ISBN: 978-981-18-3959-7
2022 the 12th International Workshop on Computer Science and Engineering (WCSE 2022)
doi: 10.18178/wcse.2022.06.023

News Topic Prediction Via Transformer

Jibing Gong, Kai Yu[+], Chaoyuan Huang, Yuting Lin, Chenglong Wang,
Jinye Zhao, Shishan Gong and Huanhuan Li

School of Information Science and Engineering, Yanshan University, China

Abstract. News topic prediction has gotten a lot of attention recently. Existing methods provide recommendation prediction primarily through content-based or collaborative filtering techniques. However, these methods do not take emotional information embedded in news text into account, they perform comparatively poor in the tasks of news classification and recommendation. To fill this gap, we propose EM-Transformer, a simple encoder-decoder model that incorporates emotional information to improve news topic prediction. Concretely, we begin with improving A-TFIDF, an aligned TFIDF method, for extracting the keywords from news components in a more precise manner. Then, we construct the emotional dictionary by calculating mutual information degree and information entropy. Finally, we combine the original Transformer with emotional encodings for news topic prediction. We conduct extensive experiments on two real-world datasets, publicly available on People's Daily Online and Xinhua Net. The results demonstrate EM-Transformer outperforms classical baselines on the two data sources, and that emotional encoding increases in model quality with 15.97% and 6.37% in accuracy respectively, as oppose to the original Transformer.

Keywords: Transformer, TFIDF, topic prediction, emotional encoding.

1. Introduction

Many news websites have become significant outlets for disseminating information to the world. Certain news reports online would quickly capture the public attention and then become the hot topics and trending news. Topic detection and tracking (TDT) [1] in document streams is an essential task in many important applications for discovering and navigating information. It aims to organize news documents in terms of news events. TDT for news events is used to identify hot topics from the massive amount of publicity and keep abreast of the direction of public opinion. The early approaches [2-3] to topic detection and tracking for news events were to extract entities from news documents using statistics research methods, most of which leverage the extraction algorithm based on Term Frequency–Inverse Document Frequency (TFIDF). But it still has shortcomings of losing positional information of keywords and predicting topic non uniqueness.

Various methods have been developed to establish meta-paths between reports based on heterogeneous information networks inspired by the advancement of deep learning technology. Previous studies [4-6] expressed all relationships in the form of meta-graph and classified topics by calculating the similarity of each meta-path. They conducted comparative experiments by using heterogeneous information networks for topic prediction and obtained promising results by comparing multi social event detection and clustering task models. The difficulty of the method, however, lies in the feature selection on the dataset and the construction of heterogeneous information networks, which makes establishing meta-paths extremely troublesome.

With the deepening of research on attention mechanisms and the development of representation learning, the Transformer based on self-attention [7] has proven to be an effective method of modelling textual sequences. This encoder-decoder architecture has achieved outstanding results in the public benchmarks for natural language processing (NLP) and has a significant improvement effect on the tasks of topic prediction. The vanilla Transformer, however, only takes into account positional encodings to word embeddings and the associations between word-vectors, and lacks the emotional information of each word. This deficiency substantially impedes the generalization of such Transformer to fine-grained tasks like sentiment analysis,

[+] Corresponding author.
E-mail address: 202022040193@stumail.ysu.edu.cn

where it is infeasible to capture emotional semantics in the corpus. Consequently, the Transformer variant should also be designed in a more effective manner. In this paper, the datasets we use are extracted from two news websites, People's Daily Online and Xinhua Net, whose corpora are mainly in Chinese-language. We propose **EM-Transformer**, a simple revised **EM**otion-based **Transformer**, to navigate the usability of emotional information of word-vectors in the structure. For accurate extraction of news report keywords, we introduce **A-TFIDF**, short for **A**ligned **T**erm **F**requency–**I**nverse **D**ocument **F**requency, to avoid indistinguishable keywords among large-scale corpora. Then we construct sentiment lexicon and generate emotional embedding corresponding to each keyword. To this end, we add positional encodings and emotional encodings to the input embeddings at the encoder and decoder stacks. Finally, we carry out topic prediction by passing the newly-generated embeddings into the EM-Transformer. Comprehensive experiments on real-world news data from benchmark dataset showcase better prediction accuracy upon our proposed model, as opposed to other comparative baselines. The contributions of this work include:

