

Contrastive Document Representation Learning with Graph Attention Networks

Peng Xu Xinchu Chen Xiaofei Ma Zhiheng Huang Bing Xiang
{pengx, xcc, xiaofei, zhiheng, bxiang}@amazon.com

AWS AI Labs

Abstract

Recent progress in pretrained Transformer-based language models has shown great success in learning contextual representation of text. However, due to the quadratic self-attention complexity, most of the pretrained Transformers models can only handle relatively short text. It is still a challenge when it comes to modeling very long documents. In this work, we propose to use a graph attention network on top of the available pretrained Transformers model to learn document embeddings. This graph attention network allows us to leverage the high-level semantic structure of the document. In addition, based on our graph document model, we design a simple contrastive learning strategy to pretrain our models on a large amount of unlabeled corpus. Empirically, we demonstrate the effectiveness of our approaches in document classification and document retrieval tasks.

1 Introduction

Document representations that capture the semantics are crucial to various document-level Natural Language Processing (NLP) tasks, including sentiment analysis (Medhat et al., 2014), text classification (Kowsari et al., 2019) and information retrieval (Lin et al., 2020). In recent years, an increasing volume of work has focused on learning a task-agnostic universal representation for long documents. While improved performance in downstream tasks have been achieved, there are two challenges towards learning a high quality document representation: (1) **absence of document structure**. Most works treat the document as a sequence of tokens without considering high-level structure. (2) **data scarcity**. Existing methods in document representation learning are significantly affected by the scarcity of document-level data.

Transformers-based pretrained language models are ubiquitously state-of-the-art across many NLP tasks. Transformer models such as BERT

(Devlin et al., 2019) and its variants have shown great success in learning contextual representation of text. Representation from large language models can partially mitigate the data scarcity issue due to pretraining on a large amounts of unlabeled data. However, those models mostly consider token-level information and their pretraining tasks are not directly targeting long document representations. Another issue of directly applying transformer-based models is the limit of the input text length. Due to the quadratic complexity of self-attention, most of the pretrained transformers models can only handle a relatively short text. A wide spectrum of efficient, fast transformer models (collectively called “X-formers”) have been proposed to tackle this problem; e.g., Longformer (Beltagy et al., 2020) and Bigbird (Zaheer et al., 2020) use sparse attention to improve the computational and memory efficiency for long sequence text. Nevertheless, these models still focus on token-level interactions without considering high-level semantic structure of the document.

Recently, there is a resurgence of interest in Contrastive Learning (CL) due to its success in self-supervised representation learning in computer vision (Chen et al., 2020; He et al., 2020). Contrastive Learning offers a simple method to learn disentangled representation that encodes invariance to small and local changes in the input data without using any labeled data. In NLP domain, contrastive learning has been employed to learn sentence representation (Wu et al., 2020; Qu et al., 2020) under either self-supervised or supervised settings.

In this work, we propose a Graph Attention Network (GAT) based model that explicitly utilizes the high-level semantic structure of the documents to learn document embeddings. We model the document as not just a sequence of text, but a collection of passages or sentences. Specifically, the proposed model introduces a graph on top of the document passages (Fig. 1) to utilize multi-granularity

information. First, passages are encoded using RoBERTa (Liu et al., 2019) to collect word-level knowledge. Then passages are connected to leverage the higher-level structured information. At last, a graph attention network (Veličković et al., 2017) is applied to obtain the multi-granularity document representation. To better learn the document embedding, we propose a document-level contrastive learning strategy to pretrain our models. In our contrastive learning framework, we split the document into random sub-documents and train the model to maximize the agreement over the representations of the sub-documents that come from the same document. This simple strategy allows us to pretrain our models on a large unlabelled corpus without any additional priors. As we will see, this simple pretraining task indeed helps the model on the downstream tasks.

The contributions of this paper can be summarized as follows.

- We propose to a graph document model with graph attention networks that can not only explicitly utilize the high-level structure of the document but also leverage pretrained Transformer encoders to obtain low-level contextual information.
- We propose a simple document-level contrastive learning strategy, which does not require any handcrafted transformations and is suitable for large-scale pretraining.
- We conduct empirical evaluations on our models and contrastive pretraining strategy. We show that our graph-roberta models achieve great performance on both document classification and retrieval tasks. Specifically we demonstrate that our contrastive pretraining helps the model learn a meaningful document representation even without fine-tuning, and improve both the training convergence speed and final performance during end-to-end fine-tuning on downstream classification tasks. For document retrieval tasks, we demonstrate that our graph-roberta models have great semantic matching performance, compensating the typical lexical matching system.

