Identifying Clickbait: A Multi-Strategy Approach Using Neural Networks[*]

Vaibhav Kumar
International Institute of Information
Technology Hyderabad
vaibhav.kumar@research.iiit.ac.in

Dhruv Khattar
International Institute of Information
Technology Hyderabad
dhurv.khattar@research.iiit.ac.in

Siddhartha Gairola
International Institute of Information
Technology Hyderabad
siddhartha.gairola@research.iiit.ac.in

Yash Kumar Lal[†]
Manipal Institute of Technology,
Manipal
yash.kumar4@learner.manipal.edu

Vasudeva Varma
International Institute of Information
Technology Hyderabad
vv@iiit.ac.in

ABSTRACT

Online media outlets, in a bid to expand their reach and subsequently increase revenue through ad monetisation, have begun adopting clickbait techniques to lure readers to click on articles. The article fails to fulfill the promise made by the headline. Traditional methods for clickbait detection have relied heavily on feature engineering which, in turn, is dependent on the dataset it is built for. The application of neural networks for this task has only been explored partially. We propose a novel approach considering all information found in a social media post. We train a bidirectional LSTM with an attention mechanism to learn the extent to which a word contributes to the post's clickbait score in a differential manner. We also employ a Siamese net to capture the similarity between source and target information. Information gleaned from images has not been considered in previous approaches. We learn image embeddings from large amounts of data using Convolutional Neural Networks to add another layer of complexity to our model. Finally, we concatenate the outputs from the three separate components, serving it as input to a fully connected layer. We conduct experiments over a test corpus of 19538 social media posts, attaining an F1 score of 65.37% on the dataset bettering the previous state-of-the-art, as well as other proposed approaches, feature engineering or otherwise.

KEYWORDS

Clickbait, Neural Network, Attention-Mechanism, Siamese Network, Text Embeddings, Image Embeddings

---

[*]First four authors have equal contribution.

[†]The author was in intern at International Institute of Information Technology Hyderabad when this work was done.

---

1 INTRODUCTION

The Internet provides instant access to a wide variety of online content, news included. Formerly, users had static preferences, gravitating towards their trusted sources, incurring an unwavering sense of loyalty. The same cannot be said for current trends since users are likely to go with any source readily available to them.

In order to stay in business, news agencies have switched, in part, to a digital front. Usually, they generate revenue by (1) advertisements on their websites, or (2) a subscription based model for articles that might interest users. However, since the same information is available via multiple sources, no comment can be made on the preference of the reader. To lure in more readers and increase the number of clicks on their content, subsequently increasing their agency's revenue, writers have begun adopting a new technique - clickbait.

The concept of clickbait is formalised as something to encourage readers to click on hyperlinks based on snippets of information accompanying it, especially when those links lead to content of dubious value or interest. Clickbaiting is the intentional act of over-promising or purposely misrepresenting - in a headline, on social media, in an image, or some combination - what can be expected while reading a story on the web. It is designed to create and, consequently, capitalise on the Loewenstein information gap [11]. Sometimes, especially in cases where such headlines are found on social media, the links can redirect to a page with an unoriginal story which contains repeated or distorted facts from the original article itself.

Our engine is built on three components. The first leverages neural networks for sequential modeling of text. Article title is represented as a sequence of word vectors and each word of the title is further converted into character level embeddings. These features serve as input to a bidirectional LSTM model. An affixed attention layer allows the network to treat each word in the title in a differential manner. The next component focuses on the similarity between the article title and its actual content. For this, we generate Doc2Vec embeddings for the pair and act as input for a Siamese net, projecting them into a highly structured space whose geometry

reflects complex semantic relationships. The last part of this system attempts to quantify the similarity of the attached image, if any, to the article title. Finally, the output of each component is concatenated and sent as input to a fully connected layer to generate a score for the task.

2 RELATED WORK

The task of automating clickbait detection has risen to prominence fairly recently. Initial attempts for the same have worked on (1) news headlines, and (2) heavy feature engineering for the particular dataset. [3]'s work is one of the earliest pieces of literature available in the field, focusing on an aggregation of news headlines from previously categorised clickbait and non-clickbait sources. Apart from defining different types of clickbait, they emphasise on the presence of language peculiarities exploited by writers for this purpose. These include qualitative informality metrics and use of forward references in the title to keep the reader on the hook. The first instance of detecting clickbait across social media can be traced to [15], hand-crafting linguistic features, including a reference dictionary of clickbait phrases, over a dataset of crowdsourced tweets [14]. However, [4] argued that work done specifically for Twitter had to be expanded since clickbait was available throughout the Internet, and not just social networks.