- The first emotion-driven framework bases on the Transformer, for integrating emotional information into Transformer, making the best use of the emotional encodings to conduct topic prediction and to systematically study this task on large-scale real-word news datasets.
- We found that the original TFIDF algorithm would cause inconsistent weights of the same keywords in different documents. Thus, we propose an aligned TFIDF algorithm. After sorting the original TFIDF values in reverse order and adding unique minimum values to keywords of the same value in order, the accuracy of keywords has improved by approximately 5% compared to before.
- It is a novel study on construction method of the Chinese emotional dictionary. We use the mutual information degree, information entropy of each word and the emotional seed words to construct the emotional dictionary, and calculate the matching rate of the keywords.

2. Related Work

News Topic Prediction is quite important in the context of NLP issues. Much research has been done in this field and many researchers have made substantial progress. Therefore, we have referred to related literature in mainly threefold for solving any downstream NLP tasks.

2.1. Research Methods Based on Heterogeneous Information Network

Heterogeneous Information Network (HIN) has gained increasing interest recently. It can integrate richer semantic relations and gather more relational information through meta-paths for intent recommendation. He et al. [8] used heterogeneous personalized space random walks to learn the embedding of multiple types of nodes in a HIN guided by meta-path, meta-graph, and meta-pattern, respectively. Most existing HIN embedding methods only consider heterogeneous relations, including adversarial learning on HIN [9]. Establishing meta-paths among reports based on the HIN, the method of topic prediction has achieved good results in experiments [10] by calculating the similarity of each meta-path. It is difficult to make an experiment focusing only on the selection of features in datasets and the construction of HINs.

2.2. Graph Attention Network and Clustering Analysis Algorithm

Graph Attention Network (GAT) leverages masked self-attention mechanism to address the defects of graph convolutional models. It implicitly specifies different weights to different nodes for the center node in the neighbourhood. Heterogeneous GAT [11] is based on the hierarchical attention, including node-level and semantic-level attentions. It employs meta-paths to identify node and semantic features. Song et al. [12] proposed a recommender system for online communities based on a dynamic-GAT. They modelled dynamic user behaviours with a recurrent neural network, and context-dependent social influence with GAT. In our study, we exploit attention mechanism to conduct advanced feature extraction from news text embeddings. Then we use clustering methods such as K-Means to isolate event clusters in a larger set of news reports.

2.3. Transformer

Transformer [7] is a transduction model that relies on self-attention, which has been used widely and has improved the performance of language processing tasks greatly. The Transformer architecture allows efficient unsupervised pre-processing of language models. Ostendorff et al. [13] trained the model combining

Transformer and XLNet that plays a good role in the classification of semantic relations among documents. Transformer currently plays an essential role in the recommendation field due to the addition of the multi-attention head mechanism. For example, multi-head self-attention is also introduced to model user's behaviour sequences for sequential recommendation. Chen et al. [14] adopted the self-attention to model the compatibility in fashion outfit generation.

In this paper we improve the Transformer and propose a new counterpart fused with emotional encodings to translate the input keyword sequence into the corresponding topic category.

3. Methodology

In this section, we introduce our methodology in detail. The construction of EM-Transformer consists of three steps. Firstly, we construct the emotional dictionary and match the keywords with it to generate the emotional embeddings. Secondly, we project word embeddings, emotional embeddings and positional embeddings into the same dimensional space, and then apply element-wise summation. Lastly, we pass them into the pure Transformer model and use the self-attention mechanism with a fully connected network to calculate the final embeddings.

3.1. Emotional Word Matching and Encoding

We need to calculate pointwise mutual information (PMI) and left-right information entropy (LRE) to conduct new word detection when constructing emotional lexicon. Specifically, we calculate PMI and LRE between emotional seed words and keywords which are from corpus for Chinese word segmentation. Then we choose the Top-K words with the highest correlation degree with respect to the emotional words based on the PMI and LRE combination. We define the PMI of a pair of x and y to be

$$PMI(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

where empirical probabilities are computed over a particular data set of interest. x and y are all the words in the Chinese-language. $p(\cdot)$ is the probability of the occurrence of one word.

