International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-022-01734-0

ORIGINAL ARTICLE

Topic-aware hierarchical multi-attention network for text classification

Ye Jiang[1] · **Yimin Wang**[1]

Received: 25 April 2022 / Accepted: 26 November 2022

Abstract
Neural networks, primarily recurrent and convolutional Neural networks, have been proven successful in text classification. However, convolutional models could be limited when classification tasks are determined by long-range semantic dependency. While the recurrent ones can capture long-range dependency, the sequential architecture of which could constrain the training speed. Meanwhile, traditional networks encode the entire document in a single pass, which omits the hierarchical structure of the document. To address the above issues, this study presents T-HMAN, a **T**opic-aware **H**ierarchical **M**ultiple **A**ttention **N**etwork for text classification. A multi-head self-attention coupled with convolutional filters is developed to capture long-range dependency via integrating the convolution features from each attention head. Meanwhile, T-HMAN combines topic distributions generated by Latent Dirichlet Allocation (LDA) with sentence-level and document-level inputs respectively in a hierarchical architecture. The proposed model surpasses the accuracies of the current state-of-the-art hierarchical models on five publicly accessible datasets. The ablation study demonstrates that the involvement of multiple attention mechanisms brings significant improvement. The current topic distributions are fixed vectors generated by LDA, the topic distributions will be parameterized and updated simultaneously with the model weights in future work.

Keywords Text classification · Topic model · Attention mechanism · Natural language processing

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP), and it aims to automatically assign label(s) to a given text. Traditional text classification approaches leverage word co-occurrence information as feature representations to train a classifier, and such representations could be typically obtained by Bag-of-Word (BoW) [1, 2], term frequency-inverse document frequency (TF-IDF) [3] or Latent Dirichlet Allocation (LDA) [4] topic distributions [5–7]. Those methods transform unstructured text into numerical data, so the large-scale text data could be structured, classified, or clustered automatically. Although such representations are feasible for tasks such as document summarization [8, 9], document classification [10–12] and clustering [13], they suffer from dimensional sparsity, which leads to high computational cost, and also miss contextual information in the text sequences [14].

For the past few years, neural networks have been commonly implemented in text classification tasks [15, 16]. They typically encode texts of varying lengths into vector representations of fixed lengths, on top of which a classifier is added. Especially, convolutional neural networks (CNNs) [17–19] are used to extract features by applying convolution filters (also called kernels) over input sequences. Recurrent neural networks (RNNs) [20, 21] perform a similar function but for each time step of the input sequences, the current output is dependent on the previous computations. CNNs are computationally fast and capable of extracting local and position-invariant features. However, they are not good at the classification tasks that are determined by a long-range semantic dependency rather than local key phrases. RNNs are able to capture long-range dependency, the computation is rather slow as it is recurrent in nature.

Recently, the Transformer [22], which entirely relies on attention mechanisms [23], demonstrates the state-of-the-art performance in many NLP tasks. Since then, attention mechanisms have been successfully applied to several fields including text summarization [24], natural language

✉ Yimin Wang
yimin.wang@qust.edu.cn

Ye Jiang
ye.jiang@qust.edu.cn

[1] School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China

Published online: 14 December 2022

inference [25], sentiment analysis [26] and image classification [27]. Specifically, self-attention, which is an attention mechanism, is able to model sequences by connecting long-range dependency via a shorter network path compared to RNNs, and could also be trained as fast as CNNs. As a variant of self-attention, co-attention [28] is proposed to simultaneously attend to two different feature representations, and jointly use one attention to guide another. Recent research [26] found that self-attention is superior in performance to CNNs and RNNs on classification accuracy in sentiment analysis, but less is known about the effectiveness of co-attention in the document classification task.

Moreover, traditional neural networks [29, 30] scan an entire document in a single pass and omit the structural features between words and sentences as well as between sentences and documents. In an attempt to resolve this issue, hierarchical neural architectures [20, 31, 32] take sentence-level and document-level inputs to a sentence encoder and a document encoder separately, and achieved state-of-the-art accuracy on many text classification tasks.

In this study, we propose **T**opic-aware **H**ierarchical **M**ulti-**A**ttention **N**etwork (T-HMAN), which builds in the form of a hierarchical structure with multiple attention mechanisms. The main contributions of this study are summarized as follows:

1. Different from other hierarchical models, a sentence co-attention is developed to learn sentence representations with different levels of abstraction interactively and is able to aggregate sentence semantic features in the document encoder at different stages.
2. Feature representations at word-level and document-level are both enriched by using LDA topic model to incorporate global co-occurrence information from the topic-word distributions, and the per-document topic probabilities from the document-topic distributions.
3. Soft attention mechanism is employed to summarize the importance of the sentence-level and document-level representations sequentially since not all parts of a word or a sentence are equally important for the model prediction.
4. To measure the performance of the proposed model, five publicly accessible document classification datasets are evaluated. The proposed T-HMAN outperforms previous approaches and also converges faster. The experiment also intensively compares hierarchical with non-hierarchical neural network architectures and demonstrates that hierarchical ones are typically superior to non-hierarchical ones in text classification tasks.

2 Related work

CNNs and RNNs have been widely applied to encode document representations. For instance, Ruchansky et al. leveraged a RNN to extract temporal text representations for detecting fake news [33]. Kim adopted a CNN to classify documents [17]. Wei el al. developed a CNN for stance detection in tweets [16]. Since Bahdanau et al. introduced the attention mechanism [23], later referred to as soft attention by [34], which improved the encoding capacity of sequence-to-sequence architectures in machine translation tasks, it has been intensively combined with various RNN and CNN architectures [28, 35, 36]. Soft attention is not only able to capture long-range semantic dependencies, but also to focus on the most salient features [37, 38]. Recently, Vaswani et al. introduced the Transformer which was entirely built based on the attention mechanism [39]. They demonstrated that self-attention could be applied directly to word embeddings to capture important relations and semantic information over a long distance, similar to RNNs. Meanwhile, the feed-forward architecture of self-attention could be trained as fast as CNNs. Compared with RNNs and CNNs, Ambartsoumian and Popowich have also shown that self-attention could achieve better classification accuracy and fast convergence speed in sentiment analysis [26].

A problem with these neural network models is that they generate document representations without considering the characteristics of the hierarchical structure of documents. To address this issue, Yang et al. proposed HAN [20], which achieved state-of-the-art performance, and indicated such hierarchical information has the potential to come up with better document representations. Hierarchical models have been applied to many text classification tasks. Gao et al. proposed a hierarchical convolutional attention (HCAN) model [31], which employed both self-attention and target-attention. Abreu et al. combined RNNs with CNNs in a hybrid hierarchical attentional neural network (HAHNN) for document classification tasks [32]. Jiang et al. proposed Embeddings from Language Model (ELMo) Sentence Representation Convolutional (ESRC) network [40], which sent contextual word embedding to a hierarchical architecture for classifying hyperpartisan news. Zheng et al. explored various hierarchical encoders on documents with varied document lengths and revealed that neural-network-based hierarchical architectures outperform their non-hierarchical counterparts for document classification [14]. Shu et al. proposed an explainable fake news detection (dFEND) network [41], which applied co-attention [28] on both news and comment inputs simultaneously, the attention weights can be shared and learn meaningful interactions between the two inputs. Tian et al.

proposed a hierarchical inter-attention network in a multi-task document classification task [42], they developed an inter-attention that shares the global information between tasks, so the components can provide common and task invariant knowledge. Liu et al. developed a hierarchical graph convolutional neural network [43], which combined a section graph network with a decoupled graph convolutional block to capture the macrostructure and fine-grained features of a document. Li et al. also proposed a hierarchical Transformer-CNN model for multi-label text classification tasks by applying two Transformer encoders on word-level and sentence-level semantic features [44]. However, the document representations were simply obtained by concatenating feature representations via a convolutional network.

