Similarity-Aware Deep Attentive Model for Clickbait Detection

Manqing Dong[1(✉)], Lina Yao[1], Xianzhi Wang[2], Boualem Benatallah[1], and Chaoran Huang[1]

[1] Department of Computer Science, University of New South Wales, Sydney, Australia
manqing.dong@unsw.edu.au
[2] School of Software, University of Technology Sydney, Sydney, Australia

Abstract. Clickbait is a type of web content advertisements designed to entice readers into clicking accompanying links. Usually, such links will lead to articles that are either misleading or non-informative, making the detection of clickbait essential for our daily lives. Automated clickbait detection is a relatively new research topic. Most recent work handles the clickbait detection problem with deep learning approaches to extract features from the meta-data of content. However, little attention has been paid to the relationship between the misleading titles and the target content, which we found to be an important clue for enhancing clickbait detection. In this work, we propose a deep similarity-aware attentive model to capture and represent such similarities with better expressiveness. In particular, we present the ways of either using similarity only or integrating it with other available quality features for the clickbait detection. We evaluate our model on two benchmark datasets, and the experimental results demonstrate the effectiveness of our approach by outperforming a series of competitive state-of-the-arts and baseline methods.

1 Introduction

Clickbait is a type of web links designed to entice users to enter specific web-pages or videos[1]. Clickbait titles are generally written in an exaggerated or ambiguous way to attract curious readers to the hyper-linked content. For example, "*You will never believe what happened when...*" and "*This is the biggest mistake you can make...*" are two representative titles of clickbait[2]. Most clickbaits are created for financial purposes. For example, Web publishers regard clickbait as a useful tool to draw attention to their websites and make money from advertisements. However, clickbaits are often malicious to the readers despite the potential benefit to the advertiser, as they are mostly misleading or meaningless

---

[1] https://en.wikipedia.org/wiki/Clickbait.
[2] https://www.thedailybeast.com/saving-us-from-ourselves-the-anti-clickbait-movement.

articles. For most of the time, the content of such articles is not even related to the title, making the detection of clickbait not only necessary but also highly significant.

Research on clickbait detection has been active in recent years. Potthast [12] made one of the first few early attempts. They consider the features from both titles and the linked web page, including linguistic information (e.g., the mean word length and sentiment polarity) and side information (e.g., the writer of the titles). They feed the features into traditional classifiers such as logistic regression, Naive Bayes, and random forest [10] and attain the accuracy of around 80%. Later, deep learning methods [5,20] are increasingly studied owing to their advantages in dealing with high-dimensional data and extracting non-linear relationship among features [8]. Most of the top teams in 2017's Clickbait Challenge[3] use deep learning based methods. Zhou's classifier [20], which won the first place, is a is a Recurrent Neural Network [8] based framework that considers the context of words, more specifically, a hybrid of bidirectional Gated Recurrent Unit (GRU) and attention model [18]. Another work worth mentioning is conducted by Maria et al. [5], which takes both image and the text representations into consideration and different deep learning methods, like Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) [8], are tried for the prediction.

Until now, few works have investigated the similarities between the misleading titles and the linked web contents for clickbait detection. Clickbaits are not necessarily spams, instead, they may actually contain genuine information but with rather low quality (e.g., unmatched contents and titles). This makes it possible to improve the performance of clickbait detection. In a recent work, Biyani et al. [2] utilize similarities between the title and the top five sentences in the bodies as features blended with traditional texture information for detecting clickbaits. Another work by Kumal et al. [7] used Siamese Networks for measuring the text and visual similarities and combined the similarities as the input of several fully connected layers. Yet existing studies have several limitations: (i) they consider the similarity as features in a linear manner and therefore lack the expressiveness when compared to non-linear methods; (ii) current efforts on leveraging such similarities typically use the partial/local information, such as quantifying the similarity between the titles and the top five sentences of content; on the other hand, they overlooking the hidden global information in the entire content. To overcome these challenges, we propose a deep attentive similarity model for capturing the discriminative information from local and global similarities. This way, we provide a way of untangling the non-linear connections between content and titles for the further prediction. The global similarity in this work measures the similarities over the pair of inputs. Specially, to alleviate the impact of noise, we propose using attentive local similarities to select the most useful similarity information for the final prediction. In a nutshell, we make the following contributions:

– We propose a deep attentive similarity model which is capable of capturing both global and local similarities of the pair of inputs. The model represents local similarities as vectors to combine them with other features for the future prediction easily.
– We introduce the ways of either using only similarity information or combining the similarity with other features to detect clickbait. We further employ an attention-based bidirectional Gated Recurrent Units (GRU) model to obtain robust representations of textual inputs.
– We evaluate our framework on two benchmark datasets of clickbait detection. The experimental results demonstrate its effectiveness in detecting clickbait and its competitive performance against the baselines.

The rest of this paper is organized as follows. Section 2 describes the related work of deep semantic similarity model. Section 3 defines the relevant operators, the target problem, and the framework of our model. Section 4 describes the real-world dataset and corresponding experimental results. Section 5 gives the concluding remarks.

2   Related Work

2.1   Clickbait Detection

As an arguably new research topic, first attempts on this problem extract latent features [12,17]. For example, Chen et al. [3] considered both content cues and non-text cues. Specifically, they extract features from lexical and semantic levels for content cues and features like user behavior, information about figures for non-text cues. Those features are then used over various classifiers (e.g. Naive Bayes classifier, SVM classifier) for the prediction tasks.

Instead of extracting features manually, some recent works utilize word vectors [13] for representing the textual information in order to take advantages of deep learning methods [4]. For example, Zheng et al. [19] transformed the titles into word embeddings and then used text-Convolutional Neural Networks as classifier. Also, Recurrent Neural Network (RNN) based methods are widely used in detecting the clickbaits, due to the efficiency in dealing with sequential data. In fact, RNN was used by all the top five teams in the aforementioned Clickbait Challenge. On the basis of RNN, Glenski et al. [5] used LSTM, and Zhou [20] used attentive bi-GRU for learning the textual inputs.

However, limited works exploited the similarity information for detecting the clickbait, although this can directly indicate the matching level between titles and contents. In early works Biyani et al. [2] made the few attempts that used the similarities information with several features, including n-grams and metrics for evaluating the informality. They then fed those features into a gradient boost decision tree (GBDT) classifier. Kumal et al. [7] used Siamese Networks for measuring the text similarity between titles and bodies, and image similarity between figures and descriptions. They then concatenated the similarities for the final prediction.

2.2   Deep Semantic Similarity Model

The Deep Semantic Similarity Model (DSSM) [6] was originally designed for web search ranking, which is a latent semantic model with a deep structure that projects queries Q and documents D into a common low-dimensional space to calculate the semantic similarity. DSSM differs from the traditional latent semantic models in the use of deep neural networks that learn the latent representations. In particular, DSSM first maps the input features x to the latent semantic space l by:

$$layer_1 = W_1 x \tag{1}$$
$$layer_i = f(W_i layer_{i-1} + b_i), i = 2, \ldots, N-1 \tag{2}$$
$$l = f(W_N layer_{N-1} + b_N) \tag{3}$$

where $layer_i$ is the ith intermediate hidden layer, W_i is the ith weight matrix, b_i is the ith bias matrix, and f is the activation function, e.g., sigmoid function. Then, the semantic relevance score between a query Q and a document D is measured:

$$R(Q, D) = cosine(l_Q, l_D) = \frac{l_Q^T l_D}{\| l_Q \| \| l_D \|} \tag{4}$$

Learning the DSSM is equivalent to maximizing the fraction of the similarity between queries Q and matching documents D_+ in the entire collection of either matching and mismatching documents. A typical improvement of DSSM is to change the way of learning the latent representations, e.g., by changing the deep neural networks with convolutional neural networks (CDSSM) [14] or with long short term memory (LSTM-DSSM) [11]. In this work, we follow the idea of DSSM to calculate the similarities in the latent space, but the latent space is produced in different ways for different types of inputs. Since the similarity in DSSM is a constant, we regard this similarity value as the global similarity and learn a local similarities vector from it for the further prediction. The next section gives the details.