It was not until [1] that neural networks were tried out for the task as the authors used the same news dataset as [4] to develop a deep learning based model to detect clickbait. They used distributional semantics to represent article titles, and BiLSTM to model sequential data and its dependencies. Since then, [18] has also experimented with Twitter data [14] deploying a BiLSTM for each of the textual features (post-text, target-title, target-paragraphs, target-description, target-keywords, post-time) available in the corpus, and finally concatenating the dense output layers of the network before forwarding it to a fully connected layer. Since it was proposed in [2], the attention mechanism has been used for a variety of text-classification tasks, such as fake news detection and aspect-based sentiment analysis. [20] used a self-attentive BiGRU to infer the importance of tweet tokens in predicting the annotation distribution of the task.

One common point in all the approaches yet has been the use of only textual features available in the dataset. Our model not only incorporates textual features, modeled using BiLSTM and augmented with an attention mechanism, but also considers related images for the task.

3 MODEL ARCHITECTURE

In this section, we present our hybrid approach to clickbait detection. We first explain the three individual components followed by their fusion, which is our proposed model. These components are (1) BiLSTM with attention, (2) Siamese Network on Text Embeddings, and (3) Siamese Network on Visual Embeddings. An overview of the architecture can be seen in Figure 1.

We start with an explanation of the features used in the first component of the model.

Distributed Word Embeddings

Considering the effectiveness of distributional semantics in modeling language data, we use a pre-trained 300 dimensional Word2Vec
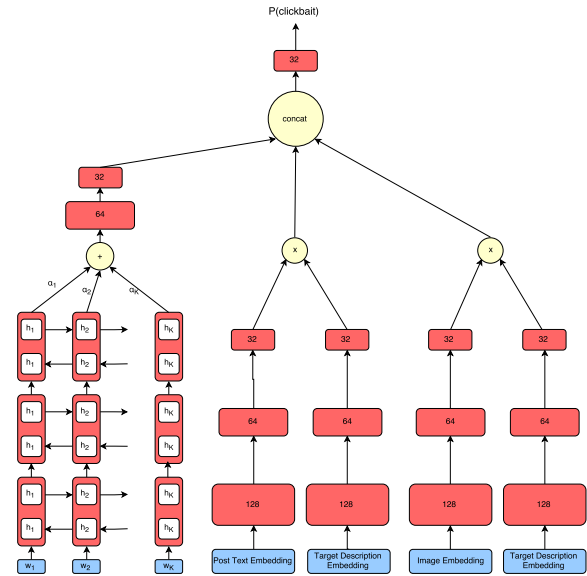

Figure 1: Model Architecture

[12] model trained over 100 billion words in the Google News corpus using the Continuous Bag of Words architecture. These map the words in a language to a high dimensional real-valued vectors to capture hidden semantic and syntactic properties of words, and are typically learned from large, unannotated text corpora. For each word in the title, we obtain its equivalent Word2Vec embeddings using the model described above.

Character Level Word Embeddings

Character level word embeddings [7] capture the orthographic and morphological features of a word. Apart from this, using them is a step toward mitigating the problem of out-of-vocabulary (OoV) words. In such a case, the word can be embedded by its characters using character level embedding. We follow [1] and first initialize a vector for every character in the corpus. The vector representation of each word is learned by applying 3 layers of a 1-dimensional Convolutional Neural Network [5] with ReLU non-linearity on each vector of character sequence of that word and finally max-pooling the sequence for each convolutional feature.

Document Embeddings

Doc2Vec [10] is an unsupervised approach to generate vector representations for slightly larger bodies of text, such as sentences, paragraphs and documents. It has been adapted from Word2Vec [12] which is used to generate vectors for words in large unlabeled corpora. The vectors generated by this approach come handy in tasks like calculating similarity metrics for sentences, paragraphs and documents. In sequential models like RNNs, the word sequence is captured in the generated sentence vectors. However, in Doc2Vec,

the representations are order independent. We use GenSim [16] to learn 300 dimensional Doc2Vec embeddings for each target description and post title available.

Pre-trained CNN Features

As seen in various visual understanding problems recently, image descriptors trained using Convolutional Neural Networks over large amounts of data such as ImageNet have proven to be very effective. The implicit learning of spatial layout and object semantics in the later layers of the network from very large datasets has contributed to the success of these features. We use a pre-trained network of VGG-19 architecture [17] trained over the ImageNet database (ILSVRC-2012) and extract CNN features. We use the output of the fully-connected layer (FC7), which has 4096 dimensions, as feature representations for our architecture.

We now go into detail about the components of the model, individual and combined, and how the parameters are learned.