Meanwhile, LDA has been employed in many document classification tasks [45, 46]. For instance, Wu et al. combined LDA with SVM to classify Chinese news, outperforming models that generate high-dimensional features such as TF-IDF models [47]. Li et al. developed LDA with a Softmax regression to overcome the curse of dimensionality of the news text [7]. Kim et al. regarded document-topic distributions generated by LDA as a document representation [48], in which both word frequencies and semantic information are considered, to enhance the performance of document classifiers. Lin et al. introduced a joint sentiment/topic model for detecting sentiment and topic from text simultaneously [49].

Recent neural network approaches have been combined with LDA for generating document representations. Liu et al. applied LDA to build topic-based word embeddings considering both words and topics [50]. Xu et al. adapted LDA to capture topic-based word relationships, which were then integrated into distributed word embeddings [51]. Wang et al. prepared LDA-based text features as input to a deep neural network to detect automobile insurance fraud [52]. Narayan et al. introduced a topic-aware convolutional neural network to generate summaries from online news articles [53]. Specifically, the topic model was used to generate document-topic distributions and word-topic distributions respectively. Then, a CNN was applied to encode and decode the combination of topic distributions and input representations sequentially. Jiang et al. also compared text classification performances of hierarchical and non-hierarchical models that integrate topic distributions [54].

3 Topic-aware hierarchical multi-attention network

The overall workflow of T-HMAN is shown in Fig. 1. The T-HMAN model aims to capture contextual semantic information in documents by taking word-level and sentence-level inputs through a sentence encoder and a document encoder sequentially. Specifically, the model first encodes word-level inputs to sentence-level representations, which are then taken as the inputs to learn the document-level representations. Finally, the document-level representations are passed to a Softmax for classification. We will discuss each component in detail in the following subsections.

3.1 Pre-training topic distributions

Let $D = (s_1, s_2, \ldots, s_m)$ denotes a document consisting of a sequence of m sentences; $S = (w_1, w_2, ..., w_n)$ stands for a sentence comprising n words. S is embedded into a distributional space $we = (we_1, we_2, \ldots, we_n)$ where $we \in \mathbb{R}^{n \times e}$, n is the sequence length and e is the dimension of word embeddings.

The LDA model generates topic-word and document-topic distributions simultaneously. The former is shared between all documents and contains global word co-occurrence features in the whole corpus; while the latter are local distributions over the topics for given documents, and are independent of all the other documents. These two distributions can be used as additional features in the sentence and document encoders hierarchically. Let $tw = (tw_1, tw_2, ..., tw_n)$ denotes the transposed topic-word distributions where $tw \in \mathbb{R}^{n \times t}$, t is the number of topics, and the document-topic distributions can be represented as $dt = (dt_1, dt_2, ..., dt_d)$ where $dt \in \mathbb{R}^{d \times t}$, d is the number of documents. The LDA model is pre-trained first. The word embeddings are then concatenated with the transposed topic-word distributions $[we; tw]$ as the input to the sentence encoder. Similarly, the document representations are concatenated with the document-topic distributions $[de; dt]$ before the final Softmax classifier is applied.

3.2 Positional encoding

In contrast to the recurrent mechanisms, self-attention does not explicitly model relative or absolute position information in its structure. We therefore apply element-wise addition between word-position embeddings $wp \in \mathbb{R}^{n \times e}$ and the word embeddings to get word-level inputs $w = ([we_1;tw_1] + wp_1, ..., [we_n;tw_n] + wp_n)$, as well as between sentence-position embeddings $sp \in \mathbb{R}^{m \times d_k}$ where d_k denotes the model dimension, and the sentence embeddings $se \in \mathbb{R}^{m \times d_k}$, to get sentence-level inputs $s = (se_1 + sp_1, ..., se_m + sp_m)$.

Two positional encoding approaches are investigated: Sinusoidal position encoding and learned position encoding.

Sinusoidal position encoding: This method was introduced with the Transformer [39] model, in which sine and cosine functions of various frequencies are used to form a geometric progression, and then added to the original word

