities for supervised learning. They also proposed two calibration methods for correcting the distortions generated by the bias for different classifiers. Using the same procedure, we examined the relationship between the predictions made by the learning methods mentioned in Section 4.3 with true posterior probabilities. Reliability graph allows us to check if the predicted probabilities of a binary classifier are well calibrated. For the reliability test, we have obtained the true posterior probabilities of each classification model from their predicted values. Then we have followed the Platt calibration technique as discussed in Niculescu-Mizil and Caruana [28]. Since our classification problem is binary, we can test the classifiers using reliability graph. In reliability graph, the curve should be as close as possible to the diagonal/identity. The respective reliability graphs are presented in Figure 7 for different classifiers. As we can see from the graph, the curve is closer to the diagonal in the case of SVM classifiers as compared with the other two classifiers.

5. Conclusion

Clickbait headlines have become a major issue in online news media. Hence, automated prevention is necessary. We can conclude from the experimental results that only one categorisation technique is not efficient enough to combat clickbait articles. The results are quite acceptable by integrating more than one feature. Furthermore, the websites are changing

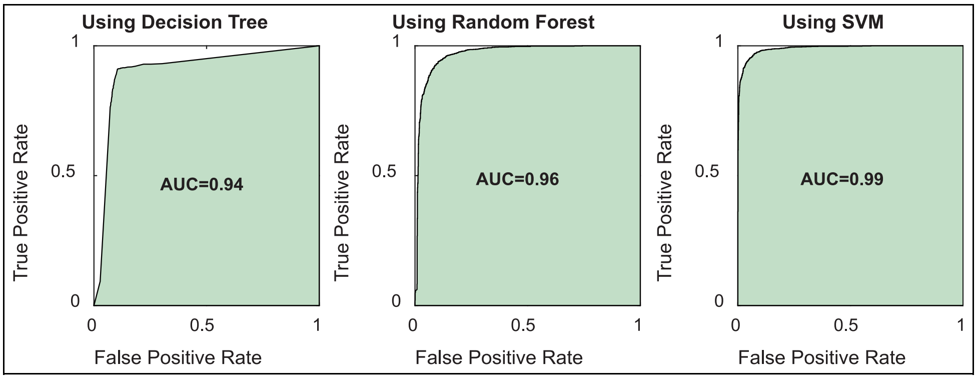

Figure 6. AUC values for different classifiers.

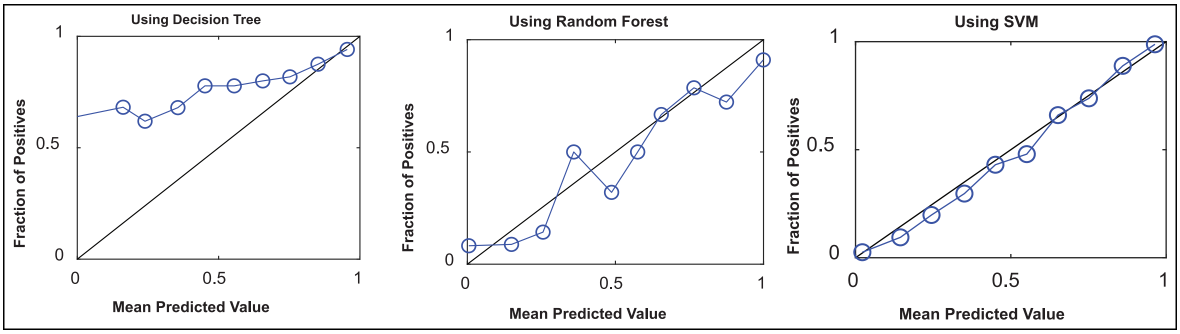

Figure 7. Reliability graph of different classifiers by integrating all the features for detecting clickbait.

their strategies with respect to time also. Hence, the clickbait manufacturer will also find out a way to bypass the categorization of clickbait articles. Thus, a robust detection technique should be built that can predict future changes in the detection procedure by considering the time domain features. Also, the websites are using graphical images as clickbait headlines, which cannot be detected using text processing. Hence, by applying some image processing and pattern recognition schemes, we can detect the same after extracting the sentences. These can be the future scope of research in this problem.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

ORCID iD

Dilip Singh Sisodia https://orcid.org/0000-0001-9845-290X

Notes

1. www.clickbait-challenge.org
2. Coh-Metrix tool is available at: http://cohmetrix.com/
3. https://github.com/bhargaviparanjape/clickbait/tree/master/dataset
4. Text Analytics Toolbox MATLAB is available at: https://in.mathworks.com/products/text-analytics.html

References

[1] Chen Y, Conroy NJ and Rubin VL. Misleading online content: recognizing clickbait as false news. In: *Proceedings of the 2015 ACM on workshop on multimodal deception detection*, Seattle, WA, 13 November 2015, pp. 15–19. New York: ACM Digital Library.

[2] Potthast M, Köpsel S, Stein B et al. Clickbait detection. In: *European conference on information retrieval*, Padua, 20–23 March 2016, pp. 810–817, Springer.

