Improving Text Matching using Graph-based Text Representation

Jia Li(1155037855)
lijia@se.cuhk.edu.hk

Wenxuan Zhang (1155115930)
wxzhang@se.cuhk.edu.hk

Kelvin K.W. Ng (1155078582)
kwng6@cse.cuhk.edu.hk

Yi Liu (1155114657)
1155114657@link.cuhk.edu.hk

Abstract

One of the most important component in a modern information retrieval system is the scoring function which will assign a score given a query q and a candidate text item, for example, a document d. The scoring function heavily relies on the quality of the embeddings of the two text items, so text embedding is becoming an important topic in IR and NLP community. Most of the existing works, such as Paragraph Vector [5], SkipThought vectors [4], FastText [3], Self-Attentive [6], treat text as a sequence of words, which may not be optimal. In this project, we propose to utilize the power of graph-based text representation and graph convolutional network to improve the encoding quality, so that the scoring function could better measure the similarity between queries and documents. To simplify the representation of the graph, we introduce the novel supervised, self-attentive graph embedding method called SAGE [11], which embeds graph instances of arbitrary size into fixed-length vectors. And then we use a siamese graph convolutional network applied to the graphs for learning the similarity between them [12]. Through experiments on SemEval-2016 Task 3: Community Question Answering, Subtask A, we demonstrate that it produce outstanding performance with our proposed model.

CCS CONCEPT

• Information systems → Information retrieval → Document representation → Document structure; • Information systems → Information retrieval → Information retrieval query processing → Query representation; • Information systems → Information retrieval → Retrieval tasks and goals → Question answering;

KEYWORDS

graph embedding; knowledge graph; semi-supervised learning; siamese network; graph convolutional networks;

1 INTRODUCTION

Recently, some approaches begin to treat text as a graph, whose nodes are composed of words. Works like [7-9] fall into this category. Compared with the traditional sequence-of-words representation, graph of words representation capture the context in a more structured way. Also, it enables us to incorporate prior knowledge to such graphs. Here shows an example in 1, the sentence is a quote from the Sun newspaper, "after the Lewinsky scandal, Hillary was forced to issue a statement reaffirming her commitment to Bill". From an external knowledge graph, we know Hillary and Bill are couples and we can add edges to connect them in graph of words representation.

Knowledge graphs such as YAGO and DBpedia contain a lot of human knowledge. This knowledge significantly benefits the understanding of input text. For example, in [10], a subgraph of DBpedia is extracted as the representations of document and then document similarity is measured based on that.

Much progress has been made in geometric deep learning on graphs recent years. This gives us a hope to learn discriminative feature representation from the data rather than handcrafted kernel representations. In [2], spectral graph theory is utilized to learn locality and stationary features from spectral domain for fixed-size graphs and Chebyshev polynomials is used to achieve linear complexity for sparse graphs. In [1], LSTM and Residual Gated Graph ConvNets are integrated to learn locality and stationary features from spatial domain for variable-size graph.

One challenge in this problem is that the graph is a much too complicated input for building a classifier. To tackle this challenge, we introduce a new graph embedding method which embeds a graph instance of arbitrary size into a fixed-length vector. The graphs then are transformed to embedding vectors which are the common input format for classification. In this work, we use a siamese graph convolutional neural network applied to graphs to improve the encoding quality, so that the scoring function could better measure the similarity between queries and documents. The next challenging task is evaluating the similarity of two graphs. Obtaining a measure of global similarity between two graphs can facilitate classification and clustering problems. There are many proposed ideas to approach this problem. However, based on experiments these methods all have their own limitations. In this work, we propose to use the method of siamese graph convolutional neural network [12] for learning similarity metric between graphs. To summarize, our contributions are as follows:

• We propose a novel graph-based text representation for query and comments classification. And we propose a novel method by incorporating the knowledge graph into the graph which makes the representation more comprehensive.
• We introduce a novel supervised, self-attentive graph embedding method called SAGE to embed graphs of arbitrary size into fixed-length vectors, which are used as a common form of input for classification. This encoding not only reduce the inconvenience of graph representation but also capture the differences between nodes at the same time.
• We use the siamese graph convolutional neural network method for learning a similarity metric between graphs with known node correspondences. As a proof of concept, we demonstrate the model performance on SemEval-2016 Task 3: Community Question Answering, Subtask A.