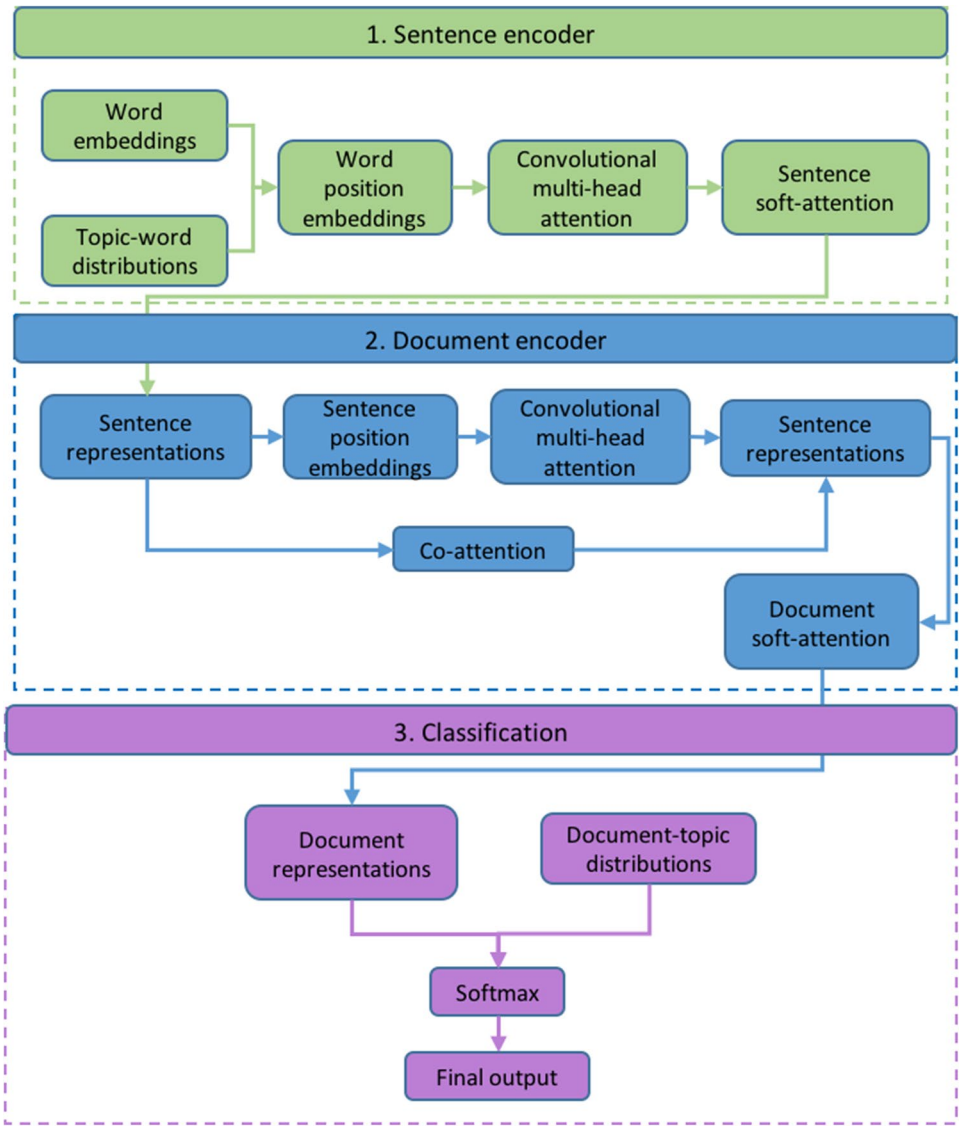

Fig. 1 The flowchart of the proposed model

embeddings. The positional embeddings are d_k dimensional vectors that contain information about specific positions in a sequence.

$$PE_{(}pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_k}}}), \tag{1}$$

$$PE_{(}pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_k}}}). \tag{2}$$

where pos stands for the desired sentence position in a sequence, i is the encoding dimension. Models receive orders of words generated by sinusoidal functions even when predicted sequences are longer than those seen during training phases.

Learned position encoding: Similarly, this method [55] generates vector representations of an absolute position of an entry in a sequence. The position embeddings are then added to the original embeddings. Vaswani et al. reported that this method performs identically to sinusoidal position encoding [39]. This study applies randomly initialized embeddings that learn the absolute positions of words during training for fast implementation.

3.3 Self-attention

Self-attention aims to attend to each position of a sequence by comparing each entry to other entries in the same sequence, which enables the model to learn contextual relations of the sequence as well as capture long-range semantic dependencies. Therefore, the outputs of self-attention contain the information of each entry as well as how it relates to all the entries. In this study, we apply scaled dot-product attention [39],

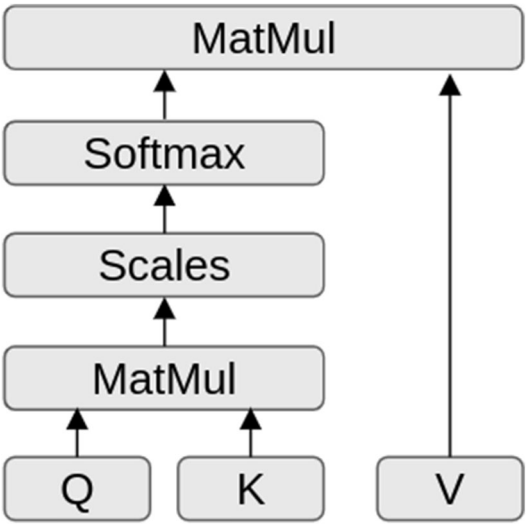

Fig. 2 Scaled dot product attention

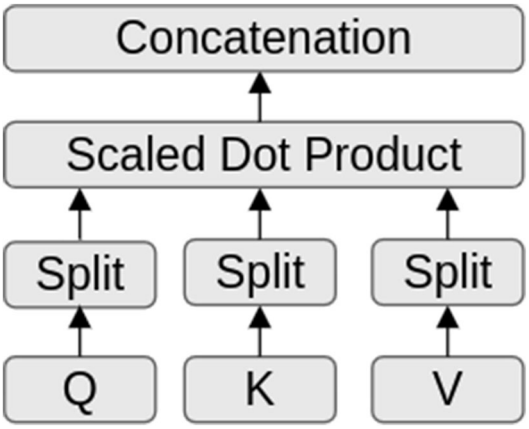

Fig. 3 Multihead self-attention

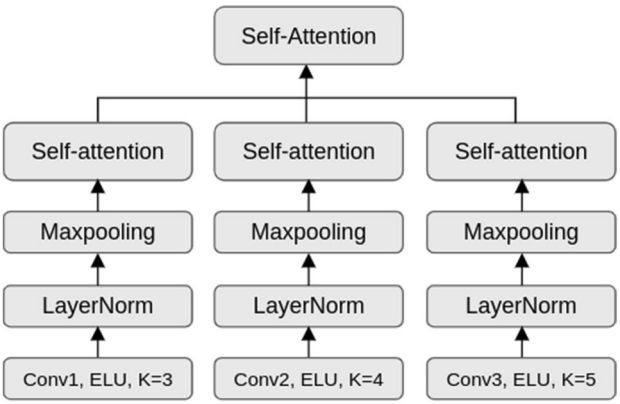

Fig. 4 Convolutional multihead self-attention block

which computes Query (Q), Key (K), and Value (V) vectors for each entry position using either linear projections or fully-connected layers.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{3}$$

where Q, K, and V can all be substituted by the same sequence of word representations. As shown in Fig. 2, Q and K are first multiplied to create a weight matrix QK^T, and scaled by a factor of $\sqrt{d_k}$. Then, the attention weights between Q and K can be computed by using the Softmax function. Finally, the attention weights are multiplied by V to produce a new output representation.

3.4 Multihead self-attention

Since the same attention weights are applied across all the d_k dimensions of the V vector, multihead self-attention [39] is employed to expand the capabilities of the self-attention module. It utilizes h parallel self-attentions and each head attends to a different portion of the dimension of the embedding as shown in Fig 3. The attention weights can be learned from different parts of the sequence and lead to more expressive output representation, rather than relying on only a single attention function to obtain the attention weights across all the embeddings in one single pass.

$$\begin{aligned} Multihead(Q, K, V) &= [head_1, ..., head_i, ..., head_h] \\ where\ head_i &= Attention(Q_i, K_i, V_i), i \in [1, h]. \end{aligned} \tag{4}$$

Specifically, multihead attention performs self-attention h times on the Q, K, and V vectors with each head splitting the original sequence dimension into d_k/h. Therefore, each head performs self-attention to unique vectors and then yields an output sequence with the same dimension d_k/h. The output of each head is then concatenated to the original dimension d_k.

3.5 Convolutional multihead self-attention block

Gao et al. proposed a convolutional multihead self-attention network enabling the self-attention network to obtain expressive representations from Q, K and V vectors [31]. In T-HMAN, we extend their work by leveraging self-attention layers to enhance the ability of feature extraction from convolved features after maxpooling as shown in Fig 4.

In the convolutional layers, three different window sizes $k = [3, 4, 5]$ are adopted to obtain n-grams features from source sequences, and then followed by Layer Normalization (LayerNorm) [56]. Jiang et al. suggested that attaching normalization to the output of encoders could significantly improve model performance [40]. In this work, LayerNorm,

which aims to normalize the activities of neurons for faster training convergence, is stacked following each convolutional layer. Then the max pooling is attached to obtain the most salient feature representations.

$$c^k = ELU(Conv1D(EW^k + b^k)), \tag{5}$$

$$Q, K, V = Maxpooling(LayerNorm(c^k)). \tag{6}$$

where E denotes input representations, W^k are filters with window sizes $k = [3, 4, 5]$, b^k represents a bias term, c^k is the output feature map of k window size. We employ exponential linear unit (ELU) [57] as the non-linear function after convolutional layers (Conv1D). By producing negative values, ELU could capture complex interactions between Q and K vectors, and thus produce expressive feature representations. Finally, the multihead self-attention is applied to obtain contextual information from the feature representations of the Maxpooling layers.