LRE is derived from the information entropy on the left and right of a word segment. Its value can be used to indicate whether a word owns fickle left-right collocations. It is believable that when a word owns higher entropy, the one-hop of this word is more variable, and the preselected word would probably be a single word. We compute LRE which are given as follows, where A and B are the word sets between word W, respectively.

$$E_{Left}(W) = -\sum_{\forall \alpha \subseteq \mathrm{A}} P(\alpha W|W) \log_2 P(\alpha W|W)$$

$$E_{Right}(W) = -\sum_{\forall \beta \subseteq \mathrm{B}} P(W\beta|W) \log_2 P(W\beta|W) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

And now we can combine PMI and LRE to form the final emotional dictionary. Then we match each keyword with the emotional words in the dictionary and get the corresponding emotional value to prepare for formulating the emotional encodings.

The encoder adds positional encodings to the input embeddings. These positional embeddings conform to a specific pattern which records the positional information of each word and the distance between different words in the sequence. The keyword embeddings should not only take positional encodings into account but emotional ones, which derives from emotional values based on aforementioned calculation. The emotional encodings have the same dimension as the word embeddings and positional embeddings, so that the three can be summed together. For example, the dimension of the input embeddings is 4, whilst the actual positional encodings and emotional encodings are shown in Fig. 1 (a). There are also many opportunities of emotional encodings, the same as positional encodings, learned and fixed.

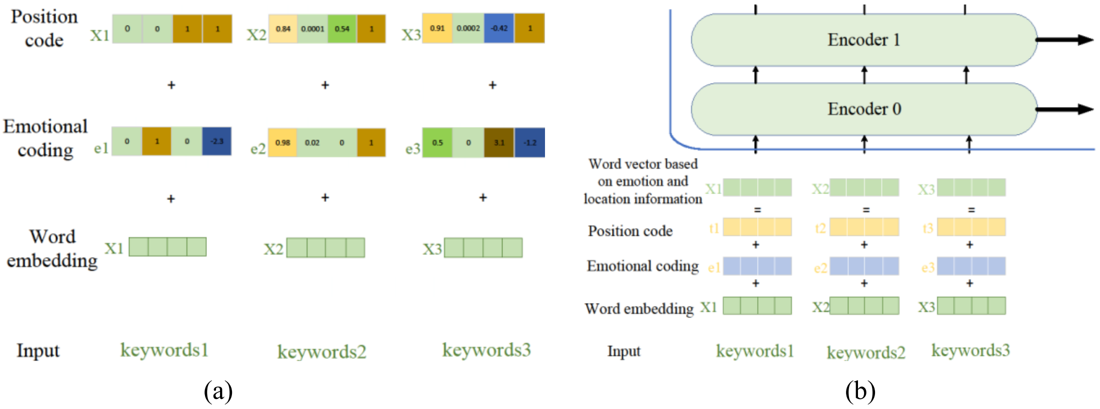


Fig. 1: Word-vector embedding diagram.

3.2. Embedding with Arbitrary Contexts

Estimating the semantic similarity in the context is a challenging task in the field of NLP. Intuitively, words that appear in similar contexts would have similar embeddings. The context vocabulary C is thus identical or similar to the word vocabulary W. It seems to undermine the effectiveness of information gaining in emotional dictionary. However, when we obtain the summation embeddings aggregated from the word encodings, the positional embeddings, and the emotional embeddings, we show that the finally embeddings of the similar words are very different empirically, which means they are semantically distinct in arbitrary contexts. Afterwards, we feed the new embeddings to the encoder layer of the naive Transformer in order to conduct the higher-level features extraction. The specific learning process is shown in Fig. 1 (b).

3.3. Encoder-Decoder Stacks and Self-Attention

The key difference between EM-Transformer and LSTM is that the training of LSTM is iterating in sequence serially, which can only process the next word after finishing the current word. In contrast, the training of EM-Transformer can be done in parallel. It greatly improves computation efficiency because words can be trained simultaneously. Transformer models are divided into encoder and decoder. The main function of the encoder is to obscure the input sequence into a hidden layer, it then returns the translated natural language sequence through the decoder. EM-Transformer adds positional and emotional encodings to understand the relation of words in the article and use the self-attention mechanism to calculate with the fully connected layer.