2 Methodology

In this section, we describe our main model and contrastive pretraining strategy.

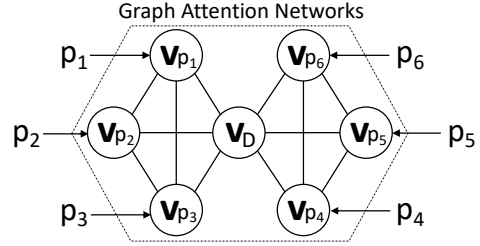


Figure 1: An example of the Graph-Roberta architecture for the document representation.

2.1 Graph Document Architecture

In this work, we model a document as graph over passages. Given a document D with passages $\{p_1, \dots, p_{|D|}\}$, we define an undirected graph $\mathcal{G} : (\mathcal{V}, \mathcal{E})$, where \mathcal{V} consists of $n + 1$ nodes $(v_D, v_{p_1}, \dots, v_{p_n})$ and the graph edges \mathcal{E} are constructed based on the document structure. An example of a document graph is shown in Fig. 1. Once the document graph is defined, we can instantiate a neural network model based on the graph structure.

Passage Node Initialization First, we use the state-of-the-art contextual language models to encode each passage text, since each passage is relatively short. Specifically, given a passage p_i consists of a sequence of words $\{w_{i,1}, w_{i,2}, \dots, w_{i,|p_i|}\}$, we use Roberta(Liu et al., 2019) as the encoder model for the passage node and project the [CLS] vector into fixed embedding space as the initial passage node representation.

$$\mathbf{v}_{p_i}^{(0)} = \tanh(\mathbf{w}\phi(w_{i,\cdot}) + \mathbf{b}), \quad (1)$$

where ϕ is RoBERTa with [CLS] vector.

Document Node Initialization For the document node, we simply use the average of all the passage node embeddings as the initial representation.

$$\mathbf{v}_D^{(0)} = \frac{1}{n} \sum_{i=1}^n \mathbf{v}_{p_i}^{(0)}. \quad (2)$$

Graph Attention Layers Finally, we apply T Graph Attention Layers (GAL)¹ to aggregate all the information from different nodes.

$$\mathbf{v}_i^{(t+1)} = \text{GAL}(\mathbf{v}_k^{(t)} | k \in \mathcal{N}(i)), \quad (3)$$

where $\mathcal{N}(i)$ is the neighbour node set of passage node p_i on the given graph structure. The step t counts from 1 to T and the final document node representation is $\mathbf{v}_D^{(T)}$.

¹Refer to Veličković et al. (2017) for the details of GAL.

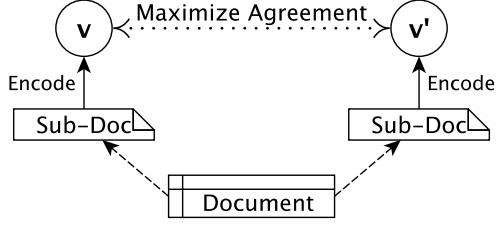


Figure 2: A simple contrastive learning strategy for document representation.

2.2 Contrastive Pretraining for Document Representation Learning

We design a simple contrastive learning task to pre-train our graph document models. The main idea follows from the contrastive learning framework in (Chen et al., 2020), where the task is to learn an encoder function to maximize the agreement between augmented views of the same image. Here we consider that any proportions inside the same document are the different “views”. The task is to maximize the agreement between different proportions that come from the same document. Since our document has been represented as a list of passages, a proportion of a document would be any subset of passages, which we call a *sub-document*. During training time, we randomly sample a mini-batch of N documents $\mathcal{D} = \{D_i\}_{i=1}^N$. For each document D_i , we randomly split passages to two subsets as sub-documents:

$$D_i \rightarrow \tilde{D}_i, \tilde{D}'_i, \quad (4)$$

where D_i is the union set of \tilde{D}_i and \tilde{D}'_i .