3.6 Soft attention and co-attention

Soft attention is applied at the end of the sentence encoder as well as the document encoder. As shown in Eq. (7) and (8), soft attention is adopted to normalize attention weights, thus measuring the importance of each entry through Softmax by computing a context vector for overall feature representations. The final output representations f of soft attention can be computed by a weighted sum over the normalized weights as in Eq. (9).

$$u = tanh(W_s c_e + b_s), \tag{7}$$

$$\alpha = \frac{exp(u^T W_{context})}{\sum_i exp(u^T W_{context})}, \tag{8}$$

$$f = \sum_i \alpha c_e. \tag{9}$$

where W_s and b_s are the weights and bias respectively, $c_e \in \mathbb{R}^{n \times d_k}$ denotes the output of the multihead self-attention block, $tanh$ is the non-linear function, u stands for the hidden state of the input sequence, $W_{context} \in \mathbb{R}^{d_k}$ represents the context vector, α is the normalized weights which indicates the importance of each entry in the sequence, f is the representations output of soft attention.

We extend document-level representations by utilizing co-attention in the document encoder. Given the sentence representations before the document encoder $se \in \mathbb{R}^{m \times d_k}$ and sentence representations after the document encoder $ce \in \mathbb{R}^{m \times d_k}$, co-attention attends to both the sentence representations interactively. Firstly, the affinity matrix $L \in \mathbb{R}^{d_k \times d_k}$ is computed as shown in Eq. (10), which is considered as a feature to predict the attention maps as in Eq. (11). Similar to soft attention, the attention weights of the two sentence representations are computed using the Softmax function, and the output vectors from co-attention are calculated by the weighted sum over both of the sentence features as in Eq. (12) and (13).

$$L = tanh(ce^T W_l se), \tag{10}$$

$$\begin{aligned} H^s &= tanh(W_s se + (W_d ce)L), \\ H^d &= tanh(W_d ce + (W_s se)L^T), \end{aligned} \tag{11}$$

$$\begin{aligned} \alpha^s &= Softmax(W_{hs}^T H^s), \\ \alpha^d &= Softmax(W_{hd}^T H^d), \end{aligned} \tag{12}$$

$$\begin{aligned} \bar{s} &= \sum_m \alpha^s se^m, \\ \bar{c} &= \sum_m \alpha^d ce^m. \end{aligned} \tag{13}$$

where $W_l \in \mathbb{R}^{d_k \times d_k}$ denotes the weight matrix to be learned through the network, $W_s, W_d \in \mathbb{R}^{p \times d_k}$ and $W_{hs}, W_{hd} \in \mathbb{R}^{1 \times p}$ are the weight parameters, p stands for the hidden size of co-attention. $\alpha^s, \alpha^d \in \mathbb{R}^{1 \times m}$ are the attention weights. $\bar{s}, \bar{c} \in \mathbb{R}^{1 \times d_k}$ are the learned features for both of the sentence representations produced by co-attention. We then concatenate $\bar{s}$ with $\bar{c}$ and pass it to soft attention in the document encoder.

Finally, the output of soft attention d_e, as well as the document-topic distributions d_t are concatenated and fed into the final Softmax function for classification. The overall model structure is shown in Fig 5.

4 Experiment

In this section, document classifications on five publicly accessible datasets are conducted to demonstrate the effectiveness of the proposed T-HMAN model. First, the classification datasets and their statistics are introduced. Then, the baseline models are discussed and analyzed. Finally, the implementation details are reported in detail.

4.1 Datasets

To explore the capacity of T-HMAN to encode contextual representations, we evaluate the method on a variety of tasks and datasets, including sentiment analysis, bias detection, and topic classification. All the datasets are obtained using the Huggingface Datasets Library[1]. Table 1

[1] https://github.com/huggingface/datasets.