2 METHOD

Fig.1 gives an example for the graph of words representation. The words are connected with edges according to the co-occurences in document with a sliding window. The construction of graph representation will be introdcued first in 2.1. Fig.2 shows the process of incorporating prior knowledge into the graph as well as adding weights for different edges, the corresponding work is discussed in 2.2. Converting the graph representation to a fixed length vector is vital for later processing with our proposed model. We use a novel technique which called SAGE for the task of this part in 2.3. Finally, in 2.4, we take the vectorizeded graph representation and fed it into the graph convolutional network for distance metric learning.

2.1 Construct graph from raw text

The task of our proposed model is handling question and answering problem. Therefore, in this work we will perform the similar process on both comment and query. The first part is related to graph construction. The idea is illustrated in Fig.1. We model a textual document as a graph of words, which corresponds to a graph whose vertices represent the unique terms in the document and whose edges explain the co-occurrences between terms within a fixed-size sliding window. The input text is then represented by a matrix. The same method will be applied to the query processing. After this step, we will get our raw text-graph representation and the query-graph rep-

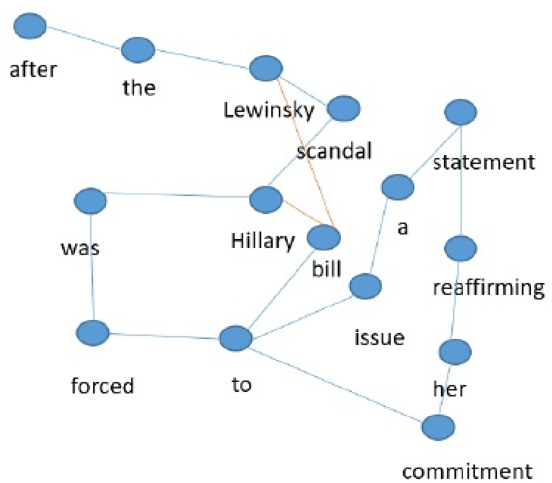

Figure 1: An example on graph of words representation.

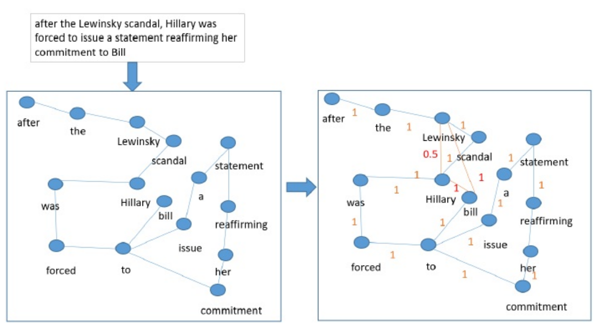

Figure 2: An example about constructing graph of words.

resentation. In addition to undirected graph shown here, directed graph can also be considered.

2.2 Incorporate prior knowledge

As the traditional graph-based text representation does not include external knowledge. In question answering problem, we often have the concern that some unique term has various highly relevant definitions, or maybe equivalent representations. For example, if we type the query like "What's the relationship between CUHK and CUHK(SZ)?", the traditional text representation-based search engine may fails to interpret the relations between "CUHK" and "The Chinese University of Hong Kong" or "SZ" is the abbreviation for "Shen Zhen" which is a city nears Hong Kong. In order to capture this kind of latent representation, we propose to incorporate external knowledge from other source into the graph (e.g.DBpedia). Speciaficly, we plug in the prior knowledge in knowledge graphs into the graph we have constructed in first step by adding edges with weight. For example, "Hillary" and "bill" is one-hop away in DBpedia, we add edge (Hillary, bill) with weight of 1. "Hillary" and "Lewinsky" is two-hop away in DBpedia, we add edge (Hillary, Lewinsky) with weight of 1/2. Fig.2 shows an example of this procedure.

2.3 Graph embedding

One of the challenge work is the interpretation of the graph representation. In the literature, graph representation techniques have recently shifted from hand-crafted kernel methods [14] to neural network based end-to-end methods, which achieve better performance in graph-structured learning tasks. In this vein, we adopt neural network methods for the graph embedding task.