We treat \tilde{D}_i and \tilde{D}'_i as the positive pair. Any pair of sub-documents that come from different documents are negative pairs. Then the noise contrastive loss function for a positive pair is defined as

$$\begin{aligned} \ell(\tilde{D}_i) &= -\log \frac{\exp(\mathbf{v}_{\tilde{D}_i} \cdot \mathbf{v}_{\tilde{D}'_i})}{\sum_{j=1}^N \exp(\mathbf{v}_{\tilde{D}_i} \cdot \mathbf{v}_{\tilde{D}'_j})}, \\ \ell(\tilde{D}'_i) &= -\log \frac{\exp(\mathbf{v}_{\tilde{D}_i} \cdot \mathbf{v}_{\tilde{D}'_i})}{\sum_{j=1}^N \exp(\mathbf{v}_{\tilde{D}_j} \cdot \mathbf{v}_{\tilde{D}'_i})}, \end{aligned} \quad (5)$$

where $\mathbf{v}_{\tilde{D}_i}$ is the encoding of the sub-document \tilde{D}_i based on the proposed graph document model.

The final loss is computed across all the pairs.

$$\ell(\mathcal{D}) = \frac{1}{2N} \sum_{i=1}^N \ell(\tilde{D}_i) + \ell(\tilde{D}'_i). \quad (6)$$

3 Experiments

We experiment on two popular applications, text classification and document retrieval to evaluate the proposed approach. The experimental results show that the graph based document representation could capture long document information and the contrastive learning strategy could utilize unlabeled data to further improve the performance and training efficiency.

Model details Throughout the paper, we use roberta-*base* model (Liu et al., 2019) as our passage node encoder. On top of that, we add 2 graph attention layers with 2 heads and skip connections. Specifically, we utilize the Deep Graph Library² for GAT implementation. The embedding size for passage and document nodes is 512. We refer this model as graph-roberta model.

3.1 Datasets

In this section, we describe all the datasets we use in this paper.

OpenWebText (Gokaslan et al., 2019) is an open-source recreation of the Webtext corpus in Radford et al. (2019). The text was extracted from Reddit post urls, which produces around 8M documents.

arXiv (He et al., 2019) is a collection of 33,388 arXiv scientific papers from 11 categories. The average document length exceeds 5,000 words. We create a random train/dev/test split of 25,568/3,196/3,197.

NewsGroup (Lang, 1995) a collection of news-group documents, partitioned (nearly) evenly across 20 different newsgroups. It contains 11,314 training and 7,532 test samples. We sample 10% of the training data for validation.

IMDB (Maas et al., 2011) is a dataset for binary sentiment classification. It contains 25,000 labeled movie reviews as the training set and another 25,000 movie reviews as the test set. We random sample 1,000 examples from the training set for validation.

Hyperpartisan (Kiesel et al., 2019) is a binary classification dataset for hyperpartisan news detection. It consists of 645 documents in total. We use the same train/dev/test split (516/64/65) from (Beltagy et al., 2020).

Robust04 (Voorhees, 2005) is the news collection from the TREC 2004 Robust track. It is a doc-

²<https://github.com/dmlc/dgl>

ument retrieval dataset consisting of 249 queries with relevance labels on a corpus of 528K documents.

MSMARCO DR (Bajaj et al., 2016) is a document ranking dataset with about 3.2M documents. It provides over 367K training queries and an official dev set of 5,193 queries. The Trec 2019 Deep Learning track (Craswell et al., 2020) also provides an additional test set of 43 queries.

WIKIR³ (Frej et al., 2019) is an open-source toolkit to create large-scale information retrieval datasets based on Wikipedia. In this work, we use the English Wikipedia dump from 2020/12/20⁴ and following the the same settings in Frej et al. (2019) except for that we preserve the punctuation and section information in the document. We obtain two datasets, WIKIR62K and WIKIRS62K, both of which contain around 60k training queries, 1k dev queries and 1k test queries. The queries in WIKIR62K are built based on titles and the ones in WIKIRS62K are based on the first sentences. The processed document corpus size is around 2.4M documents.

Since graph-roberta models take a document input as a graph over passages, we split each document into passages with around 100 words while respecting the sentence boundary. For WIKIR documents, we also respect the section boundaries. Without additional specification, we use the fully-connected graph structure by default.