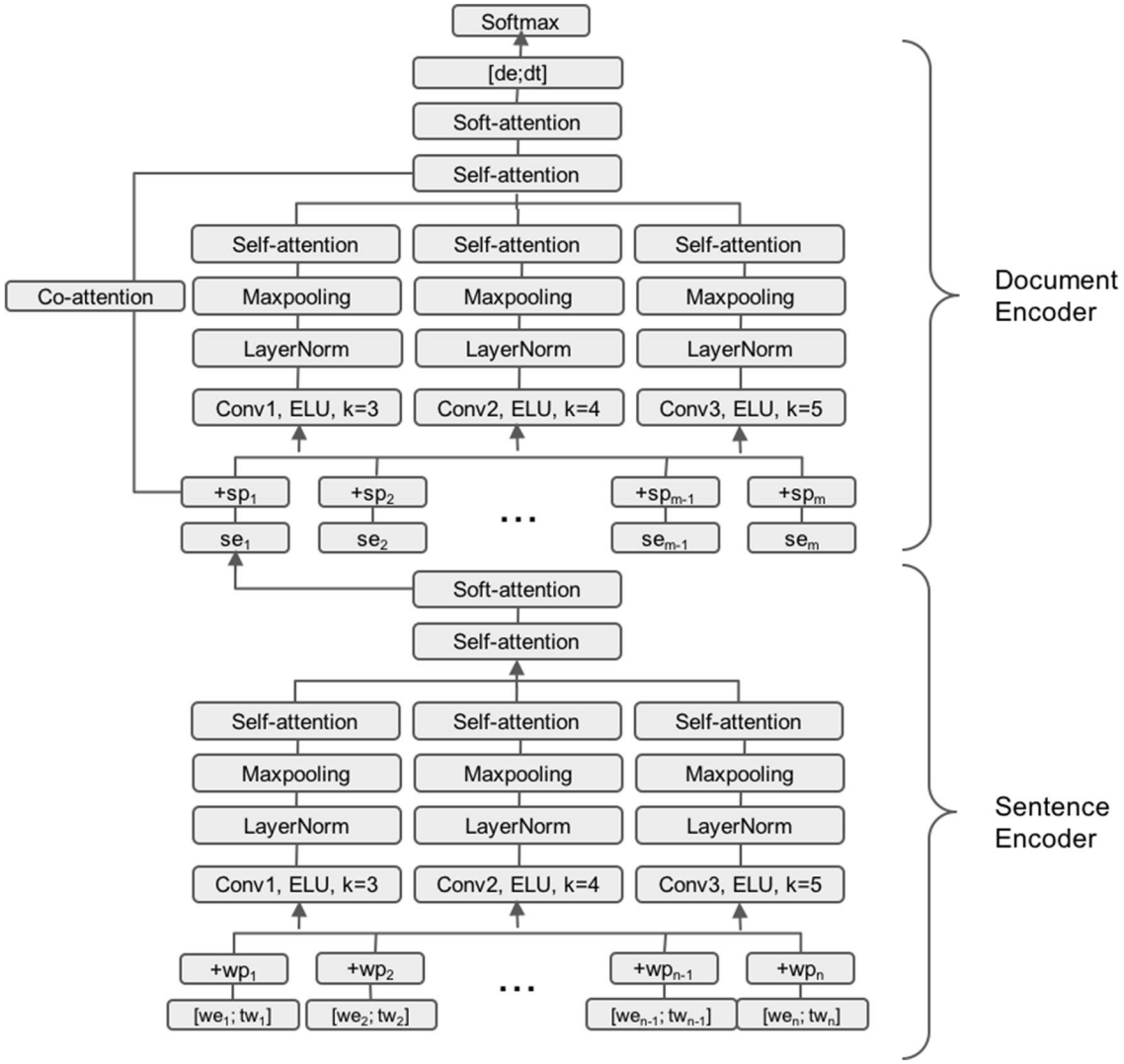

Fig. 5 The structure of T-HMAN. *we* denotes word embeddings, *tw* denotes topic-word distributions, *wp* is word position embeddings, *se* stands for sentence representations before self-attention block in document encoder,*sp* represents sentence position embeddings, *de* denotes document representations, *dt* is document-topic distributions

Table 1 Data Statistics: #s denotes the number of sentences (average(Avg) and maximum per document), #w stands for the number of words (average(Avg) and maximum per document)

Dataset	Classes	Documents	Vocabulary	Avg #s	Max #s	Avg #w	Max #w	Task
IMDB	2	50,000	108,757	13.2	150	248.4	2642	SA
Yelp 2015	5	700,000	259,171	10.1	133	147.9	1151	SA
Amazon Reviews	2	4,000,000	1,090,418	5.1	100	92.3	595	SA
Hyperpartisan News	2	645	25,505	41.1	665	746.6	6776	BD
AG News	4	127,600	67,952	1.9	32	41.2	201	TC

SA represents sentiment analysis, BD means bias detection, and TC denotes topic classification

summarizes the statistics of the datasets. A brief description of the datasets is given below:

IMDB: This is a movie review dataset for binary sentiment classification. It is composed of 50,000 movie

reviews, and the task is to predict whether a review is positive or negative.

Yelp 2015: The Yelp 2015 dataset consists of reviews from Yelp[2]. It is extracted from the Yelp Dataset Challenge 2015 with 700,000 reviews, and the labels correspond to customer satisfaction ratings ranging from 1 to 5.

Amazon Reviews: The dataset is constituted by reviews from the Amazon website. The data span over a period of 18 years, including 4,000,000 reviews up to March 2013. The task is to predict whether a review is positive or negative.

Hyperpartisan News: Hyperpartisan News Detection [58] is a dataset created for Semantic Evaluation (SemEval) 2019 Task 4[3] . Given a news article, the task is to determine whether or not it follows a hyperpartisan argumentation, i.e., whether it exhibits blind, prejudiced, or unreasoning allegiance to one party, faction, cause, or person. There are two parts of the dataset: **by-article** set is labeled through crowdsourcing. This part of the dataset contains 645 articles, for which a consensus among the crowdsourcing workers existed. **by-publisher** set is labeled by BuzzFeed journalists or MediaBiasFactCheck. In our experiment, we use the by-article set only because this is the official ranking set for the evaluation of SemEval 2019 Task 4.

AG News: AG is a collection of news articles gathered from more than 2,000 news sources. This dataset contains 127,600 news articles with four categories: World, Sports, Business and Sci/Tech.

We use 80% of the data for training, 10% for validation, and the remaining 10% for testing, we run each experiment three times to evaluate the model variability. During data preprocessing, all the characters are converted to lowercase. For non-hierarchical models, each document is truncated or padded to maximum 200 words. For hierarchical models, each document is split into 15 sentences based on periods, exclamation marks, and question marks (other punctuations are removed). Each sentence is truncated or padded to maximum 50 words. Fasttext [59] embeddings are trained using a minimum word frequency of five and the embeddings dimension size of 300.

4.2 Baselines

We compare T-HMAN with several baselines including non-hierarchical models Word-CNN and Attentive-RNN, as well as hierarchical models HAN, HCAN and HAHNN.

Word-CNN [17]: This model employs three parallel convolutional layers with window sizes 3,4 and 5, all of which contain 64 filters. Maxpooling layers are stacked after the convolutions, and the pooled feature maps coming from the varying window sizes are then concatenated. Dropout is applied with the rate of 0.3 on the concatenated vector, and the output is fed to a Softmax classifier.

Att-RNN [21]: Attentive-RNN consists a bi-directional LSTM with 50 hidden units and a dropout of probability 0.2 in each direction. In addition, the attention layer has 100 hidden units for the outputs from the bi-directional LSTM, which is then followed by a fully connected layer with 32 hidden units and the non-linearity ReLU, and then fed into a Softmax classifier.

HAN [60]: The same optimized hyperparameters are applied as mentioned in the original paper, where each level is composed of a bi-directional GRU with 50 units and soft attention with a hidden layer of 100 neurons.

HCAN [31]: The same optimized hyperparameters described in their paper are adopted. Each convolutional layer uses 512 filters with a fixed window size of 3. The convolutional multihead self-attention and target attention block are split into 8 heads. LayerNorm is applied after the elementwise multiplication of the two self-attention blocks.

HAHNN [32]: HAHNN consists of three parallel convolutional layers with window sizes 3, 4 and 5, each of which contains 64 filters. The generated feature maps are max pooled and concatenated, and then fed into a bidirectional GRU with 50 units. The soft attention layer contains 100 units and takes the outputs from the GRU to form sentence-level and document-level representations.

4.3 Implementation details

In the experiments, the coherence model is applied to find the optimal number of topics of LDA in the proposed network in a grid search strategy, and it gets the highest coherence score when the number of topics is set to 425. For both the sentence and document encoders, three parallel convolutional layers with window sizes 3, 4 and 5 are adopted, and each of them has 128 filter. The effect of number of self-attention heads on the model performance is evaluated. Among [1, 4, 8, 16] heads, the model yields the best accuracy when the number is 8. The dropout layer with the rate of 0.1 is utilized in the normalized attention weights, the word embeddings and the sentence representations. The proposed model is trained with the Adam optimizer [61] of learning rate 1e-3, beta1 0.9, and beta2 0.999. Early stopping is leveraged to save the model weights with the highest validation accuracy. The hyperparameters of T-HMAN are tuned on the validation set, and the best model is then used to evaluate the test set.

