

SWDE : A Sub-Word And Document Embedding Based Engine for Clickbait Detection *

Vaibhav Kumar, Mrinal Dhar, Dhruv Khattar
International Institute of Information Technology
Hyderabad
{vaibhav.kumar,mrinal.dhar,dhruv.khattar}@research.iiit.
ac.in

Abhimanshu Mishra
Birla Institute of Technology, Mesra
abhimanshu@gmail.com

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

Manish Shrivastava, Vasudeva Varma
International Institute of Information Technology
Hyderabad
{m.shrivastava,vv}@iiit.ac.in

ABSTRACT

In order to expand their reach and increase website ad revenue, media outlets have started using clickbait techniques to lure readers to click on articles on their digital platform. Having successfully enticed the user to open the article, the article fails to satiate his curiosity serving only to boost click-through rates. Initial methods for this task were dependent on feature engineering, which varies with each dataset. Industry systems have relied on an exhaustive set of rules to get the job done. Neural networks have barely been explored to perform this task. We propose a novel approach considering different textual embeddings of a news headline and the related article. We generate sub-word level embeddings of the title using Convolutional Neural Networks and use them to train a bidirectional LSTM architecture. An attention layer allows for calculation of significance of each term towards the nature of the post. We also generate Doc2Vec embeddings of the title and article text and perform element wise multiplication between the two, following which it is concatenated with the output of the previous component. Finally, this representation is passed through a neural network to obtain a score for the headline. We test our model over 2538 posts (having trained it on 17000 records) and achieve an accuracy of 83.49% outscoring previous state-of-the-art approaches.

KEYWORDS

Clickbait, Neural Networks, Attention Mechanism, Sub-Word Embeddings, Document Embeddings

*The first five authors have equal contribution.

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

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CompS'18, SIGIR, July 8-12, Ann Arbor, MI USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

ACM Reference Format:

Vaibhav Kumar, Mrinal Dhar, Dhruv Khattar, Yash Kumar Lal, Abhimanshu Mishra, and Manish Shrivastava, Vasudeva Varma. 2018. SWDE : A Sub-Word And Document Embedding Based Engine for Clickbait Detection . In *Proceedings of SIGIR 2017 Workshop on Computational Surprise in Information Retrieval (CompS'18, SIGIR)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

In recent years, content delivery has changed drastically, shifting from offline methods to the Internet. It is now the primary source of information for a majority of the populace, especially for ever-changing news updates. This has also caused a shift in users' preferred sources. Previously, these preferences were static, sticking to a particular news source. Now, with the plethora of information available easily, there is no differentiation in the source it has been gathered from, with users opting to go for whatever is convenient.

Keeping up with the times, news agencies have expanded their digital presence, increasing their reach exponentially. They generate revenue by (1) advertisements on their websites, or (2) a subscription based model for articles that might interest users. Since multiple agencies offer similar content, the user has his pick. To lure in more readers and increase the number of clicks on their content, subsequently enhancing their agency's revenue, writers have begun adopting a new technique - clickbait.

Merriam-Webster defines clickbait as something (such as a headline) 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. It is built to create, and consequently capitalise, on the Loewenstein information gap [9] by purposefully misrepresenting or promising what can be expected while reading a story on the web, be it through a headline, image or related text.

We propose a two-pronged approach to detect such headlines. The first component leverages distributional semantics of the title text and models its temporal and sequential properties. The article title is represented as a concatenation of its sub-word level embeddings. The representation of the sub-word serves as input to a bidirectional LSTM network. The contribution of a sub-word towards the clickbait nature of the headline is calculated in a differential manner since the output of the LSTM is passed into an

attention layer [2], following which it goes through a dense layer. The second component focuses on Doc2Vec embeddings of the title and article content, performing an element wise multiplication of the two. This is concatenated with the dense layer output from the previous component. The obtained output is then passed through multiple hidden layers which performs the final classification.

2 RELATED WORK

The importance of detecting clickbait headlines has increased exponentially in recent years. Initial work in this domain can be traced back to [3], relying on heavy feature engineering on a specific news dataset. These works define the various types of clickbait and focus on the presence of linguistic peculiarities in the headline text, including various informality metrics and the use of forward references. Applying such techniques over a social media stream was first attempted by [12] as the authors crowdsourced a dataset of tweets [11] and performed feature engineering to accomplish the task. [5] have tried to expand the work done for news headlines they collected from trusted sources.