3.2 Document Classification

In this section, we conduct empirical evaluation of our models on document classification tasks. We consider 4 datasets, arXiv, Hyperpartisan, IMDB and Newsgroup. We compare our graph-roberta models with the baseline model Roberta (Liu et al., 2019), as well as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020), two state-of-the-art transformer models that handle long text input⁵. In our experiments, we only consider the *base* version of those models.

Contrastive Pretraining We pretrain our graph-roberta models on OpenWebText dataset. During the training process of contrastive learning, for each document, we keep up to 50 passages and we randomly select half the number of passages as the

³<https://github.com/getalp/wikIR>

⁴<https://archive.org/download/enwiki-20201220/enwiki-20201220-pages-articles-multistream.xml.bz2>

⁵We take the [CLS] embedding from those models as the document representation.

sub-document and the rest of the half as the other sub-document. We train for 10 epochs with batch size 1,536, using Adam (Kingma and Ba, 2014) optimizer with a learning rate 5e-5 and warm up rate 0.1.

Finetuning For graph-roberta models, we keep up to 50 passages per document during training and at inference time, we keep up to 100 passages per document. For the other models, we truncate the document text up to the maximum sequence length they are allowed to handle; Roberta’s maximum input length is 512, and Longformer and BigBird’s maximum input length is 4,096. The detailed training configurations are shown in appendix.

Clustering First, we evaluate the capability of our graph-roberta model as an off-the-shelf document encoder through document clustering. We take the document node representation with the pretrained graph-roberta model and [CLS] embeddings from the other three models. We run k-means clustering methods on the training set and run inference on the test set. We compute the normalized mutual information (NMI) and Purity to evaluate the clustering quality. We report the results on arXiv and Newsgroup dataset in Table 1. As we can see, our pretrained graph-roberta model clearly outperforms the other three models by a large margin. This is expected that the other three models are not pretrained on any document-level tasks. Fig. 3 & 4 showcase that the simple unsupervised contrastive learning strategy indeed helps graph-roberta model learn meaningful document representations.

model	arXiv		Newsgroup	
	NMI	Purity	NMI	Purity
Roberta	0.267	0.327	0.116	0.160
Longformer	0.168	0.241	0.084	0.133
Bigbird	0.180	0.261	0.059	0.119
Graph-roberta	0.437	0.558	0.516	0.475

Table 1: Clustering performance with different document embeddings on the test sets.

End-to-end Classification To evaluate the full capability of the graph-roberta model, we also conduct end-to-end finetuning on the 4 datasets. In addition to 4 pretrained models, we also report the performance of graph-roberta without contrastive learning. The results are shown in Table 2. First, we can see that graph-roberta model outperforms all the other methods on 3 out of 4 datasets. The exception is IMDB dataset, which has relatively

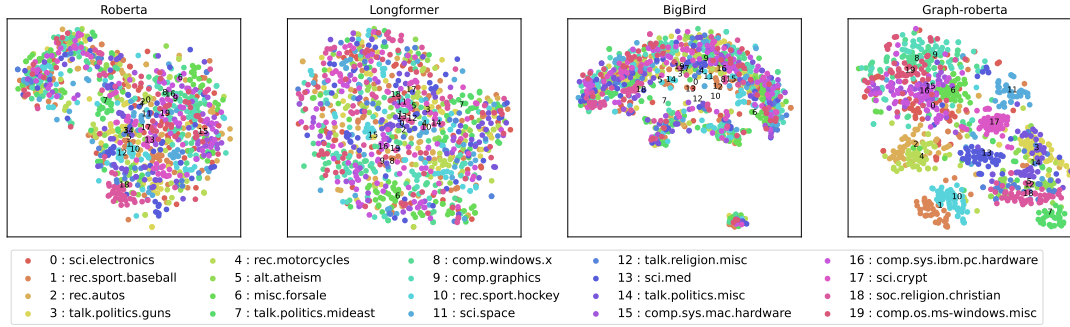


Figure 3: TSNE visualization of different representations on 1000 documents sampled from Newsgroup dataset.