3   Methodology

In this paper, we define a piece of information as clickbait when *the title does not match the content*. Given a set of titles $H = \{h_1, h_2, \ldots, h_N\}$, and their bodies $B = \{b_1, b_2, \ldots, b_N\}$, the goal is to predict a label $Y = \{y_1, y_2, \ldots, y_N\}$ of these pairs, where $y_i = 1$ if headline i is a clickbait. Our framework includes three parts: learning latent representations, learning the similarities, and using the similarity for the further predictions. Figure 1 illustrates the last two parts.

3.1   Learn Latent Representations

Here we consider transform the titles H and bodies B into the latent representations: L_H and L_B, where $L_H, L_B \in \mathbb{R}^M$. We first preprocess the text information,
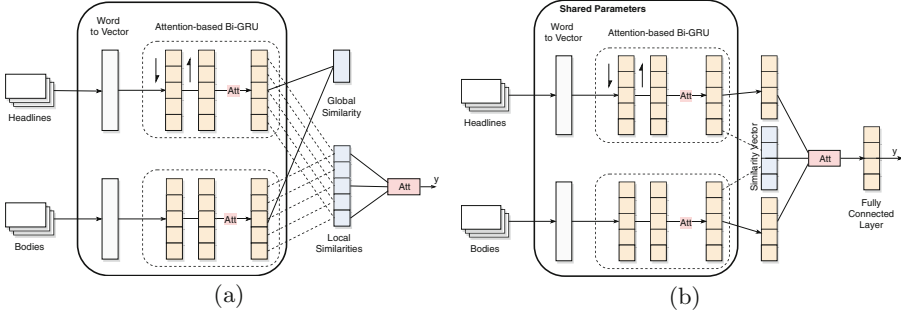
Fig. 1. The illustration of our proposed model. The left (a) shows the learning of the global similarity and local similarities. The words will first be transformed to vectors and go through the attention based bidirectional GRU models. The global similarity is then learned from the label which minimize the distance between matching titles and bodies. And the local similarity is calculated accordingly. And the right one (b) shows the combined method for doing the further prediction.

where we remove all the punctuation and stop words, make the sentence in a lower form, and do word lemmatization [15]. And we then transform the cleaned inputs as word vectors [13].

We apply the attention-based bidirectional GRU [18], one of the most popular RNN based models, to obtain hidden representations, which have shown effectiveness in dealing with natural languages tasks in recent years, by using a gating mechanism to track the state of sequences without using separate memory cells [1]. Given a $b_i, i \in [1, N]$, we first get a set of word embedding vectors $w_{i,t}$, where $t \in [1, T_i]$, T_i is the number of words in body i. Then, we use the bidirectional GRU to get annotations of words by summarizing information from both directions of a word. The bidirectional GRU contains a forward $\overrightarrow{GRU}$, which reads the sentence from w_{i1} to w_{iT_i}, and a backward $\overleftarrow{GRU}$, which reads the sentence from w_{iT_i} to w_{i1}:

$$\overrightarrow{w'_{it}} = \overrightarrow{GRU}(w_{it}), t \in [1, T_i] \tag{5}$$

$$\overleftarrow{w'_{it}} = \overleftarrow{GRU}(w_{it}), t \in [T_i, 1] \tag{6}$$

Then, we get the hidden representation w'_{it} by concatenating the forward hidden state $\overrightarrow{w'_{it}}$ and backward hidden state $\overleftarrow{w'_{it}}$: $w'_{it} = [\overrightarrow{w'_{it}}, \overleftarrow{w'_{it}}]$. And an attention mechanism is used to extract words that are important to the sentence and aggregate the representation of those words to get the latent representation L_{b_i}:

$$u_t = tanh(W_w w'_{it} + b_w) \tag{7}$$

$$a_t = \frac{exp(u_t^T u_w)}{\Sigma_t exp(u_t^T u_w)} \tag{8}$$

$$L_{b_i} = \Sigma_t a_t w'_{it} \tag{9}$$

Finally, we get the latent representation of the bodies L_B. In a similar way, we can obtain the hidden representation L_H.