[1] used the same collection of headlines as [5] and proposed the first neural network based approach in the field. They employed various recurrent neural network architectures to model sequential data and its dependencies, taking as its inputs a concatenation of the word and character-level embeddings of the headline. Their experiments yielded that bidirectional LSTMs [15] were best suited for the same. [16] built BiLSTMs to model each textual attribute of the post (post-text, target-title, target-paragraphs, target-description, target-keywords, post-time) available in the corpus [11], concatenating their outputs and feeding it to a fully connected layer to classify the post. Attention mechanisms [2] have grown popular for various text classification tasks, like aspect based sentiment analysis. Utilising this technique, [17] deployed a self-attentive bidirectional GRU to infer the importance of each tweet token and model the annotation distribution of headlines in the corpus.

Word vectors and character vectors have been used across various approaches proposed to solve this problem. However, we suggest the use of subword representations to better analyse the morphology of possible clickbait-y words.

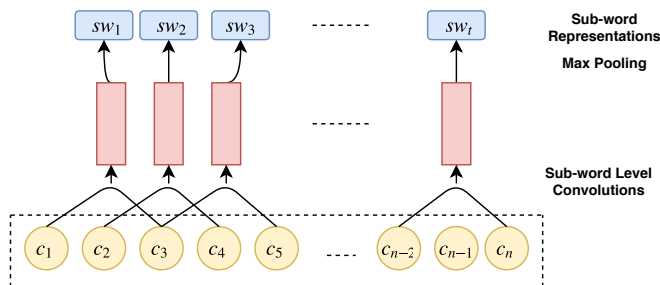


Figure 1: Architecture for learning Sub-word Level Representations using CNN

3 MODEL ARCHITECTURE

We now describe our approach to clickbait detection and the reasons behind devising such a model. Our approach is a fusion of multiple components, each exploiting a particular type of embedding: (1) BiLSTM with attention, and (2) Doc2Vec enrichment. Figure 1 lays out our proposed architecture.

We start with an explanation of the various types of embeddings we have used.

Sub-word Level Representation

Word2Vec [10] has fast become the most popular text embedding method for text since it models a word based on its context. [6] proposed a deep neural network to learn character-level representations of words in order to capture word orthography. Sub-word level embeddings learn representations for character n-grams and represent words as the sum of the n-gram vectors [4]. They work well even with highly noisy text with containing misspellings due to the model learning morpheme-level feature maps. They have proven to be extremely useful in tasks such as sentiment analysis [13]. These intermediate sub-word feature representations are learned by the filters during the convolution operation. We generate such an embedding by passing the characters of a sentence individually into 3 layer 1D convolutional neural network. Each filter then acts as a learned sub-word level feature. A representation for this architecture can be found in Figure 1.

Document Embeddings

Doc2Vec [8] 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 [10] 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 [14] to learn 300 dimensional Doc2Vec embeddings for each target description and post title available.

We now describe the components in our proposed approach, both individually and together, and explain 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 [7]. This means it will not be able to efficiently model dependencies and interactions between words or characters that are a few steps apart. LSTMs are able to tackle this issue by their use of gating mechanisms. We convert each article headline into its corresponding sub-word level representation to act as input to our bidirectional LSTMs.

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

$$[\vec{f}_t, \vec{i}_t, \vec{o}_t] = \sigma[\vec{W}[\vec{h}_{t-1}, \vec{r}_t] + \vec{b}] \quad (1)$$

$$\vec{l}_t = \tanh[\vec{V}[\vec{h}_{t-1}, \vec{r}_t] + \vec{d}] \quad (2)$$

$$\vec{c}_t = \vec{f}_t \cdot \vec{c}_{t-1} + \vec{i}_t \cdot \vec{l}_t \quad (3)$$

$$\vec{h}_t = \vec{o}_t \cdot \tanh(\vec{c}_t) \quad (4)$$

here σ is the logistic sigmoid function, $\vec{f}_t, \vec{i}_t, \vec{o}_t$ represent the forget, input and output gates respectively. \vec{r}_t denotes the input at time t and \vec{h}_t denotes the latent state, \vec{b}_t and \vec{d}_t represent the bias terms. The forget, input and output gates control the flow of information throughout the sequence. \vec{W} and \vec{V} are matrices which represent the weights associated with the connections.

$(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_R)$ 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, \dots, h_K) , where

$$h_i = \begin{bmatrix} \vec{h}_i \\ \overleftarrow{h}_i \end{bmatrix} \quad (5)$$

Finally, we are left with the task of figuring out the significance of each word in the sequence i.e. how much a particular sub-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.

$$c_{attention} = \sum_{j=1}^K \alpha_j h_j \quad (6)$$

where, h_1, \dots, 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 is represented as the left most component in Figure 1.