EM-Transformer uses positional encodings and emotional encodings on the basis of word-vector to represent the positional and emotional information of words appearing in sentences. Because EM-Transformer does not adopt the RNN structure of the circular neural network, the parallel processing of global information is adopted. Additionally, disregarding word order information would have a detrimental effect on NLP tasks. As a result, EM-Transformer saves the relative or absolute position of words in the sequence using locational encoding. The positional encoding is represented by PE, whose dimension d is the same as the word-vector. PE can be obtained by training, and can also be obtained by using the cosine function. The latter is used in EM-Transformer. Moreover, sine position code is used for even position words, while the other is used for the odd ones. The two frequencies are defined as follows.

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

Different periodic changes are generated by sine and cosine function processing. The periodic changes of positional embedding in dimension would gradually slow down with the increase of serial number, and finally the texture containing positional information is generated. The period of positional embedding function changes from 2π to $10000 \times 2\pi$, and each position in dimension would get the value combination of sine and cosine function in different periods. In this way, the unique positional textural information is generated. And finally, the model learns the spatial-temporal features of natural language. For the input sentence X, we sum up the word-vector and the its corresponding positional vector, both of which can be summed up directly because of the same dimensions. Actually, we mark the vector of the Tth word in a sentence as X_t.

EM-Transformer would then carry out the self-attention mechanism. Firstly, three matrices W_Q, W_K, W_V are defined. They are respectively used to conduct three linear transformations on all word-vectors to generate three new vectors q_t, k_t, v_t, which are respectively concatenated into query matrices, key matrices, and value matrices. To obtain the attention weight of the first word, it is necessary to multiply the query vector q_1 of the first word by the key matrix K. Then we pass them into the $softmax$ function to have values sum up to one. After obtaining the weight, we multiply the weight by the vector v_t of the corresponding word respectively, and sum up the weighted values to get the output of the first word. Finally, we repeat the same steps for the other input vectors to get all the outputs by the self-attention mechanism. The process is shown in Fig. 2.

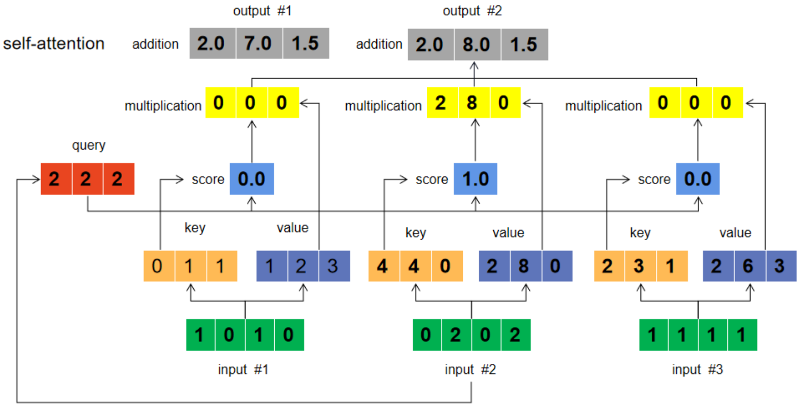


Fig. 2: Self-attention calculation diagram.

The self-attention mechanism formulae are shown as follows.

$$Q = Linear(X) = XW_Q$$
$$K = Linear(X) = XW_K$$
$$V = Linear(X) = XW_V$$
$$X_{attn} = SelfAttention(Q, K, V) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

Then we make residual connection. The formula for calculating residual connection is as follows.

$$X_{attn} = X_{attn} + X \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5)$$

The effect of Layer Normalization is to normalize the hidden layers in the neural network into standard normal distribution to accelerate the training and convergence. We calculate the mean and variance of each hidden layer respectively. The $LayerNorm$ process is shown as follows, where m denotes the number of hidden units in a layer, and ε is a value added to the denominator for numerical stability.

$$\mu_l = \frac{1}{m}\sum_{i=1}^{m} x_i^l$$
$$\sigma_l = \sqrt{\frac{1}{m}\sum_{i=1}^{m} (x_i^l - \mu^l)^2}$$
$$LayerNorm(x) = \frac{x - \mu_l}{\sqrt{\sigma_l^2 + \varepsilon}}$$
$$X_{attn} = LayerNorm(X_{attn}) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (6)$$