3.2   Learn the Similarities

For getting the global similarity, similar to DSSM, we calculate it as the cosine similarity between the L_H and L_B:

$$r(H, B) = cosine(L_H, L_B) = \frac{L_H^T L_B}{\| L_H \| \| L_B \|} \tag{10}$$

This similarity $r(H, B)$ is a constant within $[0, 1]$ (if the input space is in a positive space), a higher value of which stands for a higher level of consistency between the titles and bodies. Intuitively, we want to maximize this similarity score between the matching titles and minimize the similarity score between the mismatching pairs. For using only global similarity to predict the clickbait, we use $R(H, B) = softmax[r(H, B), (1-r(H, B))]$ as the balance value of the global similarities. Thus, the prediction for the matching is $\hat{y} = \text{argmax}_y(P(y|h, b))$, where $P(Y|H, B) = R(H, B)$. We use cross entropy for measuring the loss, which is:

$$\mathcal{L} = -\Sigma_{Y=0,1} Y \log P(Y|H, B) \tag{11}$$

Then the optimization goal is to minimize this loss:

$$\underset{\Theta}{\text{argmin}} \ \mathcal{L} + \lambda \| \Theta \|_2 \tag{12}$$

where the $\mathcal{L}$ is the loss function of Eq. 11 and the right norm is for the regularization of the parameters, λ is the hyperparameter. And we take Adam as our optimization method [8]. This way, maximizing the global similarity between matching titles and bodies also helps us update the matching latent representations accordingly. That means the corresponding latent representations for titles and bodies will be as close as possible. Plus the global matching similarity is usually sensitive to some noise like partial occlusion [16]. We learn the local similarities for a better matching representation. Recall that we have latent representations $L_H, L_B \in \mathbb{R}^M$, we set the local block size as $\mu, \mu < M$, and we move from left to right with $\nu, \nu < (M - \mu)$ strides to the next local block. Then we have $K = [\frac{M-\mu}{\nu}]$ local blocks, that is, the latent L_H can be represented as $L_H = [L_{H,1}^T, L_{H,2}^T, \ldots, L_{H,K}^T]^T$ and so as the L_B. Thus, the local similarities are then calculated by

$$LS(H, B) = (r(L_{H,1}, L_{B,1}), ..., r(L_{H,K}, L_{B,K}))^T \tag{13}$$

We use an attentive mechanism to select the most useful similarities, i.e., the local similarities $LS(H, B)$, for the final prediction. More specifically, we apply the self-attention mechanism [9] for getting the attention values (which serve as self-learned weight values),

$$A = softmax(V_a \tanh(W_a LS(H, B)^T)) \tag{14}$$

where $W_a \in \mathbb{R}^K$ and $V_a \in \mathbb{R}^{K \times K}$ are two weight matrices, and A is the attention matrix for the local similarity. Then let

$$P = softmax(W_P(A \times LS(H, B)) + b_P) \qquad (15)$$

we get the prediction for the clickbait as $\hat{y} = argmax_y P$, where W_P and b_P are weights and biases. Similarity, for using only local similarities to predict the clickbaits, we choose the combination of cross entropy and regularization as the loss function and optimize it using Adam optimization method.

3.3   Learn for Prediction

So far, we introduced how global similarities and local similarities are learned and how to utilize only attentive local similarities to the detection. Here, we will introduce the classification method which combines the features with the similarities.

Learning from raw textual information could help mine clickbait indicators such as the writing style and quality, and learning the similarity can lead to the matching degrees. To combine these two useful clues, we adopt an attentive way for the final prediction, which is shown in Fig. 1(b). We first use fully connected layers to map the hidden representations L_H and L_B into layers with K dimension.

$$L'_H = f(W_H L_H + b_H) \qquad (16)$$
$$L'_B = f(W_B L_B + b_B) \qquad (17)$$

Denote $LS(H, B)$ as L'_{LS}, then we get a concatenation layer $L' = [L'_H, L'_{LS}, L'_B]$. Similar to Eqs. 14 to 15, we calculate the self attention values $A_{L'}$ and use them get the combination layer L''. The combination layer is then fed into multilayer perceptrons and we get the $P = softmax(W_P L'' + b_P)$. Then the prediction is $\hat{y} = argmax_y P$. Similarly, we set the loss as the combination of cross entropy and L2-norm of the parameters, and we learn the parameters with Adam optimization.

