Link prediction based on graph neural networks using textual data for recommending Dutch news articles

Student	Internal supervisor	External supervisor
Naomi Rood (12666866)	Hongyun Liu	Felix van Deelen
naomi.rood@student.uva.nl	h.liu@uva.nl	felix.van.deelen@nos.nl
University of Amsterdam	University of Amsterdam	Nederlandse Omroep Stichting

1 INTRODUCTION

News articles are published online often. Each day, hundreds of news articles are added. People want to read what they are interested in and want to get a good understanding of a news topic. Online news platforms recommend news items to a user that is somehow related to a news article that a user reads at that moment. The explosion of news articles nowadays goes beyond the view limit of normal people and the chances of a reader missing news within his interest increases. It is important that the information overload problem in news articles will be tackled and news recommendations can play an important role to to help [9]. The NOS (a Dutch broadcasting organization) wants to recommend users certain articles efficiently. Currently, they suggest related news articles underneath most of the news articles which are recommended by manually assigning similar articles to the target article by the editorial team of the NOS. However, this is a time-consuming task.

A lot of research is available regarding news recommender systems and the literature on that is increasing in the past years [15]. Most of the recommender systems techniques are not interesting for this research, because it requires user reading history data. This research focuses on recommending news articles based on the content of the article since the news articles are publicly available. The popularity of graph-based recommendation models grows [3]. Graph-based learning methods see the recommender system's information from a graph's perspective. It has the ability to use structured data [22]. However, most of the state-of-the-art recommender systems using graph neural networks (GNN) require user input [20–22]. The relationship between users and items can be represented in a graph. On the other side, a graph structure that uses only information from items can also be used for recommendation, the links between items can be predicted using link prediction and the predicted links can be presented as recommendations to a user [17]. Three main traditional link prediction approaches are heuristic methods, latent-feature methods, and content-based methods [25]. Heuristic methods computed similarity scores as the likelihood of links (for example Common neighbors [12]). Latent-feature methods factorize matrix representations of a graph to learn the node embeddings (for example node2vec [4]). Content-based methods only focus on node attributes rather than the structure of a graph. However, Zhao et al [8] showed that combining graph features with node attributes can improve the performance of link prediction. Zang & Chen [26] created a new framework for link prediction that uses a subgraph enclosing technique around a link to make node embeddings. However, the input of the text is missing in this framework, but it is possible to give attributes about the articles to the nodes. They mention that their framework can be used in further research for recommender system applications. Furthermore, a Context-Aware Node Embedding method (CANE) is presented by Tu et al. [19] that uses the text of a node. It models the semantic relationships and graph structures between nodes into a context-aware node embedding. Their research focuses also on link prediction and is evaluated on three different datasets, two of them with English text and one in Chinese. However, it is not clear which framework they used to predict the links with their embedding method.

This thesis will focus on a new approach for a recommender system for Dutch news articles using data about the news articles. This means the Dutch text and metadata of the news article. The new link prediction framework of Zhang & Chen [26] will be combined with the CANE Context-Aware Node Embedding method that will use the text of Dutch news articles for node embeddings. The predicted links will be the recommended Dutch news articles.

1.1 Research question

The research question that follows up on this research gap is as follows: *To what extent can GNN-based link prediction using textual data recommend Dutch news articles?* Some subquestions are created for this research question:

- What kind of graph neural network framework can be used to predict links of Dutch news articles?
- How can we use textual information in a graphical representation of Dutch news articles to predict links?
- How can we process metadata from the news articles to be node attributes in the SEAl framework?
- How does a GNN link prediction model enriched with textual information compare to a text-only model such as TF-IDF in terms of AUC and precision?

2 RELATED WORK

This section will provide an overview of current techniques used for recommender systems and some new frameworks that have been implemented over the past few years. This section ends with a clear overview of the research gap.

2.1 News recommendations

Recommendation systems are systems that try to make relevant recommendations to users using a machine learning approach. Different approaches for making news recommendations already exist in the literature [15]. Usually, these are content-based recommendations that are recommended according to a similarity measure between articles already known to be preferred by a user [14]. This research focuses on non-personalized recommendations.

