arXiv:1710.01507v1 [cs.IR] 4 Oct 2017

A Neural Clickbait Detection Engine

The Tuna Clickbait Detector at the Clickbait Challenge 2017

Siddhartha Gairola
International Institute of Information Technology, Hyderabad

Yash Kumar Lal
Manipal Institute of Technology, Manipal

Vaibhav Kumar
International Institute of Information Technology, Hyderabad

Dhruv Khattar
International Institute of Information Technology, Hyderabad

ABSTRACT

In an age where people are becoming increasing likely to trust information found through online media, journalists have begun employing techniques to lure readers to articles by using catchy headlines, called clickbait. These headlines entice the user into clicking through the article whilst not providing information relevant to the headline itself. Previous methods of detecting clickbait have explored techniques heavily dependent on feature engineering, with little experimentation having been tried with neural network architectures. We introduce a novel model combining recurrent neural networks, attention layers and image embeddings. Our model uses a combination of distributed word embeddings derived from unannotated corpora, character level embeddings calculated through Convolutional Neural Networks. These representations are passed through a bidirectional LSTM with an attention layer. The image embeddings are also learnt from large data using CNNs. Experimental results show that our model achieves an F1 score of 65.37% beating the previous benchmark of 55.21%.

1. INTRODUCTION

With the bulk of information propagation and consumption moving to online platforms, the media world is suffering through a drastic shift. In case of traditional, offline media sources, where a user preference would be static i.e. loyalty to a particular news source would be unwavering. However, now the Internet offers the readers a plethora of choices ranging from local to international, mainstream to niche, popular to alternative, editorials to blogs. This has forced traditional news outlets to change tack in order to stay in business. An added bonus is that online sources generally do not have a subscription charge, choosing to generate revenue through advertisements on the page. To stay relevant in the midst of such competition whilst staying afloat, online journalism has adopted a new technique to attract users - clickbait.

Merriam Webster defines clickbait as something (such as a headline) designed to make readers want to click on a hyperlink especially when the link leads to content of dubious value or interest. Clickbaiting is the intentional act of over-promising or otherwise misrepresenting — in a headline, on social media, in an image, or some combination — what you're going to find when you read a story on the web. At its worst, this strategy indicates that the publisher is at a loss for how to sell audiences on a story (perhaps due to having sub-par content to begin with). Sometimes these headlines can be found on social media or on traffic-generating widgets, and clicking on them will lead to a page with an unoriginal story that just repeats (or even distorts) facts from another site that reported the story first.

In a world characterized by reducing attention spans and a marked increase in reliance on social media for information, being a news site isn't easy, business-wise. Anyone who's trying to distribute information online is at the mercy of the big companies like Facebook and Google that help people discover their content.Constantly evolving algorithms that minimize the effect of SEO techniques have put publishers in trouble when it comes to their bottomline - page views. Instead of deterring clickbaiting, this has led to the rise of new strategies, still heavily relying on traditional hooks like dramatization, luring and sensationalization.

We build on the idea of using neural networks for the task of clickbait detection. Our work involves the use of the text and target description of clickbait-y headlines. More often than not, posts are accompanied by an image to capture user attention easily. Our model leverages the information gained from the linked image and attempt to gauge its relationship to the post itself. We employ the concept of neural attention layer to add a layer of complexity to the previous work that has already been done.

2. RELATED WORK

Clickbait Detection is a relatively new and unexplored domain. The start of work in this field can be traced back to [2] and [**?**]. Initial methods relied on exploiting a set of rich, hand-crafted features based on language specific peculiarities. Such features were liable to change with the dataset and cannot be deployed for generic use cases, or modified to factor in multilingualism.

In 2014, Facebook, in an attempt to enhance user experience, announced the deployment of a software to remove clickbait stories from users' News Feeds [**?**], depending on the click-to-share ratio and the time spent on these stories. However, as is evident to this day, users scroll through mundane clickbait content on a daily basis. Several ad-hoc approaches have surfaced which utilise an extensive lexicon of possible clickbait sources to identify similar headlines. [2] worked to detect clickbait posts in the Twitter data stream. [**?**] proposed the formalization of the types of clickbait and the use

of various novel informality measures, and forward reference, for identifying clickbait. [**?**] picked out a collection of headlines from WikiNews and another from clickbait websites, attempting to detect clickbait in news. The authors still relied on handcrafted features extracted by carrying out detailed linguistic analysis on both sets of headlines.

