Enhancing Academic Paper Search with Graph Neural Networks: A Test Beyond Classification

Amos Dinh[1,⋆], *Henrik* Rathai[1,⋆⋆], *Ilgar* Korkmaz[1,⋆⋆⋆], and *Matthias* Fast[1,⋆⋆⋆⋆]

[1] *Cooperative State University Baden-Wuerttemberg Mannheim, Computer Science Department, Coblitzallee 1-9, 68163 Mannheim*

Abstract.
This paper presents an innovative approach for academic paper search, utilizing Graph Neural Networks (GNNs) for advanced text classification. The proposed framework constructs a graph linking papers, words, and authors, where edges are weighted by tf-idf metrics and word co-occurrence statistics. GNNs, potentially incorporating models like TransE, are employed to learn entity embeddings. This enriches the representation of academic content beyond simple keywords, allowing for a vector database-powered similarity search. This method enables users to find related papers, explore topics through keywords, and connect with relevant authors, significantly enhancing the efficiency and depth of academic literature exploration...

1 Introduction

Write Introduction

2 Related Work

Compare to related Work [1, p. X] [2, p. XX]

Classification vs Search

3 Proposed Approach

Explain paper structure e.g. using picture

3.1 Text pre-processing

Extract new features from dataset: timestamp, pages

format columns

Lemmatization of abstract and title words

Delete stopwords

3.2 Graph Modeling

explain theoretical graph modeling

Create Hetero-Object, e.g. picture of simple graph

3.3 Assign edge weights

There are multiple ways to weight edges. In this paper the tf-idf metric is used for document-word edges and pmi for word-word co occurrencies.

term frequency-inverse document frequency (TF-IDF)

point-wise mutual information (PMI)

explain normalized pmi

To analyze word relationships across a corpus, you can use a fixed-size sliding window to gather word co-occurrence data. PMI to calculate the weights between word pairs. The weight of the connection between two word nodes (i and j) is thus defined using this approach.

$$A_{ij} = \begin{cases} \text{nPMI}(i, j) & \text{if } i, j \text{ are words and PMI}(i, j) > 0 \\ \text{TF-IDF}_{ij} & \text{if } i \text{ is document, } j \text{ is word} \end{cases}$$

3.4 Training

explain theoretical basis of training

4 Experiment Setup

Use case with arxiv dataset

4.1 Dataset

describe dataset

4.2 Implementation Details

e.g. used parameter

used embeddings

Graph Edge Setup: window size ist set to 10 for pmi calculation

Graph Learning Setup: layer, dropout rate , ...

⋆e-mail: Mailaddressforfirstauthor
⋆⋆e-mail: s212387@student.dhbw-mannheim.de
⋆⋆⋆e-mail: Mailaddressforlastauthorifnecessary
⋆⋆⋆⋆e-mail: Mailaddressforlastauthorifnecessary

4.3 Results

compare to tfidf, Benchmark, discuss results

5 Implementation/Deployment

explain implementation in vector database

6 Conclusion and Future Work

Recommendations for future work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

References

[1] L. Yao, C. Mao, Y. Luo, *Graph convolutional networks for text classification*, in *Proceedings of the AAAI conference on artificial intelligence* (2019), Vol. 33, pp. 7370–7377

[2] S.C. Han, Z. Yuan, K. Wang, S. Long, J. Poon, *Understanding graph convolutional networks for text classification* (2022), `2203.16060`