A common approach for recommendation without user input is using TF-IDF [13]. Each word gets a weight that shows how important a word is in a specific article. In the end, the cosine similarity is computed and this gives the similarity between two feature vectors of two articles. However, this method approaches all words independently and does not use the context of the words. A large similarity score between two articles will not directly mean that it will be a good recommendation, it only tells that the articles have a great similarity in the words. BERT is an approach that uses the context of words. Juarto & Girsang [6] used BERT with embedded sentences of news articles for a news recommender system. BERT [2] focuses on learning the context of words or sentences and creates embedding of the articles which can be used for recommendation. The latest research shows some new implementations of recommender systems for news recommendation [25]. Graph neural networks (GNN) became more popular over the years. GNNs have the ability to use previous recommendations to learn from. Liu et al [10] have discovered that a PMGT (a GNN-based method) outperformed the graph-BERT approach for recommending based on textual information.

2.2 Graph neural networks for link prediction

Graph neural networks have an increasing popularity in link prediction and are a powerful tool to learn from graph structure as well as from node structures together [25]. It has shown that it has big advantages over traditional methods for link prediction. A GNN usually consists of graph convolution layers that extract substructure features for nodes and a graph aggregation layer that uses node-level features to aggregate them into a graph layer feature. Mainly two paradigms for GNN-based link prediction exist. Node-based methods aggregate node embedding of the pairs of nodes connected to a link learned by a GNN, an example of such a method is Variational Graph AutoEncoder (VGAE) [7]. The second method is a subgraph-based method, this extracts a local subgraph around a link and uses the subgraph representation learned by a GNN to predict a link. The most important subgraph-based method currently is SEAL [26]. This method uses an enclosing subgraph for a link that needs to be predicted and then applies GNN to predict if the subgraph classifies to link existence.

2.3 Link prediction using SEAL

As explained in the previous section, the SEAL method extracts local subgraphs to learn subgraph representations which are learned by a GNN for link prediction [26]. Zhang & Chen have implemented a SEAL framework to predict links. The SEAL framework can learn together from Subgraphs, Embeddings, and Attributes for link prediction. The new element is that they enclose local subgraphs from a certain link and do not look at a total graph. This is because a subgraph already contains enough information to learn good graph structure features for link prediction. The SEAL framework consists of three steps. The first one is enclosing subgraph extraction, the second one is node information matrix construction and the third step is GNN training. This method combines graph structure features with side information about individual nodes. This side information is an embedding vector of a node and other node attributes (for example groups). h-hop enclosing subgraphs around two nodes connected to a link is used to define subgraphs. SEAL automatically learns graph structure features from the network using the enclosing subgraph method, the learning process is done using a GNN. The next step is node labeling which gives a feature about the role of a node in a subgraph, for example, the nodes to which a link is connected are the target nodes. These are different from the nodes that are further away from the link but still in the subgraph. A good node labelling of different roles around a link is important for GNNs to predict link existence. In their experiment, they used Node2Vec [4] to define node embeddings, however, they mention that any kind of network node embedding method is possible. Node embedding is a latent feature method that factorizes matrix representations of a network to learn a low-dimensional representation of a node. Besides latent feature methods, explicit features are used, this is available in the form of node attributes and describes any kind of side information about individual nodes. This framework is tested on eight different datasets with links and outperformed methods that used only graph-structured features. The SEAL method had a higher AUC score for link prediction on all eight different datasets in comparison with the VGAE [7] method, the VGAE method is a latent feature method that uses a GNN to learn node embedding that constructs the network best. This framework only looks at link prediction but suggestions for further research are given to include link prediction research in machine learning problems like recommender systems.

2.4 Text-based node embedding

Link prediction using GNN needs a node embedding method to learn graph structures. Zhang & Chen did compare their SEAL framework with methods that use only node2vec embeddings. Node embeddings are a way of representing nodes as vectors in a low-dimension space, these embeddings are learned from observed graph nodes. Their SEAL framework which uses a combination of learning from graph structures, as well as latent features, is better than using latent features alone. The SEAL framework makes it possible to use any kind of node embedding. In their implementation, node2vec is used. But this project will focus on textual data. Yang et al. mention that graph embedding methods like node2vec are great for low-dimensional dense vectors but not for using textual information [24]. CANE is a method that can make context-aware text node embeddings [19]. This embedding method uses the text of a node as information to make embeddings. It does not look only at graph structures like node2vec to make embeddings. It can model more semantic relations between nodes. Tu et al. did some experiments with different kinds of textual datasets. The AUC values for link prediction for CANE compared to baselines that used node embedding based on only graph structures were higher. Moreover, the AUC values of CANE compared to other textual node embedding methods (TADW [23], CENE [16]) were also significantly higher for three different datasets.