Feedforward in the encoder is actually a two-layer linear transformation and activated by activation function (we use $Relu$ in our work). The formula is shown as follows.

$$X_{attn} = Linear\left(Relu\left(Linear(X_{att})\right)\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (7)$$

After feedforward in the encoder, we need to make a residual connection again for feedforward and have the normalization of the hidden layer, as shown in the following.

$$X_{hid} = X_{attn} + X_{hid}$$
$$X_{attn} = LayerNorm(X_{hid}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

In the section, we detail the entire construction of the encoder where the decoder is the reverse operation of the encoder. This architecture actually establishes a mapping mechanism among input words which completes the from keywords to topics prediction task by taking topic category as the output part of the decoder. We elaborate the learning process in Algorithm 1.

4. Experiments

4.1. Dataset

Our proposed model is evaluated on the real-world datasets publicly available on People's Daily Online and Xinhua Net, which are mainly in Chinese-language. After filtering out repeated and irretrievable news text, the dataset we use entirely contains 63,395 labelled news reports related to several event classes, respectively, spread over about 3 years (from 2019 to 2021). The topic clusters include domain-specific events such as Real-Time Politics, Social Legality and Finance. The number of reports in each topic ranges from 600 to 15,000. The quantitative distribution of the topic clusters is shown in Table 1.

ALGORITHM 1: EM-Transformer: Emotion-based Transformer for news topic prediction

```
Require: Multiple topics in the news corpus are stratified into training set and test set.
Initialize: Collect News Document Sets; Process Corpora.
Initialize: Parent Keyword List generated by jieba text segmentation for headline and body; Top-K.
Initialize: vanilla Transformer.
Output: News topic prediction category.
// loop for generating keywords
for $c \in Parent\ Keyword\ List$ do
    for t ∈ $Corpus$ do
        // Generate news keywords list by A-TFIDF
        $Keyword\ List \leftarrow Keyword\ List \cup \{t\}$
    end for
end for
return Keyword List
// EM-Transformer
for $i \rightarrow Keyword\ List$ do
    // Get emotional words encoding for $i$
    Calculate $PMI$ via Eq. (1)
    $W \leftarrow i$, Calculate $LRE$ via Eq. (2)
    Combine PMI and LRE to calculate emotional values so as to form the emotional dictionary
    // The corresponding topic prediction value is obtained by Transformer
    Final Embedding ← Input Embedding + Positional Embedding + Emotional Embedding
    Topic ← Transformer ($Final\ Embedding$)
end for
return all prediction results $p$
```

Table 1: The descriptions of the dataset

Data Source	Topic classification	Number
People's Daily Online	Finance	689
	Sports	2755
	Technology	1605
	Real-Time politics	3255
	Military	4585
	Science	9632
Xinhua Net	Business	14659
	Nomocracy	2689
	Culture	3644
	Travel	9853
	Photo	2069
	Finance	7960

4.2. Data Preprocessing

As part of the data pre-processing phase, we first formalize corpora, so that each sample has two parts: the headline part and the body part. Then we use jieba to segment each pair of headline and body. In the keyword extraction stage, a specified number of keywords are extracted from each report heading and body by the TFIDF algorithm [15], including named entities such as person name and place name. TFIDF is a method for extracting keywords from documents using statistical analysis theory so as to reflect how important a word is to a document in a collection or a corpus. It contains the frequency information and discrimination power of a keyword. The value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word.