4   Experiments

In this section, we will test our model on two benchmark datasets. We will first give some details about these two datasets, and then present the comparison results of our method and several related works. Furthermore, we conduct the sensitivity analysis of the proposed method with different parameter settings.

4.1   Dataset Description

Here we use two datasets for evaluating the model.

- **Clickbait Challenge**[4] is a benchmark dataset for the clickbait detection that released in 2017. The dataset contains over 20,000 labelled pairs of posts for training and validation. There are five judges, each giving a clickbait score (from 0 to 1) to label the post. And a higher score stands for the higher probability of a post being clickbait. Then we regard the post with the mean score over 0.5 as clickbait.
- **FNC dataset**[5] is from the Fake News Challenge in 2017. The data describe pairs of titles and bodies and are labeled as 'agree', 'disagree', 'discuss' and 'unrelated'. We regard data with label 'unrelated' as clickbait. The dataset contains 49,972 pairs of titles and bodies for training and 25,413 pairs for the testing.

As mentioned in the latent representation learning part, we first preprocess the texts by removing the stop words and lemmatization. The processed Clickbait Challenge dataset has an average of 10 words in the titles and 50 words in the bodies, while the FNC dataset has an average of 8 and 200 words accordingly. We further vectorize the data using word-embedding techniques [13], which can be conducted unsupervised. Given that some titles only contain one word and this word is unique among the corpus, we train the word vectors with the "Min Word Count" set to 1.

4.2   Comparison Methods

We have compared our LSDA model with a series of baseline models and state-of-the-arts. Where the first two are latent semantic similarity based models and the other four are the most current works for clickbait detection.

- Huang et al. [6]: propose a deep semantic similarity model (DSSM) that uses deep neural networks, to get the latent representations of the inputs and calculate the similarity in the latent representation space. They then use the calculated similarities as introduced in learning global similarity for the prediction. The difference is they use N-gram to preprocess the textual features.[6]
- Shen et al. [14]: propose a similar structure to DSSM, yet they use convolutional neural networks for latent representations.
- Kumar et al. [7]: propose a hybrid method for detecting the clickbait. They first use attentive bidirectional RNN based methods for learning the inputs, and then concatenate the latent inputs with the relationship information that learned with Siamese Net for the final prediction.

---

[4] https://www.clickbait-challenge.org/.
[5] http://www.fakenewschallenge.org/.
[6] https://en.wikipedia.org/wiki/N-gram.

– Zheng et al. [19]: only consider using characteristics from titles to detect the clickbait. They first transform the titles into word vectors and then use text-CNN for predicting the labels.
– Glenski et al. [5]: consider information from both titles and bodies. They also learn from the textual information by firstly vectorizing them and learn for the predictions with using LSTM networks.
– Zhou et al. [20]: use attentive bi-GRU model for learning the hidden representations of titles and bodies. Then, the two learned hidden representations are concatenated and fed into fully connected layers for the prediction.

Different settings are considered in terms of evaluation. We denote the combination of the local similarities and the raw input features as LSDA, and the variant of our method that considers deep local similarity but not the attention by LSD, as shown in Fig. 1(b). For our experimental setting, we initialize the weight and bias parameters with random variables. Besides, we set the word embedding dimension as 100 and the hidden size M as 100. The comparison results are shown in Table 1.