2.5 Research gap

This research will use a dataset that has never been used. This will be a dataset from the NOS with information about Dutch news articles and data about links between them. The SEAL framework will be implemented on the NOS dataset which is never done before. By revisiting the aforementioned works, this work will implement the SEAL framework on the Dutch news articles and links datasets for the first time. Furthermore. This research will integrate the SEAL framework with the context-aware CANE node embeddings and also apply this to the NOS dataset to do link prediction, which is also novel.

3 METHODOLOGY

3.1 Data sets

This research will be carried out in collaboration with the NOS. For this research, two data sets provided by the NOS will be used. The first data set contains data about all news articles from the NOS from April 2008. In total there are 475796 news articles. This contains information about the id, title, description, text, publishing data, and keywords of a news article.

The second data set contains all links between articles. These links are the recommended news articles underneath the text of a news article or a news article that is mentioned in the text of an article. This is hand-labelled by the editorial teams of the NOS. Most of the articles contain 0,1,2 or 3 linked articles, there are in total 305586 links. This dataset will be used for training the model.

3.2 Implementation

The SEAL framework with the CANE node embedding will be implemented for link prediction between Dutch news articles. The links that will be predicted are the recommended items. Zhung & Chen stated that any GNN or node embedding can be used in SEAL. Like in their implementation, the architecture DGCNN [27] as GNN will be used. But as an embedding method, CANE will be used that uses the text from the news articles to make node embeddings based on the text. Apart from the positive examples (links between articles), negative examples need to be created. So for each positive example, a negative example will be randomly chosen (a non-existing link). The dataset of the links will be split into a training and test set. 90% of the positive and negative links will be used for training the model. The other 10% will be used for testing. The open-source code for the SEAL implementation is available on GitHub as well as the implementation for CANE node embeddings. CANE needs the text from the Dutch news articles as well as the links between articles. The text that will be used is the title and the text from the articles dataset. Apart from the node embeddings that the SEAL framework uses, node attributes can also be used. Any kind of side information other than its structure can be used as explicit features [26]. Features that can be used from the article dataset are published_at, owner, sub_category_list. Published_at is the publication date. The owner is the department of NOS that owns the article, for example, news or sports. Sub_category_list represents a broad subcategory of the article, for example, foreign countries, politics, or economics.

3.3 Evaluating

The proposed link prediction model of SEAL combined with CANE for recommending Dutch news articles will be evaluated. Two standard metrics are often used to quantify the accuracy of a predictive link model, these are area under the curve (AUC) and precision [11]. The AUC can be seen as the probability that a randomly selected missing link is more likely to appear than a randomly chosen nonexistent link [5]. Precision represents the number of good predicted links divided by the total number of predicted links. But concerning the purpose of this research, it will not be a limited evaluation metric. The dataset of the hand-labelled links contains possible recommended items. This does not directly mean that these are the only correct articles. False positive articles, could be good recommended articles as well. An evaluation metric that will not directly punish false positives is recall. Recall is used when the objective is to minimize false negatives. In this case, we want the hand-labelled links as the predicted links. Recall is also a commonly used evaluation metric in link prediction [1, 18].

3.3.1 *Baselines.*

To validate the performance SEAL framework with the CANE node embedding based on Dutch text, it will be compared against several baseline methods. The first one will be a random link predictor. The second method is the SEAL [26] implementation without the CANE context-aware text node embeddings. The node2vec embedding method will be used that only uses the structural information of the graph instead of the textual information of the nodes. The last method where the new approach will be compared is not based on link prediction but is a textual approach. This is TF-IDF. All words in news articles will be vectorized by a TfidfVectorizer [13]. Based on this, the cosine similarity will be calculated for two different articles, the most similar news articles to a certain article will be recommended. TF-IDF uses the text but does not use the context of the text or any kind of graph structure.

4 RISK ASSESSMENT

First of all, the data is already been given by the company. Opening the data does not cause any problems and the data looks very clean. This will likely not give any problems.

The article of Zhang & Chen [26] has open-source code on GitHub. This will likely mean that the implementation of the link prediction model will not give any big problems. A part where problems may arise is in the node embeddings. This is because a different node embedding method will be used than implemented in the experiment of the SEAL framework. The CANE node embedding method has also code available on GitHub [19]. However, it could be the case that for some reason it can not be used by the SEAL implementation. If that happens, a different node embedding method that uses text needs to be used. However, these will probably have worse performance than CANE [19].

Another identified risk can be that the dataset is too big to train the model with since a large number of Dutch news articles are provided by the NOS. A solution could be the work with a subset of the data.