The first significant attempt at applying deep learning techniques to this task can be seen in [**?**]. Expanding on the work done in the news headline dataset of [**?**], the authors have proposed the use of word and character level embeddings as features for their model. They explore various Recurrent Neural Network architectures to capture contextual information and preserve long-term dependencies in posts. Post evaluation, it is seen that a bidirectional LSTM is the best option for the task.

3. APPROACH

This section describes our approach. Our model consists of three components (i.) Bi-directional LSTM with Attention, (ii.) Siamese Neural Network on Text embeddings and (iii.) Siamese Neural Network on Visual embeddings. Two types of features are used in this experiment. Figure 1 shows the model architecture that has been used by us.

Word Embeddings:.

In our work, we use the pre-trained 300 dimensional word2vec embeddings [**?**] which were trained on about 100B words from the Google News dataset using the Continuous Bag of Words architecture. For each word in the title of the social media post we get it's equivalent word2vec embeddings which are described above.

Document Embeddings:.

Doc2vec [**?**] is an unsupervised algorithm to generate vectors for sentence/paragraphs/documents. The algorithm is an adaptation of word2vec which can generate vectors for words. The vectors generated by doc2vec can be used for tasks like finding similarity between sentences/paragraphs/documents. Unlike sequence models like RNN, where word sequence is captured in generated sentence vectors, doc2vec sentence vectors are word order independent. For each target description and the social media post title we use gensim [**?**] to learn doc2vec [**?**] embeddings for those. The size of the embeddings is set to 300.

Pre-trained CNN features:.

Image descriptors computed using convolutional neural networks (CNNs) pre-trained on large data such as ImageNet have proven to be very effective for a number of visual understanding tasks. The success of these features can be attributed to implicit learning of spatial layout and object semantics at later layers of the network from very large datasets. We used pre-trained network (i) VGG19 architecture [**?**] trained on ILSVRC-2012 (ImageNet) dataset and extract CNN features. For this architectures, we use the output of FC7 (fully-connected) layer (4096 dim.) as feature representation. We did not fine-tune these networks for our task due to lack of labeled data. Each feature is a 4096 dimensional vector.

3.1 Bi-directional LSTM with Attention

Standard RNNs have difficulty preserving long range dependencies due to the vanishing gradient problem[11]. In the given problem, this corresponds to interaction between words that are several steps apart from each other. The LSTM is able to tackle this problem through the use of gating mechanism. For each social media post, the content of the post itself as well as the main content of the linked target web page is available. We then use the words of the title for each post as inputs to our bidirectional LSTMs. The forward state updates of the LSTM satisfy the following equations

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

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

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

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

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. The backward states ($\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_R$) are computed in a similar manner as above. The number of bidirectional LSTM units is set to a constant K which is the max length of the lengths of the title for the posts used in training. We then concatenate the forward and backward states to obtain the annotations $(h_1, h_2, \ldots, h_K)$, where

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

We then need to identify the extent/granularity of how much each word contributes to the post being a clickbait. Recently in [**?**], the effectiveness of attention mechanisms has been shown for the task of neural machine translation. The goal of the attention mechanism in such tasks is to derive a context vector that captures relevant source side information to help predict the current target word. In our case, we want to use the sequence of annotations generated by the encoder to come up with a context vector that captures how each word contributes to the post linking to a clickbait. Though, in a typical RNN encoder-decoder framework [**?**], a context vector is generated at each time step to predict the target word, in our case, we only need to calculate the context vector for a single time step.

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

where, h_1,...,h_K represents the sequence of annotations to which the encoder maps the sequence of read news articles and each α_j represents the respective weight corresponding to each annotation h_j. The user view (left view) of the model can be seen in Figure 1. The input to this is a selected amount of reading history of each user. Each r_i in the figure is a word2vec embedding of dimension 300.