3.1 Bidirectional LSTM with Attention

Recurrent Neural Network (RNN) is a class of artificial neural networks which utilizes sequential information and maintains history through its intermediate layers. A standard RNN has an internal state whose output at every time-step which can be expressed in terms of that of previous time-steps. However, it has been seen that standard RNNs suffer from a problem of vanishing gradients [9]. This means it will not be able to efficiently model dependencies and interactions between words that are a few steps apart. LSTMs are able to tackle this issue by their use of gating mechanisms. For each record in the dataset, the content of the post as well as the content of the related web page is available. We convert the words from the title of both attributes into the previously mentioned types of embeddings to act as input to our bidirectional LSTMs.

$(\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_R)$ represent forward states of the LSTM and its state updates satisfy the following equations:

$$[\overrightarrow{f_t}, \overrightarrow{i_t}, \overrightarrow{o_t}] = \sigma \left[\overrightarrow{W} \left[\overrightarrow{h}_{t-1}, \overrightarrow{r_t} \right] + \overrightarrow{b} \right] \tag{1}$$

$$\overrightarrow{l_t} = \tanh \left[\overrightarrow{V} \left[\overrightarrow{h}_{t-1}, \overrightarrow{r_t} \right] + \overrightarrow{d} \right] \tag{2}$$

$$\overrightarrow{c_t} = \overrightarrow{f_t} \cdot \overrightarrow{c}_{t-1} + \overrightarrow{i_t} \cdot \overrightarrow{l_t} \tag{3}$$

$$\overrightarrow{h_t} = \overrightarrow{o_t} \cdot \tanh(\overrightarrow{c_t}) \tag{4}$$

here σ is the logistic sigmoid function, $\overrightarrow{f_t}, \overrightarrow{i_t}, \overrightarrow{o_t}$ represent the forget, input and output gates respectively. $\overrightarrow{r_t}$ denotes the input at time t and $\overrightarrow{h_t}$ denotes the latent state, $\overrightarrow{b_t}$ and $\overrightarrow{d_t}$ represent the bias terms. The forget, input and output gates control the flow of information throughout the sequence. $\overrightarrow{W}$ and $\overrightarrow{V}$ are matrices which represent the weights associated with the connections.

$(\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_R)$ denote the backward states and its updates can be computed similarly.

The number of bidirectional LSTM units is set to a constant K, which is the maximum length of all title lengths of records used in training. The forward and backward states are then concatenated to obtain $(h_1, h_2, \ldots, h_K)$, where

$$h_i = \begin{bmatrix} \overrightarrow{h}_i \\ \overleftarrow{h}_i \end{bmatrix} \tag{5}$$

Finally, we are left with the task of figuring out the significance of each word in the sequence i.e. how much a particular word influences the clickbait-y nature of the post. The effectiveness of attention mechanisms have been proven for the task of neural machine translation [2] and it has the same effect in this case. The goal of attention mechanisms in such tasks is to derive context vectors which capture relevant source side information and help predict the current target word. The sequence of annotations generated by the encoder to come up with a context vector capturing how each word contributes to the record's clickbait quotient is of paramount importance to this model. In a typical RNN encoder-decoder framework [2], a context vector is generated at each time-step to predict the target word. However, we only need it for calculation of context vector for a single time-step.

$$c_{attention} = \sum_{j=1}^{K} \alpha_j h_j \tag{6}$$

where, $h_1, \ldots, h_K$ represents the sequence of annotations to which the encoder maps the post title vector and each α_j represents the respective weight corresponding to each annotation h_j. This component is represented on the leftmost in Figure 1.

3.2 Siamese Net with Text Embeddings

The second component of our model is a Siamese net [13] over two textual features in the dataset. Siamese networks are designed around having symmetry and it is important because it's required for learning a distance metric. We use them to find the similarity between the title of the record and its target description. The words in the title and in the target description are converted into their respective Doc2Vec embeddings and concatenated, after which they are considered as input into a Siamese network. A visual representation of this can be found in the middle of Figure 1.

3.3 Siamese Neural Network with Visual Embeddings

The final component of our hybrid model is also a Siamese net. However, it considers visual information available in the dataset, and sets our model apart from other approaches in this field. The relevance of the image attached to the post can be quantified by capturing its similarity with the target description. The VGG-19 architecture outputs a 4096 dimensional vector for each image which, in turn, is fed as input into a dense layer to convert each representation to a 300 dimensional vector. This serves as one input to the visual Siamese net. The target description is converted into its 300 dimensional vector representation by passing it through the pre-trained Doc2Vec model, which acts as the second input for the network. It is the rightmost part of Figure 1.