Our graph embedding task is to produce a fixed-length discriminative embedding vector of a graph instance.To this end, we use a self-attentive graph embedding method, called SAGE [11], which can take a variable-sized graph instance, and combine each node to produce a fixed-length vector according to their importance within the graph. In SAGE, we first utilize a multilayer GCN [15] to smooth each node's fea-

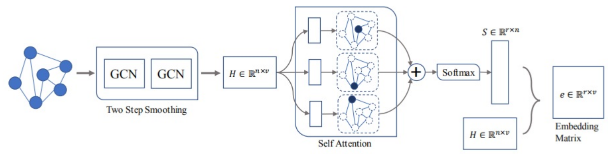

Figure 3: The supervised self-attentive graph embedding method SAGE.

tures over the graph's topology. Then we use a self-attentive mechanism to learn the node importance and then transform a variable number of smoothed nodes into a fixed-length embedding vector, as proposed in [16]. Then the output is multiplied with a self-attention mechanism to highlight the important node of the learned graph. Finally, the embedding vector is cascaded with a fully connected layer and a softmax function, in which the label information can be leveraged to discriminatively transform the embedding vector. Fig.2 depicts the overall framework of SAGE.

2.4 Graph convolutional networks

Given the constructed graph in step 1, we get the embedding of the graph from step 2. Then the vectorized representation of the graph constructed in step 2 and the vectorized representation of query are treated as input fed into siamese network with two graph convolutional networks [12]. The siamese network, presented in Fig. 4, consists of two identical sets of convolutional layers sharing the same weights, each taking a graph as input. An inner product layer combines the outputs from the two branches of the network and is followed by a single fully connected (FC) output layer with a sigmoid activation function and one output, that corresponds to the similarity estimate. The FC layer accounts for integrating global information about graph similarity from the preceding localised filters. Each convolutional layer is succeeded by a non-linear activation, i.e. Rectified Linear Unit (ReLU). The final result is a binary output indicates whether the comment is relevant to the input query.

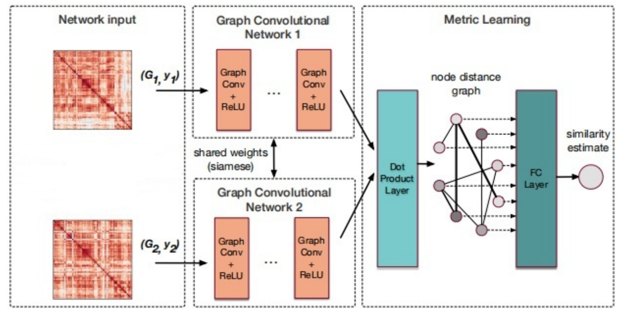

Figure 4: Pipeline used for learning to compare query and comment graphs.

3 EXPERIMENT

We evaluate the performance of the proposed model for similarity metric learning on the SemEval-2016 Task 3: Community Question Answering, Subtask A dataset.

4 RELATED WORK

There are mainly three research lines that are highly related to our work: Text Graph Representation, Graph Embedding and Graph Similarity Measuring.

A various of graph methods has been proposed for text representation. For different methods, these graph representation are mainly different on the construction of nodes and edges. Based on different types of graph nodes, the majority of existing works can be generalized as word graph, text graph, concept graph, and hybrid graph. Based on different types of graph edges, there can be different relation representations. However, none of these work consider including the prior knowledge of major concern.

As previously mentioned, the graph embedding method is related to the work of [11]. They use a novel supervised, self-attentive graph embedding method called SAGE to embed graphs of arbitrary size into fixed-length vectors. This embedding technique simplify the later process of graph classification work.

The estimation of (dis)similarity between two graphs is not unique, there are four main stream ap-