5 PROJECT PLAN

Table 1 shows the achievements in order to complete the thesis per week. The thesis will be a full-time project from April, but March already has the first milestone after completing the thesis design. The part-time period is stated with weeks 0. The planning takes into account the milestones that must be handed in on Canvas. Enough time is allocated in different weeks to write the thesis. I know that I want to revise different parts multiple times before I want to hand in different section milestones of the thesis. The last

Week	Date	Achievement
Weeks 0	20/2 - 19/3	Finished the exploratory data analysis and handed in on Canvas
Weeks 0	20/3 - 2/4	Have the baseline method TF-IDF implemented, and ready for evaluation.
Week 1	3/4 - 7/4	Finish methodology section and hand in by supervisor for feedback. Started with implementing the baseline SEAL method.
Week 2	10/4 - 14/4	Implement feedback and finish draft section Methodology and handed in on Canvas, worked on the baseline SEAL implementation.
Week 3	17/4 - 21/4	Started with the CANE node embedding implementation and finished baseline SEAL implementation.
Week 4	24/4 - 28/4	Worked on implementing CANE node embedding in the SEAL framework and started with evaluation.
Week 5	1/5 - 5/5	Finished CANE node embedding implementation in the SEAL framework. Finished the evaluation and started writing the result section.
Week 6	8/5 - 12/5	Finished writing result section and handed in by supervisor for feedback.
Week 7	15/5 - 19/5	Implement feedback and finished draft result section and handed in on Canvas.
Week 8	22/5 - 26/5	Finished writing Introduction and Related work section. Started thinking and writing discussion.
Week 9	29/5 - 2/6	Finish conclusion and discussion section, hand in for feedback supervisor.
Week 10	5/6 - 9/6	Implemented feedback conclusion and discussion section, revised complete thesis.
Week 11	12/6 - 16/6	Finished draft Thesis and handed in on Canvas
Week 12	19/6 - 23/6	Implement last feedback and some buffer time
Week 13	26/6 - 30/6	Buffer time and finished final thesis and handed in on Datanose

Table 1: Week-by-week planning for the thesis project

two weeks represent some buffer time. In case some writing parts still need to be revised. April is mainly focused on implementing the new model and baseline models and the first half of May is focused on the evaluation.

REFERENCES

[1] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, Vol. 30. 798–805.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. 2022. Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1623–1625.

[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[5] James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 1 (1982), 29–36.

[6] Budi Juarto and Abba Suganda Girsang. 2021. Neural collaborative with sentence BERT for news recommender system. *JOIV: International Journal on Informatics Visualization* 5, 4 (2021), 448–455.

[7] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[8] Chaozhuo Li, Senzhang Wang, Dejian Yang, Zhoujun Li, Yang Yang, Xiaoming Zhang, and Jianshe Zhou. 2017. PPNE: property preserving network embedding. In *Database Systems for Advanced Applications: 22nd International Conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, Proceedings, Part I 22*. Springer, 163–179.

[9] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. 31–40.

[10] Zhuang Liu, Yunpu Ma, Matthias Schubert, Yuanxin Ouyang, and Zhang Xiong. 2022. Multi-Modal Contrastive Pre-training for Recommendation. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*. 99–108.

[11] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390, 6 (2011), 1150–1170.

[12] Mark EJ Newman. 2001. Clustering and preferential attachment in growing networks. *Physical review E* 64, 2 (2001), 025102.

[13] Constituency Parsing. 2009. Speech and language processing. (2009).

[14] Michael Pazzani and Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine learning* 27 (1997), 313–331.

[15] Shaina Raza and Chen Ding. 2022. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review* (2022), 1–52.

[16] Xiaofei Sun, Jiang Guo, Xiao Ding, and Ting Liu. 2016. A general framework for content-enhanced network representation learning. *arXiv preprint arXiv:1610.02906* (2016).

[17] Nitish Talasu, Annapurna Jonnalagadda, S Sai Akshaya Pillai, and Jampani Rahul. 2017. A link prediction based approach for recommendation systems. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2059–2062.

[18] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. 2003. Link prediction in relational data. *Advances in neural information processing systems* 16 (2003).

[19] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. 2017. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1722–1731.

[20] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[21] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[22] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[23] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. 2015. Network representation learning with rich text information. In *Twenty-fourth international joint conference on artificial intelligence*.

[24] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.

[25] Muhan Zhang. 2022. Graph Neural Networks: Link Prediction. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao (Eds.). Springer Singapore, Singapore, 195–223.

[26] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems* 31 (2018).

[27] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.