3.2 Siamese Neural Network on Text Embeddings

The second component consists of a siamese network [**?**] for the title component and the target description component of the social media post. We use a siamese network to find the similarity between the title of the given social media post and the target description. It takes as input doc2vec embeddings of the target description and concatenated doc2vec embeddings of the title.

3.3 Siamese Neural Network on Visual Embeddings

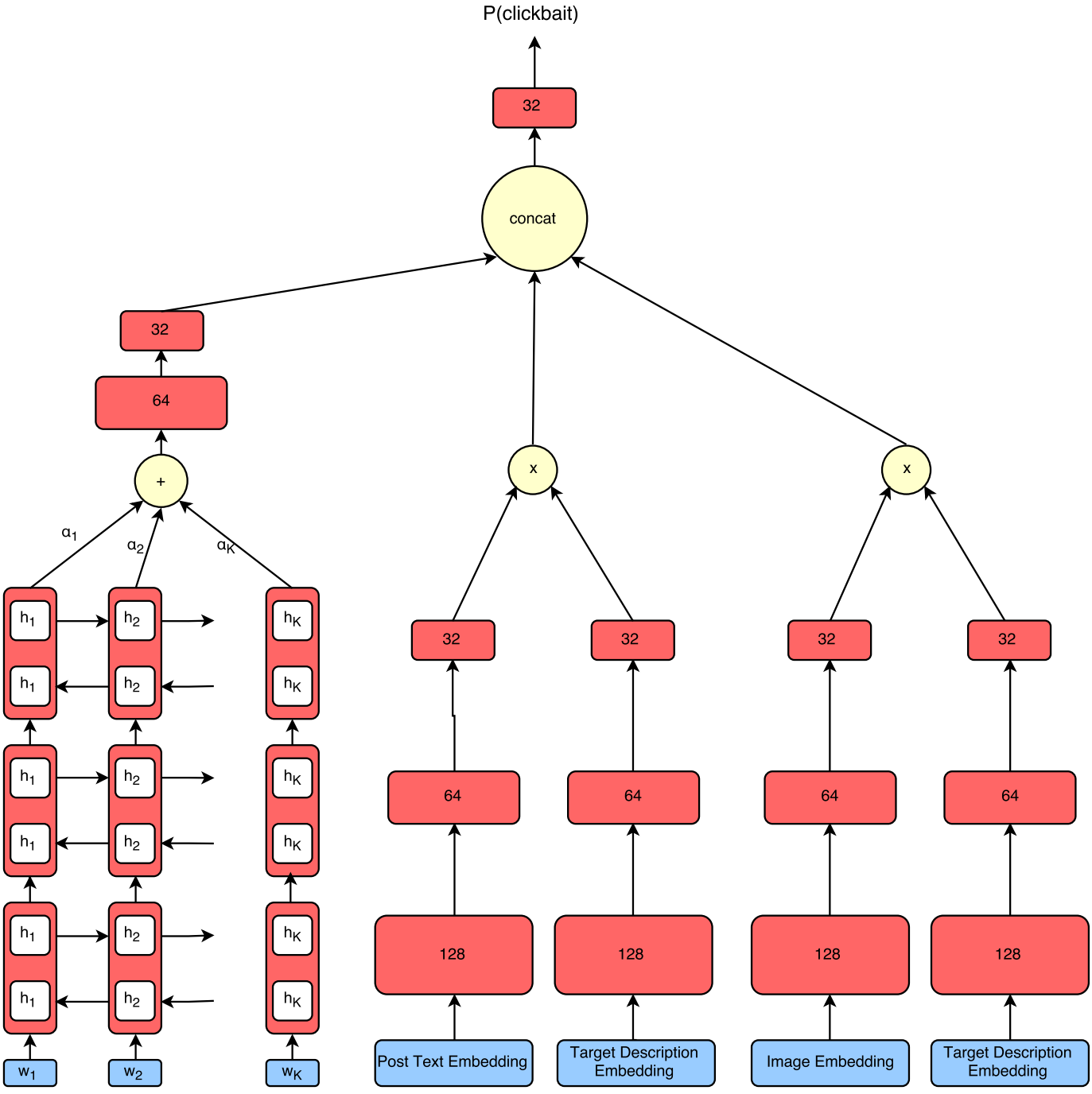

Figure 1: Model Architecture

The third component consists of a siamese network [**?**] for the image component of the social media post. To capture the relevance of the image of the social media post we need to find the similarity between the image and the target description. The input to this network are the doc2vec embeddings (300 dimensional) of the target description and the visual embeddings of the image. But before this the 4096 visual embeddings are given as input to a dense layer which gives a 300 dimensional vector as output. This is then given as input to the siamese network along with the doc2vec embeddings of the target description.