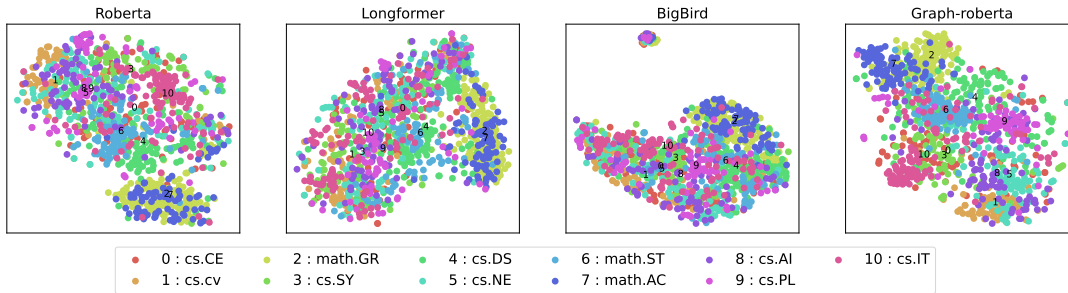


Figure 4: TSNE visualization of different representations on 1000 documents sampled from arXiv dataset.

Model	arXiv	Newsgroup	IMDB	Hyperpartisan	avg
Roberta	89.99	86.22	95.63	90.77	90.65
Longformer	90.90	86.42	95.77	92.31	91.35
BigBird	88.99	81.41	95.32	89.23	88.74
Graph-roberta w/o CL	86.83	85.36	94.32	94.12	90.16
Graph-roberta	91.21	86.66	94.26	96.15	92.07
⁶ Longformer (Beltagy et al., 2020)	-	-	95.70	94.80	-
⁶ BigBird (Zaheer et al., 2020)	92.31	-	95.20	92.20	-

Table 2: End-to-end classification performance of different models on the test set. The numbers are in percent.

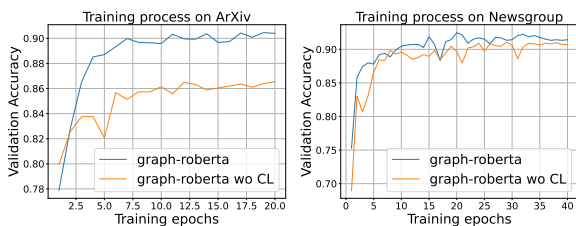


Figure 5: Validation accuracy over the training process.

short text. Also we see that contrastive learning indeed helps improve the final performance. Fig. 5 shows the end-to-end training processes of graph-roberta models on arXiv and Newsgroup datasets. It demonstrates that contrastive learning task speeds up the finetuning progress and helps learn a better model.

⁶The reported numbers from Beltagy et al. (2020); Zaheer et al. (2020) on arXiv and Hyperpartisan datasets are not

3.3 Document Retrieval

In this section, we extend our model to embedding-based document retrieval task. In this case, we consider the query as a single-node graph, where the representation is computed by the initial node representation. With that, we apply dot-product similarity to retrieve relevant documents. Our approach is essentially representation-based model.

Contrastive Pretraining To better align with the retrieval tasks, during pretraining, we sample one passage from each document as the sub-document and the rest as the other sub-document instead of an even random split, and we only compute the contrastive loss over the long sub-documents. In addition, we set 50% of the time we select the first passage of each document and 50% of the time comparable with ours because they did not release the train/test split of the data.

we sample uniformly from the document. This is very similar to the Inverse Cloze Task (ICT) introduced from Lee et al. (2019), except for the difference that ICT randomly selects one sentence from the passage whereas we randomly selects passages from the document. We pretrain the graph-roberta model on the OpenWebText data for 10 epochs with batch size 1,024, using Adam (Kingma and Ba, 2014) optimizer with learning rate 5e-5 and warmup rate 0.1.

Finetuning To finetune the model on ranking datasets, we use the similar training loss in the contrastive pretraining except we use the actual training queries-documents pairs. Besides in-batch negatives, we also sample additional negative candidates either uniformly or from some hard negatives such as the top BM25 retrieval pool for each training query.

First, we run the experiment with Robust04 dataset. We train and cross-validate machine learning models on the given 5 folds. During each run, we finetune the model 10 epochs with batch size 32, using Adam optimizer with learning rate 5e-5 and warmup rate 0.1. We sample additional 8 random negatives uniformly for each training query. We compare our models with BM25 baseline. The results are shown in Table 3. First we can see that the contrastive pretraining significantly improves the graph-roberta model performance (e.g. P@20 improves by over 100%). Still as a retrieval model, graph-roberta underperforms BM25. We conjecture that there are two reasons. (1) robust04 query set is too small to train such a complex neural representation model. (2) robust04 queries are all short key word queries, which favor lexical-matching methods such as BM25 over contextual transformer models. Nevertheless, we combine the retrieval results from graph-roberta model and BM25 through a weighted average of their scores (the weight is selected through cross-validation), we improve the nDCG@20 by 2% in absolute value over BM25, which indicates our model compensates BM25 results for semantic matching.