Table 1: Experiment Result

Algorithm	MAP	Precision	Recall
1 Kelp-primary	**79.19**	**76.96**	**55.30**
ConvKN-contrastive1	78.71	77.78	53.72
SUper team-contrastive1	77.68	75.59	55.00
2 ConvKN-primary	**77.66**	**75.56**	**58.84**
3 SemanticZ-primary	**77.58**	**74.13**	**53.05**
ConvKN-contrastive2	77.29	74.74	59.67
4 ECNU-primary	**77.28**	**70.46**	**63.36**
SemanticZ-contrastive1	77.16	75.29	53.20
5 SUper team-primary	**77.16**	**74.43**	**56.73**
MTE-NN-contrastive2	76.98	58.71	70.28
SUper team-contrastive2	76.97	74.31	56.36
MTE-NN-contrastive1	76.86	55.84	77.35
SLS-contrastive2	76.71	59.45	67.95
SLS-contrastive1	76.46	60.09	69.68
6 MTE-NN-primary	**76.44**	**56.28**	**76.22**
7 SLS-primary	**76.33**	**60.36**	**67.72**
ECNU-contrastive2	75.71	63.60	66.67
SemanticZ-contrastive2	75.41	73.19	50.11
ICRC-HIT-contrastive1	73.34	63.43	69.30
8 ITNLP-AiKF-primary	**71.52**	**73.18**	**19.71**
ECNU-contrastive1	71.34	66.95	41.31
9 ICRC-HIT-primary	**70.90**	**62.48**	**62.53**
10 PMI-cool-primary	**68.79**	**47.81**	**70.58**
UH-PRHLT-contrastive1	67.57	54.10	50.11
11 UH-PRHLT-primary	**67.42**	**55.64**	**46.80**
UH-PRHLT-contrastive2	67.33	54.97	49.13
12 QAIIIT-primary	**62.24**	**50.28**	**53.50**
QAIIIT-contrastive2	61.93	49.48	49.96
QAIIIT-contrastive1	61.80	49.85	50.94
13 OUR METHOD	**75.11**	**68.23**	**59.14**

proaches [13]: graph kernels, graph embedding, motif counting and graph edit distance. Recently, different neural network models have been explored to learn a similarity function that compares images patches [17, 18]. The network architectures investigated employ 2D convolutions to yield hierarchies of features and deal with the different factors that affect the final appearance of an image. In the work of [12], they use a novel method for learning a similarity metric between irregular graphs with known node correspondences by a siamese graph convolutional neural network using the polynomial filters formulated in [19].

5 Conclusion

In this work we aim at solving question and answering problem with our proposed graph model. Firstly, we construct the the initial graph representation for both query and comments with vertices as terms and edges co-occurences. In the second step, we propose to incorporate knowledge graph into the graph stucture. This step help us to interprete the terms with more comprehensive understanding. Then, we introduce a new method to generate embeddings for text, which digests text sequentially to form a graph. After that we use a siamese network to learn the similarity between query and comments, it outputs the relevance of the query and comments. We train the network on SemEval-2016 Task 3: Community Question Answering, Subtask A and demonstrate that our proposed model has outstanding performance on the task.

Future works include efficiency considerations and more crafted experiments. We expect that this method improves the encoding quality of text, so that different applications which make use of text embeddings, such as scoring function of search engine, could benefit from it. Depending on the knowledge graph used, domain-specific knowledge can be incorporated into the embeddings, so that domain-specific search engines could also benefit from the improved quality.

References

[1] Xavier Bresson and Thomas Laurent. Residual gated graph convnets, 2018.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 3837–3845, 2016.*

[3] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers, pages 427–431, 2017.*

[4] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 3294–3302, 2015.*

[5] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pages 1188–1196, 2014.*

[6] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations, 2017.*

[7] Giannis Nikolentzos, Polykarpos Meladianos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Shortest-path graph kernels for document similarity. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 1890-1900, 2017.*

[8] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pages 1063-1072, 2018.*

[9] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. Text categorization as a graph classification problem. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 1702-1712, 2015.*

[10] Michael Schuhmacher and Simone Paolo Ponzetto. Knowledge-based graph document modeling. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014, pages 543-552, 2014.*

[11] Jia Li[1,2], Yu[2], Hong Cheng[1], Helen Meng[1], Wenbing Huang[2], Junzhou Huang[1]. 2019. Semi-Supervised Graph Classification: A Hierarchical Graph Perspective. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13-17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313461.*

[12] Sofia Ira Ktena*, Sarah Parisot, Enzo Ferrante, Martin Rajchl, Matthew Lee, Ben Glocker, and Daniel Rueckert. Distance Metric Learning using Graph Convolutional Networks: Application to Functional Brain Networks.

[13] Livi, L., Rizzi, A.: The graph matching problem. Pattern Analysis and Applications 16(3), 253-283 (2013)

[14] P. Yanardag and S.V.N. Vishwanathan. 2015. Deep Graph Kernels. *In KDD. 1365-1374.*

[15] T. N. Kipf and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR.*

[16] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. 2017. A Structured Self-attentive Sentence Embedding. In *ICLR.*

[17] Kumar, B., Carneiro, G., Reid, I., et al.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *IEEE CVPR. pp. 5385-5394 (2016).*

[18] Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In *IEEE CVPR. pp. 4353-4361 (2015).*

[19] Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In textitNIPS. pp. 3837-3845 (2016).