The TFIDF is the product of two statistics, term frequency and inverse document frequency. More specifically, term frequency (TF) refers to the probability of a word appearing in an article. It is usually defined as follows, where $f_{t,d}$ is the number of times that word t occurs in article d, and the denominator is the total number of words in the same article d.

$$TF = \frac{f_{t,d}}{\sum_{t'\epsilon d} f_{t',d}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (9)$$

The inverse document frequency (IDF) is a measure of how much information a word provides. If the word repeats in many articles, its ability to distinguish the article will be weakened. It is a logarithmic function where the denominator represents the number of documents containing the keyword. To avoid the division-by-zero error, it is common to adjust the denominator by adding a small positive number. The IDF formula is defined as follows, where N is the total number of articles in the corpus.

$$IDF = \log \frac{N}{|\{d\epsilon D: t\epsilon d\}|} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

In general, the more frequent a word appears in an article and the fewer times it occurs in the overall corpus, the better it represents the article. The TFIDF is defined as follows.

$$TFIDF = TF \times IDF \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (11)$$

Each article is represented as a collection of different types of keywords by mean of the keyword extraction. Although each keyword has an independent weight, it is calculated from a single report rather than all articles, which results in inconsistent weights for the same keyword in different articles. Therefore, the inverse document probability value of word frequency for each keyword in all keyword collections needs to be calculated again. For keywords with the same frequency, the original TFIDF may derive the same results, which would bring a great challenge to keyword extraction. Therefore, an aligned TFIDF algorithm is presented. It arranges the original TFIDF values in reverse order and adds the keywords with the same values a unique minimum w in order, where w is the product of the number n with the same values and the uniform minimum o. The minimum w is defined as follows.

$$w = n \times o \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (12)$$

Our improved A-TFIDF ensures that each keyword has a unique numeric. The equation is calculated as follows.

$$A - TFIDF = TFIDF + w \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (13)$$

The A-TFIDF value calculated by keywords is used to represent entity information respectively in each news report. Thus, we could sequentially combine the values to form the textual vector in one article. For data with insufficient keywords, the minimum values will fill in the corresponding vector during the subsequent vectorization process. In this phase, we utilize the aligned TFIDF algorithm to extract news keywords information, which constitutes the news keywords dataset for our following experiments.

4.3. Effectiveness of EM-Transformer

To evaluate the effectiveness of the EM-Transformer, we train the model by using the news text data of specified topics from People's Daily Online and Xinhua Net respectively, and take 1000 news reports as the test set from each of the two original sources. The test set data has been assigned topic labels during the collection stage. Our experiment masks the labels of the test set to verify the performance of EM-Transformer. The model inputs word embeddings, positional encodings, and emotional encodings of the report keywords and outputs the topic category corresponding to the report. The probability value of each topic can be obtained through $softmax$ function after training numerous epochs. In general, the higher the probability value is, the more likely it belongs to the certain category. It is shown that the loss gradually decreases during the model training. And the loss function and accuracy curve are shown in Fig. 3.

4.4. Hyperparameter Analysis

We employ the evaluation indicators including Accuracy (ACC), Recall (R), Precision (P), and their harmonic average F1 score, to evaluate the recommendation performance of all models in our experiments. Some hyperparameters play essential roles in the model performance. Here we explore three hyperparameters in EM-Transformer on the basis of ACC and Macro-F1. One is the number of layers of the encoder and decoder, another is the embedding size, and a third is the number of attention heads.

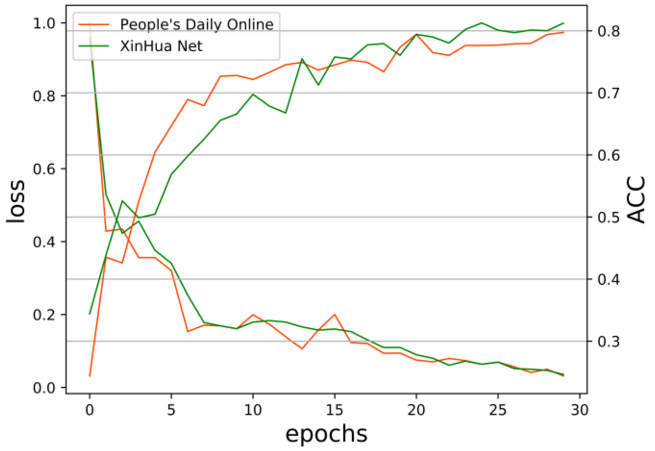
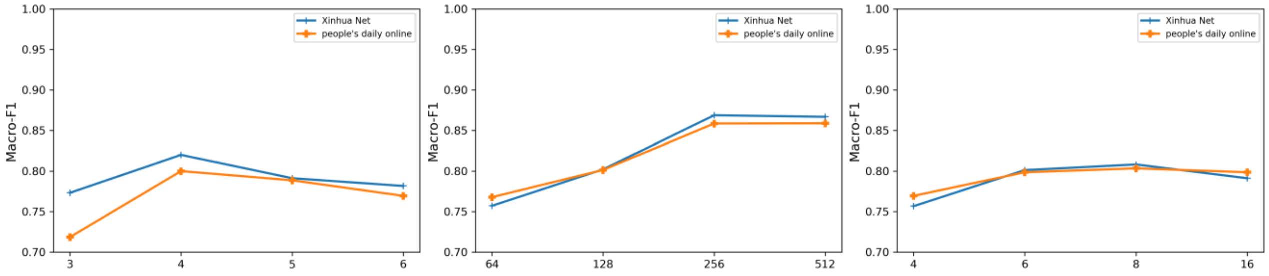