Now we present the experiment on much larger document ranking dataset MSMARCO. We finetune graph-roberta models on MSMARCO training set with batch size 128 for 10 epochs. For each training query, we also sample one hard negative in addition to the batch negatives. For the first 5 epochs, we randomly sample one negative from the top 100 BM25 retrieval results. For the latter

Model	Robust04	
	nDCG@20	P@20
BM25	41.63	35.68
Graph-roberta w/o CL	13.97	11.39
Graph-roberta	20.02	23.89
Graph-roberta +BM25	43.90	37.21

Table 3: Document retrieval on the test set of Robust04 dataset. The numbers are in percent.

5 epochs, we randomly sample one from the top 100 results retrieved using the 5-epoch checkpoint model. We also sample 100 queries from the official training set as our own validation set to monitor the training progress. We report the retrieval performance (without reranking) in Table 4.

Model	Dev	Trec DL
	MRR	nDCG@10
BM25	25.87	52.97
DE-Hybrid-E (Luan et al., 2020)	28.70	59.50
ME-Hybrid-E (Luan et al., 2020)	31.00	61.00
ACNE <i>FirstP</i> (Xiong et al., 2020)	37.40	61.50
Graph-roberta w/o CL	33.69	50.95
Graph-roberta	34.85	54.05
Graph-roberta +BM25	37.60	61.44

Table 4: Document retrieval on the test set of MSMARCO document dataset. The numbers are in percent.

Similarly to Robust04 experiment, contrastive learning as a pretraining strategy again improves the graph-roberta model performance. Note the improvement on MSMARCO is not as significant as in Robust04. Considering the fact that MSMARCO has a much larger training set, it is expected that the benefit of pretraining is less. Comparing with BM25, graph-roberta as a dense retrieval method achieves almost 9 points better in MRR@100 on the official dev set. We also list the performance of the state-of-the-art neural retrieval methods. DE-Hybrid-E and ME-Hybrid-E methods (Luan et al., 2020) are the two hybrid sparse-dense models that combining BM25 and BERT encoded dense presentations. Note that graph-roberta already outperforms the hybrid models on the official dev set, indicating that the representation learned by the graph-roberta model is very effective. Lastly, combining graph-roberta and BM25 retrieval results through simple weighted average, gives us the similar performance by the SOTA method ACNE ⁷ (Xiong et al., 2020). Furthermore, we believe that

⁷ACNE *MaxP* which produces the best numbers (MRR

Model	Graph	WIKIR62K (Title)			WIKIRS62K (First Sentence)		
		P@5	P@10	P@20	P@5	P@10	P@20
BM25	-	28.66	22.17	16.62	22.20	16.74	12.65
MatchPyramid (Pang et al., 2016)	-	20.76	16.83	13.11	20.74	16.76	12.89
ConvKNRM (Dai et al., 2018)	-	17.46	14.87	12.16	18.94	16.51	13.05
Graph-roberta w/o CL	full	20.68	16.66	12.86	20.92	17.39	13.51
Graph-roberta	full	22.98	18.46	14.19	24.02	19.65	15.39
Graph-roberta + BM25	full	39.38	30.52	21.98	35.24	27.53	19.74
Graph-roberta w/o CL	section	20.16	16.07	12.56	21.98	17.93	13.86
Graph-roberta	section	23.32	18.69	14.16	23.78	19.78	15.27
Graph-roberta + BM25	section	39.70	30.64	22.13	35.40	27.45	19.86

Table 5: Document retrieval benchmark on the test sets of WIKIR datasets. The numbers are in percent.

the training strategy introduced in ACNE can also be applied on graph-roberta model training and we leave it to the future work.