[2] https://www.yelp.com/dataset

[3] https://pan.webis.de/semeval19/semeval19-web/

Table 2 Test data accuracy and standard deviations

	IMDB	Yelp	Amazon Reviews	Hyperpartisan News	AG News
Word-CNN	0.8832±0.0025	0.6071±0.0042	0.8996±0.0069	0.7547±0.0025	0.8933±0.0080
Att-RNN	0.8789±0.0057	0.6015±0.0072	0.9021±0.0074	0.7601±0.0038	0.8878±0.0092
HAN	0.9109±0.0081	0.6381±0.0790	0.9245±0.0096	0.7656±0.0031	0.9248±0.0081
HAHNN	0.9024±0.0019	0.6471±0.0042	0.9268±0.0038	0.7700±0.0011	0.9257±0.0094
HCAN	0.9059±0.0044	0.6489±0.0051	0.9313±0.0065	0.7717±0.0031	0.9389±0.0073
T-HMAN	**0.9136**±0.0105	**0.6607**±0.0092	**0.9420**±0.0111	**0.7851**±0.0108	**0.9477**±0.0147

The Best accuracies are in bold

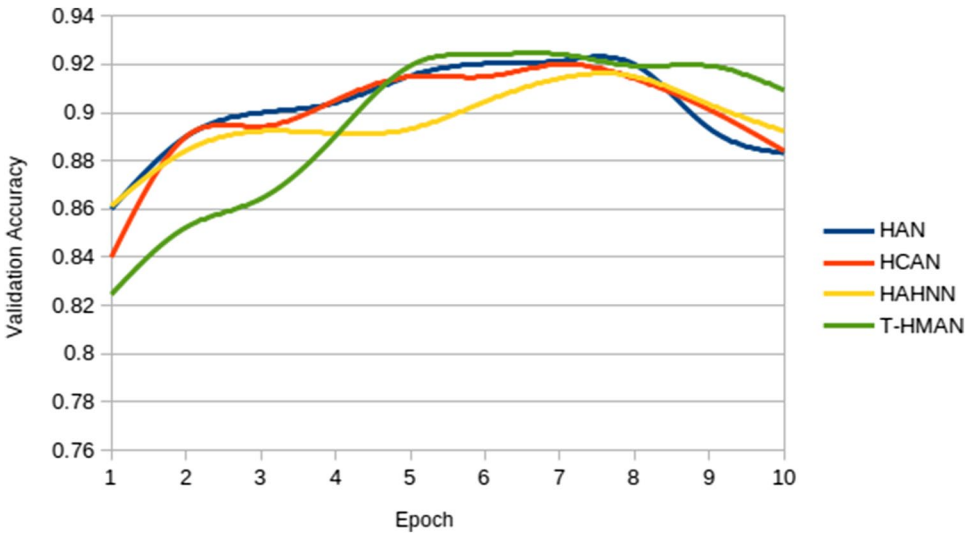

Fig. 6 Validation accuracy in the first 10 epochs on IMDB dataset

5 Results and discussion

The experimental results are shown in Table 2. T-HMAN achieves the highest test accuracy throughout the five public datasets.

5.1 Comparsion with non-hierarchical models

T-HMAN outperforms Word-CNN (Att-RNN) by 3.04% (2.5%) on small datasets IMDB (Hyperpartisan News). This improvement is even enhanced when the datasets are larger, i.e. 5.36%, 3.99%, and 5.44% on Yelp 2015, Amazon Reviews, and AG News, respectively. The result confirms that taking account of document structural information, such as the relationships between words and sentences, and between sentences and documents, could significantly improve the feature representations and thus yield better performance in text classification. Meanwhile, the sequence length in the non-hierarchical models is constrained to maximum of 200 words. However, the hierarchical models can take longer sequence lengths, e.g. words per sentence 50 multiplied by the number of sentences 15 resulting in a maximum of 750 words per document. The longer input allows models to encode more expressive feature representation from them and take longer semantic dependencies into account.

5.2 Comparsion with hierarchical models

Exploring the hierarchical models, T-HMAN outperforms the second best model HAN by 0.27% on IMDB, and HCAN by 1.18%, 1.07%, 1.34%, and 0.88% on Yelp 2015, Amazon Reviews, Hyperpartisan News and AG News respectively. It is worth noting that T-HMAN generally yields higher accuracy than HCAN on datasets that have a higher average number of words per document such as Yelp 2015 and Hyperpartisan News. This confirms that the multiple attention mechanisms of the proposed model are able to capture longer semantic dependency.

Figure 6 demonstrates the validation accuracy of all the hierarchical models on the IMDB dataset over the first 10 epochs. T-HMAN reaches its peak faster than the other hierarchical models. Meanwhile, the confusion matrices (as shown in Fig.7) demonstrate the performance of the hierarchical models in the multi-class classification task. Specifically, the most mismatches in the AG News dataset are