[3] Bourgonje P, Moreno Schneider J and Rehm G. From clickbait to fake news detection: an approach based on detecting the stance of headlines to articles. In: *Proceedings of the 2017 EMNLP workshop: natural language processing meets journalism*, Copenhagen, 7 September 2017, pp. 84–89. Stroudsburg, PA: Association for Computational Linguistics.

[4] Hurst N. *To clickbait or not to clickbait? An* examination of clickbait headline effects on source credibility. Master's Thesis, University of Missouri, Columbia, MO, 2016.

[5] Agrawal A. Clickbait detection using deep learning. In: *2016 2nd international conference on next generation computing technologies (NGCT)*, Dehradun, India, 14–16 October 2016, pp. 268–272. New York: IEEE.

[6] Hoffmann C. What is clickbait? (Check all that apply). In: Hoffmann C (ed.) *Stupid humanism*. Berlin: Springer, 2017, pp. 109–128.

[7] Zannettou S, Chatzis S, Papadamou K et al. The good, the bad and the bait: detecting and characterizing clickbait on YouTube. In: *IEEE symposium on security and privacy workshops (SPW 2018)*, San Francisco, CA, 24 May 2018, pp. 63–69. New York: IEEE.

[8] López-Sánchez D, Herrero JR, Arrieta AG et al. Hybridizing metric learning and case-based reasoning for adaptable clickbait detection. *Appl Intell* 2018; 48(9): 2967–2982.

[9] Kumar V, Khattar D, Gairola S et al. Identifying clickbait: a multi-strategy approach using neural networks. In: *proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval*, Ann Arbor, MI, 8–12 July 2018, pp. 1225–1228. New York: ACM.

[10] Heylighen F and Dewaele J. Formality of language?: definition, measurement and behavioral determinants. *Interner Bericht* 1999; 1999: 38.

[11] Graesser AC, McNamara DS, Louwerse MM et al. Coh-Metrix: analysis of text on cohesion and language. *Behav Res Methods Instruments Comput* 2004; 36(2): 193–202.

[12] Finnis A. List of bad words. *BuzzFeed*, 29 October 2015, https://www.buzzfeed.com/alexfinnis/the-100-most-brilliantly-british-swear-words-in-existence

[13] Chakraborty A, Paranjape B, Kakarla S et al. Stop clickbait: detecting and preventing clickbaits in online news media. In: *Proceedings of 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, San Francisco, CA, 18–21 August 2016, pp. 9–16. New York: IEEE.

[14] Manning C, Surdeanu M, Bauer J et al. The Stanford CoreNLP natural language processing toolkit. In: *Proceedings of 52nd annual meeting of the Association for Computational Linguistics: system demonstration*, Baltimore, MD, 23–24 June 2014, pp. 55–60. Stroudsburg, PA: Association for Computational Linguistics.

[15] Biyani P, Tsioutsiouliklis K and Blackmer J. '8 amazing secrets for getting more clicks': detecting clickbaits in news streams using article informality clickbait classification. In: *Proceedings of the 30th AAAI conference on artificial intelligence (AAAI-16)*, Phoenix, AZ, 12–17 February 2016, pp. 94–100. Menlo Park, CA: AAAI.

[16] Friedman JH. Greedy function approximation?: a gradient boosting machine. *Ann Stat* 2001; 29(5): 1189–1232.

[17] Rony MMU, Hassan N and Yousuf M. 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*, Sydney, NSW, Australia, 31 July–3 August 2017, pp. 232–239. New York: ACM.

[18] Bojanowski P, Grave E, Joulin A et al. Enriching word vectors with subword information, 2016, https://arxiv.org/abs/1607.04606

[19] Kincaid JP, Fishburne RPJr, Rogers RL et al. *Derivation of new readability formulas (automated readability index, fog count and Flesch reading ease formula) for navy enlisted personnel*. Research branch report 8-75, 1 January 1975. Millington, TN: Institute for Simulation and Training, University of Central Florida.

[20] Mikolov T, Chen K, Corrado G et al. Efficient estimation of word representations in vector space, 2013, https://arxiv.org/abs/1301.3781

[21] Van Der Maaten LJP and Hinton GE. Visualizing high-dimensional data using t-SNE. *J Mach Learn Res* 2008; 9: 2579–2605.

[22] Hinton GE and Roweis ST. Stochastic neighbor embedding. In: Thrun S and Saul LK and Scholkopf B (eds) *Advances in neural information processing systems*. Cambridge, MA: The MIT press, 2003, pp. 857–864.

[23] Cortes C and Vapnik V. Support vector machine. *Mach Learn* 1995; 20(3): 273–297.

[24] Quinlan JR. *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1993.

[25] Breiman L. Random forests. *Mach Learn* 1999; 45(5): 1–35.

[26] Rahman A and Tasnim S. Ensemble classifiers and their applications?: a review. *Int J Comput Trends Technol* 2014; 10(1): 31–35.

[27] Dietterich TG. Ensemble methods in machine learning. In: Dietterich TG (ed.) *Multiple classifier systems*, vol. 1857. Berlin: Springer, 2000, pp. 1–15.

[28] Niculescu-Mizil A and Caruana R. Predicting good probabilities with supervised learning. In: *Proceedings of the 22nd international conference on machine learning*, Bonn, 7–11 August 2005, pp. 625–632. New York: ACM.