Table 1. Comparison results

Methods	Clickbait Challenge				FNC dataset			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Huang et al. [6]	0.817	0.655	0.661	0.658	0.747	0.894	0.740	0.811
Shen et al. [14]	0.833	0.683	0.643	0.662	0.756	0.959	0.762	0.853
Kumar et al. [7]	0.826	0.699	0.474	0.565	0.859	0.920	0.877	0.907
Zheng et al. [19]	0.844	0.654	0.653	0.653	0.789	0.852	0.845	0.857
Glenski et al. [5]	0.827	0.642	0.621	0.631	0.868	0.925	0.884	0.913
Zhou et al. [20]	0.856	0.719	0.650	0.683	0.879	0.924	0.897	0.919
LSD	0.847	0.697	0.675	0.686	0.885	0.928	0.901	0.923
LSDA	0.860	0.722	0.699	0.710	0.894	0.933	0.912	0.928

The evaluation is conducted with four commonly used metrics: accuracy, recall, precision, and F1 score[7]. Generally, the comparison results show the effectiveness of our proposed method for detecting the clickbait detection. We observed that both the CNN- and RNN-based models perform better than traditional deep neural networks. This may largely resort to the capability of CNN and RNN in capturing the location information. The attention-based bidirectional GRU shows superior performance in dealing with textual information. Besides the superiority of the bi-AttGRU itself, which we use for learning latent representations, both the similarity information and the attention mechanism help with the final prediction.

---

[7] https://en.wikipedia.org/wiki/Precision_and_recall.

4.3   Sensitivity Analysis

In this part, we test the model's sensitivity to different parameter settings on Clickbait Challenge dataset. Similar results can also be found using the FNC dataset. As mentioned above, we mainly have parameters related to the latent representation learning part and the similarity learning part, as well as some hyper-parameters to indicate the learning rate. As the default setting, we separate the training dataset with a ratio of 80%, set the dimensions of word vectors to 50, and pad each sentence to the same length for the input of attentive Bi-GRU model. We also set size of hidden units in Bi-GRU to 50, the local similarity block size to 50 and the default learning rate for Adam optimizer to 0.001.
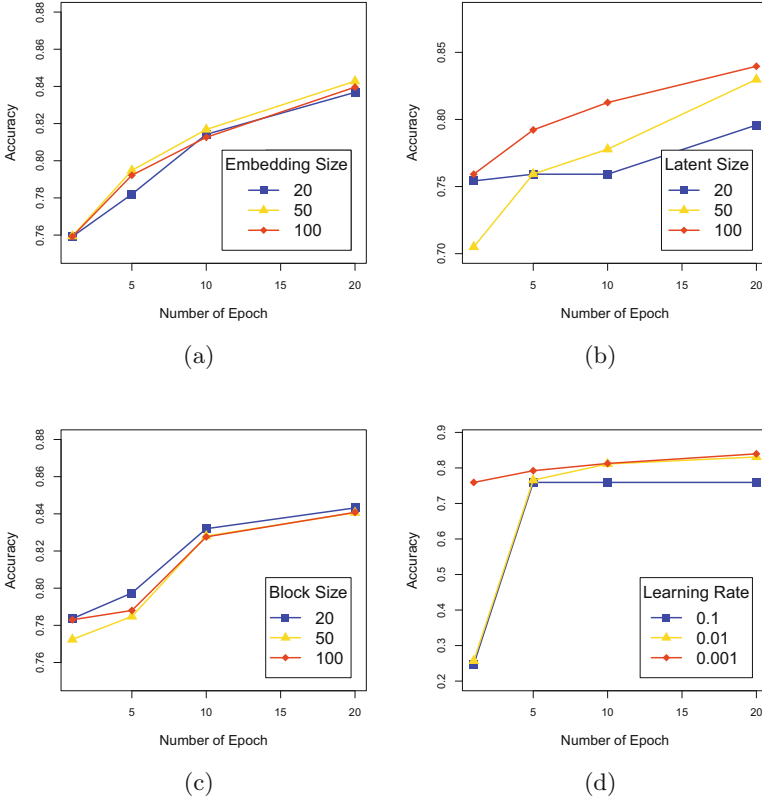

Fig. 2. Sensitivity towards different parameter settings: (a) word embedding size, (b) dimension of latent representations, (c) local similarity block size, and (d) learning rate.

In particular, we compare the following parameters settings: the word embedding size, the latent representation size, the local similarity block size, and the optimizer learning rate. Figure 2 presents the comparison results, where the horizontal axis shows the learning epochs and the vertical axis stands for accuracy.