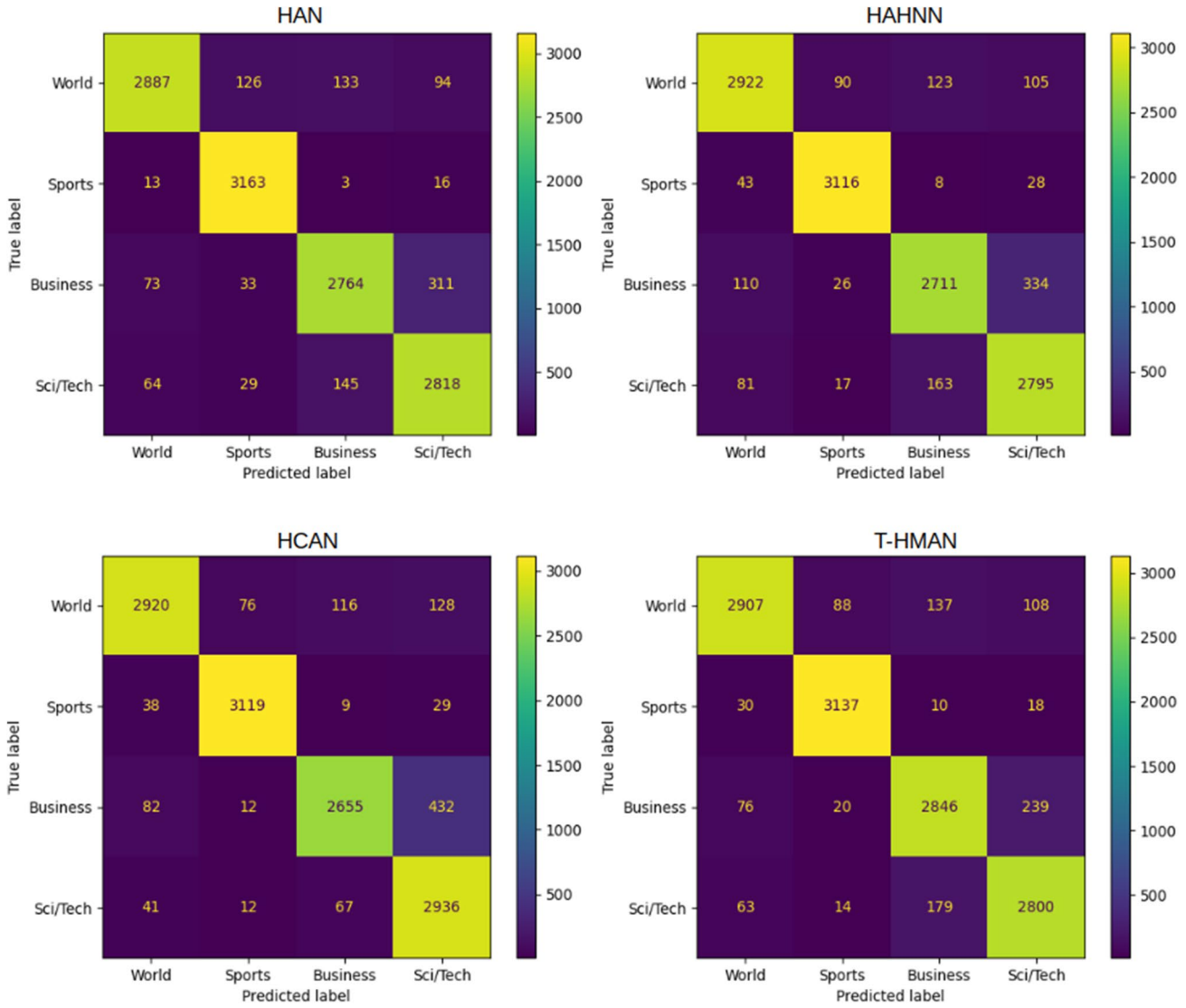

Fig. 7 Confusion matrices for the hierarchical models on AG news dataset

between Business and Sci/Tech. Although HCAN achieves the lowest number of mismatches between Business (Predicted) and Sci/Tech (True label), it has the highest mismatches (i.e. 432) between Business (True label) and Sci/Tech (Predicted). The average of the above two mismatch pairs are calculated for all the four hierarchical models. The proposed T-HMAN achieves the least averaged mismatches (i.e. 209) compared with HCAN (249.5), HAHNN (248.5), and HAN (228).

The improvement is mainly attributed to the fact that T-HMAN leverages multiple attention mechanisms as well as topic distributions, enabling it to capture more expressive sentence-level and document-level representations. The convolutional multihead self-attention blocks combine multiple window sizes stacked with max pooling to extract the most salient information from input sequences, rather than taking the entire convolved sequences as input to self-attention as in HCAN.

In terms of co-attention, it expands document representations by learning sentence representations before and after the self-attention block in the document encoder. Before the self-attention block, the sentence representations are only formed by the sentence encoder. While after self-attention block, it involves sentence-level attention weights that reflect how each sentence contributes to the final prediction. Then, the soft attention module summarizes the importance of sentence-level and document-level representations by assigning attention weights to the output sequences of the self-attention block.

The topic-word distributions contain global word co-occurrence information shared between topics, and the document-topic distributions that have local (per-document) topic

Table 3 Test set accuracy with standard deviation

Methods	**IMDB**	**Yelp 2015**	**Amazon Reviews**	**Hyperpartisan News**	**AG News**
Basic + AVE	0.8957±0.0028	0.6268±0.0097	0.9215±0.0019	0.7701±0.0031	0.9261±0.0016
Basic + CA	0.8991±0.0079	0.6290±0.0045	0.9239±0.0038	0.7715±0.0051	0.9267±0.0072
Basic + CA + SA	0.9035±0.0086	0.6344±0.0390	0.9307±0.0036	0.7749±0.0053	0.9386±0.0090
Basic + CA + SA + *tw*	0.9040±0.0121	0.6375±0.0098	0.9335±0.0077	0.7757±0.0180	0.9389±0.0105
Basic + CA + SA + *dt*	0.9097±0.0123	0.6451±0.0071	0.9357±0.0103	0.7768±0.0083	0.9423±0.0089
All (T-HMAN)	0.9136±0.0105	0.6607±0.0092	0.9420±0.0111	0.7851±0.0108	0.9477±0.0147

AVE denotes taking the average of the output sequence from the self-attention block to form feature representations, CA is co-attention, SA represents soft attention, *tw* and *dt* stand for topic-word and document-topic distributions respectively

probabilities are independent of other documents. These distributions enrich feature representations when concatenated with word embeddings and document-level representations. However, the topic distributions also bring variability to the model performance, i.e. the standard deviations of the proposed model are generally higher than other models on the five datasets. This is because the topic distributions are pre-trained only with the topic model, and T-HMAN does not update the pre-trained topic distributions after it is concatenated with input representations. The ablation study also confirms that models without topic distributions yield smaller standard deviations.

5.3 Ablation study

Table 3 demonstrates how co-attention, soft attention, and topic distributions affect the model performance. Since position embeddings have been investigated in the previous studies [26, 31, 39] and proved effective in improving model performance, the absolute position embeddings are combined to the word/sentence embeddings by default throughout our experiments. Firstly, a basic model called Basic + AVE is built, in which co-attention and topic distributions are removed, and soft attention is replaced by directly taking the average of the output sequence of each multihead self-attention block. Secondly, co-attention is then added back to the document encoder and the model is referred to as Basic + CA. Thirdly, soft attention is employed in place of mean operation at the end of both encoders, and this model is denoted as Basic + CA + SA. Finally, topic-word and document-topic distributions are evaluated respectively, i.e. Basic + CA + SA + *tw* and Basic + CA + SA + *dt*.

The results in Table 3 prove the superiority of combing the components over the simple models. Compared with Basic + AVE, the model with attention mechanisms (i.e., Basic + CA + SA) improves the accuracy by 0.83% on average across the five datasets. The effectiveness of each topic distribution (i.e. topic-word distributions and document-topic distributions) is evaluated separately. In comparison to Basic + CA + SA, the accuracy of the model coupled with both topic distributions (i.e., T-HMAN) is improved by 1.30% on average across the five datasets. The performance improvement can be attributed to more expressive feature representations when the word-level (document-level) inputs are augmented with topic-word (document-topic) distributions. However, the concatenation between the input representation and the topic distributions leads to higher standard deviations throughout the five datasets. This is potentially because the pre-trained topic distributions do not update during the training phases in the proposed model.

6 Conclusion and future work

In this work, a topic-aware hierarchical multi-attention model for text classification named T-HMAN is proposed. The multi-head self-attention armed with convolutional layers is developed to parallelize feature extraction on long-range semantic dependencies. In order to strengthen its power for aggregating feature representations and extending document-level feature space, soft attention and co-attention are incorporated, respectively. For further enriching feature representations at both the sentence-level and document-level, the topic distributions generated by the LDA model are employed. By evaluating T-HMAN on the five publicly accessible datasets, we demonstrate that our approach achieves state-of-the-art classification accuracy and converges faster than other hierarchical models. The contributions of each component of T-HMAN is analyzed in the ablation study. The result suggests that the combination of multiple attention mechanisms and topic distributions can significantly improve model performance in text classification tasks. One of the main limitations of the proposed model is that the topic distributions are pre-trained with the LDA topic model, in which the distributions are not updated with the hierarchical contextual feature representations. In the future work, the topic distributions will be parameterized and updated simultaneously with the model parameters.