3.2 Doc2Vec Enrichment

Each record in the dataset has a target description attached with it. This is the entire text of the article whose title has been given. We generate document embeddings for both the title and the article text and perform element wise multiplication over the two. This is used to augment the output obtained from the first component.

3.3 Fusion of Components

The outputs from the aforementioned components are now concatenated and passed through two dense layers and finally goes into a fully connected layer. This layer finally gives out the probability that a post can be marked clickbait.

3.4 Learning the Parameters

We use binary cross-entropy as the loss optimization function for our model. The cross-entropy method [?] 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.

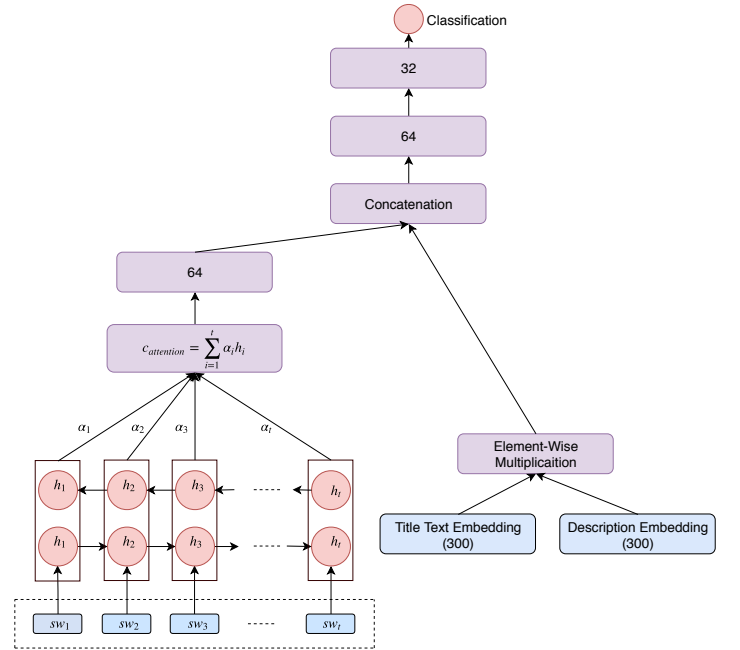


Figure 2: Full Model Architecture

4 EVALUATION AND RESULTS

The model was evaluated over a collection of 2538 social media posts taken from [11], 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 $\sqrt{6/(fanin + fanout)}$

Table 1: Model Performance Comparison

Model	F1 Score	Accuracy
Proposed Approach	0.63	83.49%
BiLSTM [1]	0.61	83.28%
Feature Engineering SotA [12]	0.55	83.24%
Concatenated NN Architecture [16]	0.39	74%

[?]. We used a batch size of 256 and adadelata [?] as a gradient based optimizer for learning the parameters of the model.

4.2 Model Comparison

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. 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 AND FUTURE WORK

We have devised an approach to detecting clickbait that puts emphasis on utilising the linguistic value of words by learning its morphological features through its sub-word representations. These embeddings and their dependencies are, in turn, modeled by the LSTM. Attention mechanism allows us to understand the importance of individual representations towards the nature of the post. Using the document embeddings for title and article text allows us to augment the generated embeddings and use as input to a neural network to finally classify the post. In the future, we would like to explore the possibility of integrating the sub-word representations with deep neural networks to better model the temporal and sequential properties of text.

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] Dzmity 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 Tsioutsoulis, 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] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR abs/1607.04606* (2016). [arXiv:1607.04606](http://arxiv.org/abs/1607.04606) <http://arxiv.org/abs/1607.04606>
- [5] 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.
- [6] Cicero 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>
- [7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [8] 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.
- [9] George Loewenstein. 1994. The Psychology of Curiosity: A Review and Reinterpretation. 116 (07 1994), 75–98.
- [10] 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>
- [11] Martin Potthast, Tim Gollub, Kristof Komlosy, Sebastian Schuster, Matti Wiegmann, Erika Garces, Matthias Hagen, and Benno Stein. 2017. Crowdsourcing a Large Corpus of Clickbait on Twitter. In *(to appear)*.
- [12] Martin Potthast, Sebastian Köppl, 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
- [13] Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text. *CoRR abs/1611.00472* (2016). [arXiv:1611.00472](http://arxiv.org/abs/1611.00472) <http://arxiv.org/abs/1611.00472>
- [14] 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>.
- [15] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.
- [16] Philippe Thomas. 2017. Clickbait Identification using Neural Networks. *CoRR abs/1710.08721* (2017). [arXiv:1710.08721](http://arxiv.org/abs/1710.08721) <http://arxiv.org/abs/1710.08721>
- [17] 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) <http://arxiv.org/abs/1710.05364>