We can tell that <mark>model with smaller word embedding size have lower accuracy in predicting the clickbait,</mark> which can be the result of the inadequate grasping of the content information in low dimensional word space. On the other hand, the model with large word embedding size requires more time to learn for a decent result. And it can be observed that the dimension of the latent feature representations do affect the results, where larger latent size helps with higher predicting performance. Noted that the average length of bodies of the FNC dataset is 200, thus for training a dataset with word embedding 100, each sample will be sized 200×100, which ends up with latent representations with higher dimensions, and contain relatively more information. For the block size of the local similarities, it can be claimed that a smaller block size performs better than bigger ones. Figure 3 gives an example of the local similarities that with block size 25 and stride 25, thus we have four similarity scores of matching pairs, where the bottom 50 rows are for clickbaits. We can see that for some instances of clickbaits, the subsets of the input are not significantly unrelated. And compared with FNC dataset, the patterns of similarities in Clickbait Challenge dataset are naturally more considerable. Thus it is quite important to automatically weighting the similarity blocks where we used attention mechanism for solving this problem. As for the learning rate, we can observe that larger learning rates make it difficult for the optimization of the models.
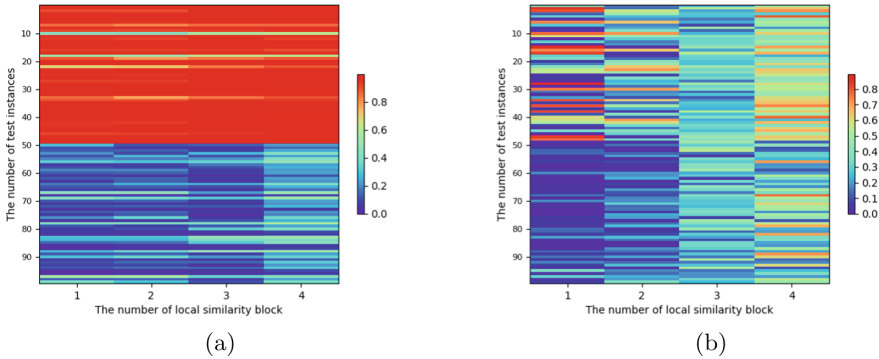

(a)   (b)

Fig. 3. Example of local similarities of correctly predicted instances with block size 25 and stride 25 on (a) Clickbait Challenge dataset and (b) FNC dataset. The top 50 rows are for genuine clicks and the bottom 50 are for clickbaits.

We also consider the impact of different settings for the model to find which part help more with the final prediction. In particular, we consider the following: whether local similarity performs better than global ones; whether adding the latent representations is helpful; and the efficiency of adding the attention. We give five variants of the models, and Fig. 4 shows the results. GS represents the model that considers using only the global similarity for the prediction; LS stands for the model using only local similarities, and similarly, LSA stands
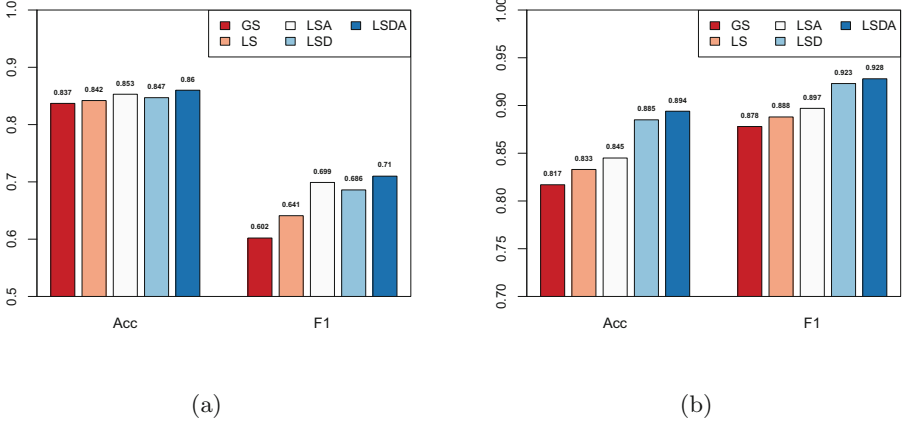
(a)   (b)

Fig. 4. Ablation studies on variants of models on (a) Clickbait Challenge dataset and (b) FNC dataset.

for the model that adds attentions to local similarities, LSD combining latent representations with local similarities, and LSDA is for a combination method shown in Fig. 1(b).