References

1. Rubin V, Conroy N, Chen Y, Cornwell S (2016) Fake news or truth? using satirical cues to detect potentially misleading news. In: Proceedings of the Second Workshop on Computational Approaches to Deception Detection, pp 7–17
2. Zhao R, Mao K (2018) Fuzzy bag-of-words model for document representation. IEEE Trans Fuzzy Syst 26(2):794–804
3. Fortuna B, Galleguillos C, Cristianini N (2009) Detection of bias in media outlets with statistical learning methods. In: Text Mining, pp 57–80. Chapman and Hall/CRC
4. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
5. Lin C, Ibeke E, Wyner A, Guerin F (2015) Sentiment-topic modeling in text mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5(5):246–254
6. Ibeke E, Lin C, Wyner A, Barawi MH (2017) Extracting and understanding contrastive opinion through topic relevant sentences. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp 395–400
7. Li Z, Shang W, Yan M (2016) News text classification model based on topic model. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp 1–5 . IEEE
8. Steinberger J, Křišťan M (2007) Lsa-based multi-document summarization. In: Proceedings of 8th International PhD Workshop on Systems and Control, vol. 7
9. Hosseinalipour A, Gharehchopogh FS, Masdari M, Khademi A (2021) Toward text psychology analysis using social spider optimization algorithm. Concurr Comput Pract Exp 33(17):6325
10. Lu Y, Mei Q, Zhai C (2011) Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. Inf Retrieval 14(2):178–203
11. Khataei Maragheh H, Gharehchopogh FS, Majidzadeh K, Sangar AB (2022) A new hybrid based on long short-term memory network with spotted hyena optimization algorithm for multi-label text classification. Mathematics 10(3):488
12. Jiang Y, Song X, Harrison J, Quegan S, Maynard D (2017) Comparing attitudes to climate change in the media using sentiment analysis based on latent dirichlet allocation. In: Proceedings of the 2017 EMNLP Workshop: Natural Language Processing Meets Journalism, pp 25–30
13. Keller M, Bengio S (2004) Theme topic mixture model: A graphical model for document representation. In: PASCAL Workshop on Text Mining and Understanding
14. Zheng J, Cai F, Chen W, Feng C, Chen H (2019) Hierarchical neural representation for document classification. Cognit Comput 11(2):317–327
15. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Won K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. Ijcai
16. Wei W, Zhang X, Liu X, Chen W, Wang T (2016) pkudblab at semeval-2016 task 6 : A specific convolutional neural network system for effective stance detection. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). https://doi.org/10.18653/v1/s16-1062
17. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882
18. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Thirtieth AAAI Conference on Artificial Intelligence
19. Wang Y, Liu J, Jiang Y, Erdélyi R (2019) Cme arrival time prediction using convolutional neural network. Astrophys J 881(1):15
20. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) 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
21. Lin Z, Feng M, Santos CNd, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130
22. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint
23. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473
24. Xu S, Li H, Yuan P, Wu Y, He X, Zhou B (2020) Self-attention guided copy mechanism for abstractive summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 1355–1362
25. Shen T, Zhou T, Long G, Jiang J, Pan S, Zhang C (2018) Disan: Directional self-attention network for rnn/cnn-free language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32
26. Ambartsoumian A, Popowich F (2018) Self-attention: A better building block for sentiment analysis neural network classifiers. arXiv preprint arXiv:1812.07860
27. Dosovitskiy A, Beyer L, Kolesnikov, A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
28. Lu J, Yang J, Batra D, Parikh D (2016) Hierarchical question-image co-attention for visual question answering. arXiv preprint arXiv:1606.00061
29. Yin W, Schütze H (2016) Multichannel variable-size convolution for sentence classification. arXiv preprint arXiv:1603.04513
30. Conneau A, Schwenk H, Barrault L, Lecun Y (2017) Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 1107–1116. Association for Computational Linguistics, Valencia, Spain (2017). https://www.aclweb.org/anthology/E17-1104
31. Gao S, Ramanathan A, Tourassi G (2018) Hierarchical convolutional attention networks for text classification. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2018)
32. Abreu J, Fred L, Macêdo D, Zanchettin C (2019) Hierarchical attentional hybrid neural networks for document classification. arXiv preprint arXiv:1901.06610 (2019)
33. Ruchansky N, Seo S, Liu Y (2017) Csi: A hybrid deep model for fake news detection. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM 17 . https://doi.org/10.1145/3132847.3132877
34. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015). Show, attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning, pp 2048–2057. PMLR
35. Cheng J, Dong L, Lapata M (2016) Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733
36. Kokkinos F, Potamianos A (2017) Structural attention neural networks for improved sentiment analysis. arXiv preprint arXiv:1701.01811
37. Daniluk M, Rocktäschel T, Welbl J, Riedel S (2017) Frustratingly short attention spans in neural language modeling. arXiv preprint arXiv:1702.04521
38. Zhou Y, Zhou J, Liu L, Feng J, Peng H, Zheng X (2018) Rnn-based sequence-preserved attention for dependency parsing. In:

Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Lukasz Kaiser Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems
40. Jiang Y, Petrak J, Song X, Bontcheva K, Maynard D (2019) Team Bertha von Suttner at SemEval-2019 Task 4: Hyperpartisan News Detection using ELMo Sentence Representation Convolutional Network. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 840–844
41. Shu K, Cui L, Wang S, Lee D, Liu H (2019) defend: Explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 395–405
42. Tian B, Zhang Y, Wang J, Xing C (2019) Hierarchical inter-attention network for document classification with multi-task learning. In: IJCAI, pp 3569–3575
43. Liu T, Hu Y, Wang B, Sun Y, Gao J, Yin B (2022) Hierarchical graph convolutional networks for structured long document classification. IEEE Transactions on Neural Networks and Learning Systems
44. Li J, Wang C, Fang X, Yu K, Zhao J, Wu X, Gong J (2022) Multi-label text classification via hierarchical transformer-cnn. In: 2022 14th International Conference on Machine Learning and Computing (ICMLC), pp 120–125
45. Ibeke E, Lin C, Wyner A, Barawi MH (2020) A unified latent variable model for contrastive opinion mining. Front Comput Sci 14(2):404–416. https://doi.org/10.1007/s11704-018-7073-5
46. Lin C, Ibeke E, Wyner A, Guerin F (2015) Sentiment-topic modeling in text mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5(5):246–254. https://doi.org/10.1002/widm.1161
47. Wu X, Fang L, Wang P, Yu N (2015) Performance of using LDA for Chinese news text classification. In: 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), pp 1260–1264 . IEEE
48. Kim D, Seo D, Cho S, Kang P (2019) Multi-co-training for document classification using various document representations: Tf-idf, lda, and doc2vec. Inf Sci 477:15–29
49. Lin C, He Y (2009) Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp 375–384
50. Liu Y, Liu Z, Chua T-S, Sun M (2015) Topical word embeddings. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
51. Xu H, Dong M, Zhu D, Kotov A, Carcone AI, Naar-King S (2016) Text classification with topic-based word embedding and convolutional neural networks. In: Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, pp 88–97. ACM
52. Wang Y, Xu W (2018) Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. Decis Support Syst 105:87–95
53. Narayan S, Cohen SB, Lapata M (2018) Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. arXiv preprint arXiv:1808.08745
54. Jiang Y, Wang Y, Maynard XSD (2020) Comparing topic-aware neural networks for bias detection of news. In: Proceedings of 24th European Conference on Artificial Intelligence (ECAI 2020). International Joint Conferences on Artificial Intelligence (IJCAI)
55. Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN (2017) Convolutional sequence to sequence learning. In: International Conference on Machine Learning, pp 1243–1252. PMLR
56. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450
57. Clevert D-A, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289
58. Kiesel J, Mestre M, Shukla R, Vincent E, Adineh P, Corney D, Stein B, Potthast M (2019) Semeval-2019 task 4: Hyperpartisan news detection. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp 829–839
59. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
60. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies
61. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980