To further demonstrate the effectiveness of graph-roberta models for document retrieval task, we evaluate our models on the two large document retrieval datasets created via WIKIR (Frej et al., 2019), namely WIKIR62K and WIKIRS62K. In this experiment, we also consider different graph structures in modeling the Wikipedia documents. Besides the default fully-connected graph, we also consider the section structure information in the documents. Specifically, we consider the structure that all the passages within each section are mutually connected. The document node and the first passage nodes are connected with each other. We denote this graph as the *section* graph. We finetune the models on the training data for 5 epochs with batch size 128, using Adam optimizer with learning rate $2e-5$ and warmup rate 0.1. For each query, we also sample one hard negative from the top 100 BM25 retrieved candidates. The final retrieval benchmark is shown in Table 5.

In Table 5, we see that contrastive pretraining consistently helps improve the model performance in both title queries and first-sentence queries. BM25 performs much better for title queries than the first-sentence queries, as observed in Frej et al. (2019) since title queries are usually keyword queries. Our graph-roberta model outperforms MatchPyramid (Pang et al., 2016) and ConvKNRM (Dai et al., 2018), and performs consistently on both title and first-sentence queries. We further combine the results of graph-roberta and BM25. Overall, the ensemble of BM25 and graph-roberta

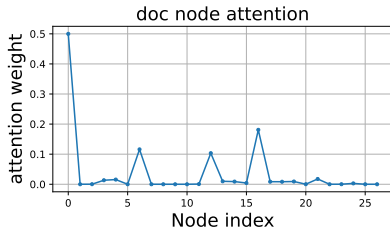
gives the best results.

We notice that for graph-roberta models, utilizing the section graph as described earlier performs slightly better than the default fully-connected graph, although the difference is small. We conjecture that on this dataset, the document representation does not rely much on the interaction between passages. We look into the graph attention patterns by the two models (graph-roberta with fully-connected graph and graph-roberta with section graph). We compute the average attention weights of the last graph attention layer. We observe that on both models, the document node usually attends to similar passages. As an example, we plot the graph attention weights on both models. Fig. 6 shows the attention weights of the document node. As we can see, for this example, both models attend to similar passages besides the document node. In Fig. 7, we observe that for graph-roberta with the fully-connected graph, all the other passage nodes have similar attention patterns as the document node, while for graph-roberta with the section graph, the passage nodes can actually learn some nontrivial patterns, which we believe could be beneficial for more complex tasks.

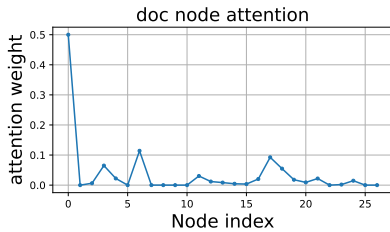
4 Related work

Document Representation Learning One line of related work is to utilize the successful pre-trained Transformer models (Radford et al., 2018; Devlin et al., 2019) to obtain contextual text representations. It has been shown to be successful on sentences and short passages in textual similarity tasks and passage retrieval tasks (Reimers and Gurevych, 2019; Minaee et al., 2021; Karpukhin et al., 2020; Liang et al., 2020). He et al. (2019); Dai et al. (2019); Zaheer et al. (2020) propose sparse attention to enable the transformer models to

38.38% on the official dev set) is not based on learned document embeddings, but on a set of passage representations for each document.

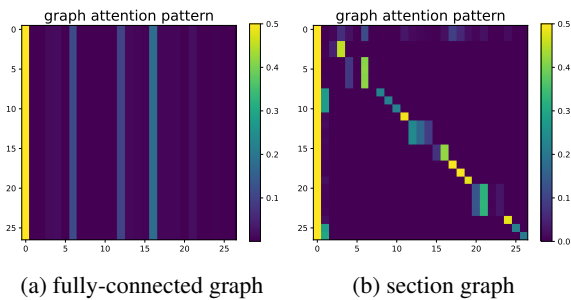


(a) Fully-connected graph



(b) section graph

Figure 6: An example of the document node attention patterns on graph-roberta models. The axis is the graph node index. Node index 0 is the document node and the rest are passage nodes.



(a) fully-connected graph

(b) section graph

Figure 7: An example of the graph attention pattern on graph-roberta models. The axes are graph node index. Node index 0 is the document node and the rest are passage nodes.

handle long text sequences more efficiently. However these pretrained models still focus on token-level interactions. Jiang et al. (2019) proposed a hierarchical attention model on top of recurrent neural network to tackle text matching of long documents, which later was extended by Yang et al. (2020) to transformer architectures. Both works only focus on the text matching task. Pappagari et al. (2019) proposed a hierarchical transformer model to encode long documents, where they apply a recurrent network or transformer layer on top of the original BERT model. In our work, we use a GAT network which can better leverage the existing document structure and we design a simple and effective contrastive learning framework based on our graphic model.