3.4 Fusion of the components

To combine the components and complete our hybrid model, the output from each of the three parts is concatenated and subsequently acts as input for a fully connected layer. This layer finally gives as its output the probability/extent that a post, together with its related information, can be considered clickbait.

Table 1: Model Performance Comparison

Model	F1 Score	Accuracy
Proposed Hybrid Approach	**0.65**	**83.53**%
BiLSTM [1]	0.61	83.28%
Feature Engineering Baseline [15]	0.55	83.24%
Concatenated NN Architecture [18]	0.39	74%

3.5 Learning the Parameters

We use binary cross-entropy as the loss optimization function for our model. The cross-entropy method [6] is an iterative procedure where each iteration can be divided into two stages:

(1) Generate a random data sample (vectors, trajectories etc.) according to a specified mechanism.

(2) Update the parameters of the random mechanism based on the data to produce a "better" sample in the next iteration.

4 EVALUATION RESULTS

The model was evaluated over a collection of 19538 social media posts [14], each containing supplementary information like target description, target keywords and linked images. We performed our experiments with the aim of increasing the accuracy and F1 score of the model. Other metrics like mean squared error (MSE) were also considered.

4.1 Training

We randomly partition the training set into training and validation set in a 4:1 ratio. This ensures that the two sets do not overlap. The model hyperparameters are tuned over the validation set. We initialise the fully connected network weights with the uniform distribution in the range $-\sqrt{6/(fanin + fanout)}$ and $\sqrt{6/(fanin + fanout)}$ [8]. We used a batch size of 256 and adadelta [19] as a gradient based optimizer for learning the parameters of the model.

4.2 Comparison with other models

In Table 1, we compare our model with the existing state-of-the-art for the dataset used and other models which have employed similar techniques to accomplish the task. Calculation and comparison across these metrics was conducted on TIRA [15], a platform that offers evaluation as a service. It is clear that our proposed model outperforms the previous feature engineering benchmark and other work done in the field both in terms of F1 score and accuracy of detection.

5 CONCLUSION

In this work, we have come up with a multi-strategy approach to tackle the problem of clickbait detection across the Internet. Our model takes into account both textual and image features, a multimedia approach, to score the classify headlines. A neural attention mechanism is utilised over [1] to improve its performance, simultaneously adding Siamese nets for scoring similarity between different attributes of the post. To build on this approach, we would like to explore better image embedding techniques to better relate it to the article.

REFERENCES

[1] Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. 2017. We used Neural Networks to Detect Clickbaits: You won't believe what happened Next!. In *Advances in Information Retrieval. 39th European Conference on IR Research (ECIR 17) (Lecture Notes in Computer Science)*. Springer.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Prakhar Biyani, Kostas Tsioutsiouliklis, and John Blackmer. 2016. "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 94–100. http://dl.acm.org/citation.cfm?id=3015812.3015827

[4] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop Clickbait: Detecting and preventing clickbaits in online news media. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2016), 9–16.

[5] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. 1990. Advances in Neural Information Processing Systems 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter Handwritten Digit Recognition with a Back-propagation Network, 396–404. http://dl.acm.org/citation.cfm?id=109230.109279

[6] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research* 134, 1 (01 Feb 2005), 19–67. https://doi.org/10.1007/s10479-005-5724-z

[7] Cícero Nogueira Dos Santos and Bianca Zadrozny. 2014. Learning Character-level Representations for Part-of-speech Tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, II–1818–II–1826. http://dl.acm.org/citation.cfm?id=3044805.3045095

[8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks.. In *Aistats*, Vol. 9. 249–256.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1188–1196.

[11] George Loewenstein. 1994. The Psychology of Curiosity: A Review and Reinterpretation. 116 (07 1994), 75–98.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

[13] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. (01 2016).

[14] Martin Potthast, Tim Gollub, Kristof Komlossy, Sebastian Schuster, Matti Wiegmann, Erika Garces, Matthias Hagen, and Benno Stein. 2017. Crowdsourcing a Large Corpus of Clickbait on Twitter. In *(to appear)*.

[15] Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait Detection. In *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16) (Lecture Notes in Computer Science)*, Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello (Eds.), Vol. 9626. Springer, Berlin Heidelberg New York, 810–817. https://doi.org/10.1007/978-3-319-30671-1_72

[16] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[17] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[18] Philippe Thomas. 2017. Clickbait Identification using Neural Networks. *CoRR* abs/1710.08721 (2017). arXiv:1710.08721 http://arxiv.org/abs/1710.08721

[19] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[20] Yiwei Zhou. 2017. Clickbait Detection in Tweets Using Self-attentive Network. *CoRR* abs/1710.05364 (2017). arXiv:1710.05364 http://arxiv.org/abs/1710.05364