Fig. 3: Accuracy and loss diagram.



(a) (b) (c)

Fig. 4: Influence of three hyperparameters under metric Macro-F1.

Number of encoder and decoder layer. The number of encoder and decoder layers is searched in $\{3,4,5,6\}$. The effect of the number of encoder and decoder layers on the performances in Fig. 4 (a) shows a significant trend. That is, the performance increases with an increasing number of layers and then decreases after reaching a threshold. Specifically, the performance is poor when $n = 0$, because the number of model layers is insufficient, leading to the lack of learning ability and resulting in low accuracy and recall. The best performance is achieved when $n = 4$. And the performance remains good when $n = 5$, indicating that multiple self-attention calculations are helpful for learning complex semantic relationship interaction. The reason that the performance begins to sharp decline when $n = 6$ is that the model becomes more and more complex. However, the semantic relationship between the keywords does not necessitate so complex. We set $n = 4$ after considering the influence of the number of encoder and decoder layers.

Embedding size. In our model, the dimension size is relevant not only to the item embedding size but also the dimension of the projection matrices in self-attention. We searched embedding sizes in $\{64,128,256,512\}$, and the results are shown in Fig. 4 (b). We observed that the larger embedding dimension makes the model have a better effect on accuracy and recall. Considering the efficiency and the cost of the model, we set the embedding size to 512 on both datasets.

Number of attention heads. In the EM-Transformer model, the number of self-attention heads is crucial for learning context representations. The model is capable of tackling the information from different representation subspaces by means of multi-head self-attention. We varied the number of self-attention heads of $\{2,4,6,8\}$ to optimize the performance of our model, and illustrated its influence in Fig. 4 (c). There is a significant raise trend for performance when the number of heads increases from 4 to 8, as the rich textual semantics is fully exploited and stationarily represents when there are a few heads. However, it declines a little when the number of heads increases from 8 to 16. Thus, the outcome of unreasonable number of heads tends to be local optimal. Based on the aforementioned discussion, we set the number of attention heads to be 8.

4.5. Comparison with Baselines

To verify the effectiveness of EM-Transformer, a large number of experiments have been done to compared with traditional classification and cluster in the tasks of topic prediction. We made comparisons with the following representative baselines to evaluate the performance of our proposed model. The topic prediction results of the controlled experiments separately on two datasets are shown in Table 2.

- **GCN** [16] is a feature extractor for graph data, which has the same function as CNN. However, this model cannot calculate the different influence of different nodes in the same order neighborhood for the center node but adopts the sharing weight mechanism, which limits ability of the GCN to capture the different features in the information space.
- **GAT** [17] is proposed to solve the problems of GCN. It uses semantic level attention to detect meta-path differences. Center nodes can pay attention to their neighborhoods through the self-attention layers, which gives attentional weights to different neighboring nodes and extracts node-level features.
- **HAN** [11] is a new heterogeneous GNN based on attention mechanism. This architecture includes node-level attention and semantic-level attention. The node attention mainly learns the weights between center nodes and their neighbors, and semantic attention learns the consequences based on different meta-paths. The final node representation is obtained through corresponding aggregation.
- **Transformer** [7][18] has abandoned traditional CNN and RNN. The whole network structure is completely composed of the self-attention mechanism. A trainable neural network based on Transformer can be built by stacking Transformer. It uses only the attention mechanism to encode each position, to relate two distant words of both the inputs and outputs in respect of itself.