Another line of related work is to use graph neu-

ral network for document modeling. Peng et al. (2018, 2019) proposed to use graph convolutional networks (GCN) to model document as a graph of words, which allows the model to capture long-distance semantics. Yao et al. (2019) built a single graph for a whole corpus based on both word-to-word and document-to-word relations, which is learned by a GCN model.

Contrastive Learning Contrastive learning used as a self-supervised pretraining method has been widely used in NLP models (Rethmeier and Augenstein, 2021). Token or sentence-level contrastive learning tasks have been shown to be very useful in learning better contextual presentations (Clark et al., 2020; Giorgi et al., 2020; Meng et al., 2021). There also have been works that propose data augmentations for contrastive learning. Fang et al. (2020) proposed to use back-translation to construct positive sentence pairs in their contrastive learning framework. Wu et al. (2020); Qu et al. (2020) proposed multiple sentence-level augmentations strategies to do sentence contrastive learning. Most of these work still focus on either local token-level tasks or short sentence-level tasks. In our work, we directly work on document-level contrastive learning task. More recently Luo et al. (2021) proposed to use multiple data augmentations such as synonym substitution and back-translation to do unsupervised document representation learning. The difference in our work is that we have a much simpler framework that does not require those hand-craft transformations and we demonstrate that our contrastive learning strategy as a pre-training task can help the downstream tasks across various datasets.

5 Conclusions

In this work, we propose a simple graph attention network model to learn document embeddings. Our model not only can leverage the recent advancement of pretrained Transformer models as building blocks, but also explicitly utilize the high-level structure of the documents. In addition, we propose a simple document-level contrastive learning strategy that does not require handcraft transformations. With this strategy, we conduct large scale contrastive pretraining on a large corpus. Empirically we demonstrate our methods achieve great performance on both document classification and document retrieval tasks.

References

- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.
- Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 285–294, Online. Association for Computational Linguistics.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.
- Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.
- Jibril Frej, Didier Schwab, and Jean-Pierre Chevallet. 2019. Wikir: A python toolkit for building a large-scale wikipedia-based english information retrieval dataset. *arXiv preprint arXiv:1912.01901*.
- John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*.
- Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. <http://Skylion007.github.io/OpenWebTextCorpus>.
- Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. 2019. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 7:40707–40718.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.
- Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic text matching for long-form documents. In *The World Wide Web Conference*, pages 795–806.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.
- Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.
- Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.

- Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.
- Dongsheng Luo, Wei Cheng, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Bo Zong, Yanchi Liu, Zhengzhang Chen, Dongjin Song, Haifeng Chen, et al. 2021. Unsupervised document embedding via contrastive augmentation. *arXiv preprint arXiv:2103.14542*.
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.
- Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.
- Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*.
- Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.
- Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844. IEEE.
- Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.
- Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip Yu, and Lifang He. 2019. Hierarchical taxonomy-aware and attentional graph capsule rnns for large-scale multi-label text classification. *IEEE Transactions on Knowledge and Data Engineering*.
- Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Jiawei Han, and Weizhu Chen. 2020. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. *arXiv preprint arXiv:2010.08670*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Nils Rethmeier and Isabelle Augenstein. 2021. A primer on contrastive pretraining in language processing: Methods, lessons learned and perspectives. *arXiv preprint arXiv:2102.12982*.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Ellen Voorhees. 2005. [Overview of the trec 2004 robust retrieval track](#).
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabza, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for document matching. *arXiv preprint arXiv:2004.12297*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.
- Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

A Training details for document classification

We list the hyperparameters for finetuning the models on 4 document classification datasets in Table 6.

hyperparameters	arXiv	Newsgroup	IMDB	Hyperpartisan
learning rate	1.00E-04	5.00E-05	5.00E-05	3.00E-05
batch size	32	32	32	32
epoch	20	20	20	15
warmup	0.1	0.1	0.1	0.1
weight decay	0.01	0.01	0.01	0.01

Table 6: Hyperparameters for document classification.