3.4 Combining the output from the three components

Finally, the output from the three components are concatenated and sent as input to a fully-connected layer. It uses a binary cross-entropy loss to optimize its model parameters. This gives as output the probability that the social media post is a clickbait.

3.5 Learning the parameters

Binary cross entropy is used as the loss optimization function to learn the parameters of the model. In information theory, the cross entropy between two probability distributions *p* and *q* over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set, if a coding scheme is used that is optimized for an "unnatural" probability distribution *q*, rather than the "true" distribution *p*.

The cross entropy method [**?**] is an iterative procedure where each iteration can be divided into two phases:
(i) Generate a random data sample (vectors, trajectories etc.) according to a specified mechanism
(ii) Update the parameters of the random mechanism based on the data to produce "better" sample in the next iteration

4. EVALUATION RESULTS

The model was evaluated over a collection of 19538 posts [4] with attached a target article, keywords, description and linked images, if any. We performed our experiments with a primary target of achieving the lowest mean square error (MSE) metric. Additional metrics such as precision, recall, F1 score and accuracy were also kept in mind.

4.1 Training:

We randomly divide the training set into training and validation set in a 4:1 ratio. This helps us to ensure that the two sets do not overlap. We tuned the hyper-parameters of our model using the validation set. All the model and its variants are learnt by optimizing the log loss of Equation 8. We initialise the fully connected network weights with the uniform distribution in the range between $-\sqrt{6/(fanin + fanout)}$ and $\sqrt{6/(fanin + fanout)}$ [**?**] . We used a batch size of 256 and used adadelta [**?**] as a gradient based optimizer for learning the parameters of the model.

4.2 Comparison with existing models:

We compare our model with the state-of-the-art for this dataset as presented in [2].

Model	MSE	F1 Score
Bi-LSTM with Attention [%]	4.56	65.37
Feature Engineering [%] [2]	4.35	55.21

Table 1: Comparison of our model with existing methods

TIRA [1], a platform that offers evaluation as a service, was used to calculate and compare models across different metrics. From Table 1, it is seen that our model significantly outperforms the previously set benchmark in terms of the F1 score, while the difference in mean squared error is almost negligible.

5. CONCLUSION

In this paper, we have introduced a new neural network architecture for clickbait detection, one that exploits text as well as images. A neural attention layer has been applied over a previously proposed clickbait detection model [**?**] to improve on its performance. Siamese architecture has been utilised to find and exploit similarity scores between various pieces of information available in the dataset. In the future, we would like to tweak the image embedding component of the model in order for it to relate images with articles better.

References

[1] M. Potthast, T. Gollub, F. Rangel, P. Rosso, E. Stamatatos, and B. Stein. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *CLEF*, pages 268–299. Springer, 2014.

[2] M. Potthast, S. Köpsel, B. Stein, and M. Hagen. Clickbait Detection. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. Di Nunzio, C. Hauff, and G. Silvello, editors, *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)*, volume 9626 of *Lecture Notes in Computer Science*, pages 810–817, Berlin Heidelberg New York, Mar. 2016. Springer. .

[3] M. Potthast, T. Gollub, M. Hagen, and B. Stein. The Clickbait Challenge 2017: Towards a Regression Model for Clickbait Strength. In *Proceddings of the Clickbait Chhallenge*, 2017.

[4] M. Potthast, T. Gollub, K. Komlossy, S. Schuster, M. Wiegmann, E. Garces, M. Hagen, and B. Stein. Crowdsourcing a Large Corpus of Clickbait on Twitter. In *(to appear)*, 2017.