Table 2: Topic prediction results compared to the baselines

Data Source	**Method**	**ACC**	**Macro-F1**
People’s Daily Online	GCN	0.588	0.603
	GAT	0.652	0.676
	HAN	0.698	0.688

	Transformer	0.745	0.812
	EM-Transformer	**0.864**	**0.840**
Xinhua Net	GCN	0.602	0.596
	GAT	0.638	0.654
	HAN	0.672	0.694
	Transformer	0.785	0.824
	EM-Transformer	**0.835**	**0.830**

We carried out five sets of comparative experiments each of two data sources. The experimental results show that GCN performs worse than any other models. Because GCN uses the same weight matrix when generating the feature matrix, and it does not specifically calculate the attention coefficient. Besides, the performance of HAN is better than GAT. The reason is that the HAN model separately enhances the training of the GAT model on meta-paths. Although the effect of comparing a single GAT model is improved, the increase of the accuracy of HAN is not particularly obvious. Moreover, the Transformer has greatly improved in all indicators compared with other baseline models. Also, it is still not difficult to find that EM-Transformer exceeds 15.97% and 6.37% the traditional Transformer on the two data sources in terms of Accuracy, respectively. Note that we initialized the network parameters with the same seed across all the experiments to keep reasonability and the same configurations were used for multi-scenario learning experiments.

5. Conclusion and Future Work

In this paper, we conducted a novel study to make topic prediction in a large number of news reports. We defined the problem precisely and proposed a news topic prediction algorithm based on Transformer. More specifically, we integrated emotional information into the Transformer to recognize the news topics. The experimental results demonstrate that EM-Transformer achieves excellent performance in terms of Accuracy and Macro-F1 compared with several baseline methods on two real-world datasets.

We are excited about the future of emotion-based models and aim to further our study based on emotional encoding in the domain of NLP. Also, it is interesting to extend our model to other recommendation tasks.

6. References

[1] L. Chen, H. Zhang, J. M. Jose, H. Yu, Y. Moshfeghi, and P. Triantafillou, "Topic detection and tracking on heterogeneous information," *Journal of Intelligent Information Systems*, vol. 51, no. 1, pp. 115–137, 2018.

[2] C. Wang, X. Zhao, Y. Zhang, and X. Yuan, "Online hot topic detection from web news based on bursty term identification," in *Asia-Pacific Web Conference*. Springer, 2016, pp. 393–397.

[3] S. D. Tembhurnikar and N. N. Patil, "Topic detection using bngram method and sentiment analysis on twitter dataset," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO).* IEEE, 2015, pp. 1–6.

[4] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," *arXiv preprint arXiv:1906.04580,* 2019.

[5] C. Wang, Y. Song, H. Li, Y. Sun, M. Zhang, and J. Han, "Distant meta-path similarities for text-based heterogeneous information networks," in *Proceedings of the 2017 ACM on conference on information and knowledge management*, 2017, pp. 1629–1638.

[6] D. Yang, Y. Xiao, H. Tong, W. Cui, and W. Wang, "Towards topic following in heterogeneous information networks," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* IEEE, 2015, pp. 363–366.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng, "Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 639–648.

[9] Y. Lu, C. Shi, L. Hu, and Z. Liu, "Relation structure-aware heterogeneous information network embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4456–4463.

[10] S. Sajadmanesh, S. Bazargani, J. Zhang, and H. R. Rabiee, "Continuous-time relationship prediction in dynamic heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 4, pp. 1–31, 2019.

[11] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032.

[12] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the Twelfth ACM international conference on web search and data mining*, 2019, pp. 555–563.

[13] M. Ostendorff, T. Ruas, M. Schubotz, G. Rehm, and B. Gipp, "Pair-wise multi-class document classification for semantic relations between wikipedia articles," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, 2020, pp. 127–136.

[14] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, and B. Zhao, "Pog: personalized outfit generation for fashion recommendation at alibaba ifashion," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2662–2670.

[15] I. Yahav, O. Shehory, and D. Schwartz, "Comments mining with tf-idf: the inherent bias and its removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 437–450, 2018.

[16] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.

[18] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, 2021.