First, we can see that models with concerning local similarities perform better than a model that only considers the global similarity, which might reasons from the sensitiveness of the global similarity to the data noises. Besides, we can observe that concatenating the raw features with the similarity information do help with the prediction, especially on the FNC dataset. While raw features can be used to extract some text patterns like content quality, we can say that these patterns are helpful for detecting the clickbaits. And comparing LSA with LS, and LSDA with LSD, it can be observed that adding the attention mechanism helps to improve the results. Generally, those results indicate the model is effective for extracting essential information in different settings, and further demonstrate the superiority of the attentive similarity in encoding the features.

5   Conclusions

In this paper, we solve the problem of clickbait detection from the similarity perspective, as opposed to the traditional feature engineering which lack the properties in representing the matching information between titles and targeted bodies. We have presented a local similarity-aware deep attentive model that learns both local similarities and raw input features to make predictions in an attentive manner. To the best of our knowledge, the model is novel in the area of clickbait detection and yields competitive results among a series of baseline and state-of-the-arts methods on two real world datasets. Noted that we have not considered other features like image information in this work, which may also be found on those clickbait web-pages. This will be included in our future investigations.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Biyani, P., Tsioutsiouliklis, K., Blackmer, J.: 8 amazing secrets for getting more clicks: detecting clickbaits in news streams using article informality. In: AAAI, pp. 94–100 (2016)
3. Chen, Y., Conroy, N.J., Rubin, V.L.: Misleading online content: recognizing clickbait as false news. In: Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection, pp. 15–19. ACM (2015)
4. Dong, M., Yao, L., Wang, X., Benatallah, B., Sheng, Q.Z., Huang, H.: DUAL: a deep unified attention model with latent relation representations for fake news detection. In: Hacid, H., Cellary, W., Wang, H., Paik, H.-Y., Zhou, R. (eds.) WISE 2018. LNCS, vol. 11233, pp. 199–209. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02922-7_14
5. Glenski, M., Ayton, E., Arendt, D., Volkova, S.: Fishing for clickbaits in social images and texts with linguistically-infused neural network models. arXiv preprint arXiv:1710.06390 (2017)
6. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: International Conference on Information & Knowledge Management, pp. 2333–2338. ACM (2013)
7. Kumar, V., Khattar, D., Gairola, S., Kumar Lal, Y., Varma, V.: Identifying clickbait: a multi-strategy approach using neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1225–1228. ACM (2018)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
9. Lin, Z., et al.: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017)
10. Nasrabadi, N.M.: Pattern recognition and machine learning. J. Electron. Imaging **16**(4), 049901 (2007)
11. Palangi, H., et al.: Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP) **24**(4), 694–707 (2016)
12. Potthast, M., Köpsel, S., Stein, B., Hagen, M.: Clickbait detection. In: Ferro, N., et al. (eds.) ECIR 2016. LNCS, vol. 9626, pp. 810–817. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30671-1_72
13. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, May 2010
14. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: ACM International Conference on Conference on Information and Knowledge Management, pp. 101–110. ACM (2014)
15. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010)
16. Wang, D., Lu, H., Bo, C.: Visual tracking via weighted local cosine similarity. IEEE Trans. Cybern. **45**(9), 1838–1850 (2015)
17. Wang, X., et al.: Truth discovery via exploiting implications from multi-source data. In: Conference on Information and Knowledge Management, pp. 861–870. ACM (2016)

18. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
19. Zheng, H.T., Chen, J.Y., Yao, X., Sangaiah, A.K., Jiang, Y., Zhao, C.Z.: Clickbait convolutional neural network. Symmetry **10**(5), 138 (2018)
20. Zhou, Y.: Clickbait detection in tweets using self-attentive network. arXiv preprint arXiv:1710.05364 (2017)