Link Prediction for the Context of Dutch News Articles using the Graph Neural Network SEAL Framework with CANE Embeddings

submitted in partial fulfillment for the degree of master of science

Naomi Rood
12666866

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on 30-06-2023

	UvA Supervisor	External Supervisor
Title, Name	Hongyun Liu MSc	Felix van Deelen MSc
Affiliation	University of Amsterdam	Nederlandse Omroep Stichting
Email	h.liu@uva.nl	felix.van.deelen@nos.nl

ABSTRACT

With the proliferation of news articles, effectively managing information overload by providing relevant recommendations is critical. Graph Neural Networks have emerged as a promising approach for news recommendation systems through link prediction. The SEAL framework, a novel approach in link prediction research, utilizes enclosed subgraphs, node embeddings, and node attributes as inputs to train a Graph Neural Network that outputs a probability of link existence. However, it does not incorporate text information in its node embeddings. This study proposed a method for link prediction that combines the state-of-the-art framework SEAL with Context-Aware Network Embeddings (CANE). CANE integrates information from the text of surrounding nodes and the graph structure into a node embedding. Additionally, a new dataset from the NOS with 468k Dutch news articles and hand-labeled links representing article recommendations is presented. Evaluation on the NOS dataset using article text showed that SEAL with CANE embeddings (AUC = 88.53) has an improved ability to discriminate positive and negative links compared to SEAL (subgraphs only) (AUC = 80.69) and SEAL with Node2vec embeddings (AUC = 82.21). However, it is noteworthy that the text-only method TF-IDF outperforms the proposed method in recommending articles.

KEYWORDS

Dutch news articles, Graph Neural Network, link prediction, node embedding, SEAL

GITHUB REPOSITORY

https://github.com/Naomirood/SEAL-CANE

1 INTRODUCTION

In the present era, the rapid growth of digital news articles has become a ubiquitous phenomenon, leading to an influx of a plethora of new articles on a daily basis. Given this circumstance, people tend to consume material that matches their individual preferences and contributes to their in-depth understanding of current events. In order to accommodate these user preferences, online news platforms utilize recommendation systems that use the user's current reading material to suggest relevant news articles. The abundance of news articles exceeds the average person's ability to fully consume them, increasing the likelihood of readers missing important news items. It is important that the problem of information overload in news articles is tackled, and news recommendations can be an important part of the solution [10]. The NOS (the Dutch Public Broadcasting Foundation) uses recommendation systems to efficiently provide users with relevant articles. This research introduces a new dataset of news articles with article text written in Dutch and a dataset of recommendations between articles from the NOS. These recommendations are manually assigned by the editorial team of the NOS. Furthermore, the research includes experiments on link prediction using GNN models.

There are different techniques available today for news recommendation systems, Raza and Ding providing an overview [16]. Since news articles are publicly available, this research focuses on recommending news articles based on the content of the article. Graph-based recommendation models are becoming popular [4, 16], they look at the available recommendations from the perspective of a graph. However, user input is required for most state-of-the-art recommendation systems using Graph Neural Networks (GNN) [24, 26, 27]. On the other hand, a graph structure using only information from items can also be used for recommendation. The links between items can be predicted using a link prediction method [19]. Currently, GNNs achieve top performance in link prediction with AUC scores around 0.94 as compared to 0.56-0.93 in previous approaches [31, 33]. Zhang and Chen [31] proposed a new link prediction model learning from local enclosed subgraphs, node embeddings that capture the structural information of the nodes in the graph, and node attributes that represent side information from the nodes. The current framework does not have the ability to incorporate learning from node embeddings that capture textual information, which is a potential way to exploit the textual content of news articles. Notably, the authors acknowledge that their framework paves the way for future research in recommendation system applications. In this study, we investigate if we can perform link prediction by expanding the SEAL framework with node embeddings that learns from textual data. In order to do this, we use the Context-Aware Network Embedding (CANE) method presented by Tu et al. [22], which uses the text of the nodes. CANE node embeddings model the semantic relationships and graph structures between nodes. Moreover, we use a new dataset of Dutch news articles in this research, this data is the only dataset in Dutch containing editorially selected links between articles.

In short, this research aims to propose a new approach for recommending Dutch news articles using metadata about the articles. The link prediction framework of Zhang & Chen [31] will be combined with the CANE method, which uses the text of Dutch news articles for node embeddings. The combination of these methods results in a new framework for link prediction. Additionally, this research uses Dutch news article data, which is also novel.

1.1 Research Question

The research question that follows up on this research goal is as follows: *To what extent can a GNN-based link prediction model benefit from textual data incorporated in node embeddings in the context of Dutch news articles?* Some subquestions are created for this research question where the AUC and recall refer to link prediction evaluation metrics [12, 21]:

- How does a GNN-based link prediction model, when used as a recommendation system, compare to a text-only model in terms of AUC and recall?
- To what extent does incorporating the title, title + description, or title + description + article text of the Dutch news articles from the NOS, as well as considering the temporal aspect, affect the AUC of link predictions?
- What is the behavior of articles outside the network when introduced to a link prediction model?

2 RELATED WORK

This section provides an overview of existing techniques used in news recommendation systems and link prediction. The section ends with a clear overview of the research gap.

2.1 News Recommendations

Recommendation systems for news articles use a variety of techniques to provide users with personalized and relevant recommendations. State-of-the-art approaches often involve building an item or user profile that represent individual items or capture user preferences and interests [16]. However, user input is not always available. In such cases, alternative methods that rely solely on content features are valuable options for generating recommendations. TF-IDF is one such technique that operates solely on the content of news articles without requiring explicit user input [14, 16]. TF-IDF calculates the importance of a term in an article by taking into account its frequency within the article and inversely weighting it by its presence across all articles in the collection. This generates a TF-IDF vector representing each article, which can be used to calculate cosine similarity with other news articles. However, this method treats all words independently and does not use the context of the words. A high similarity score between two articles does not directly mean that it will be a good recommendation, it means that the articles have a high similarity in the words.

BERT is an approach that uses the context and semantic meaning of words. Juarto and Girsang [7] used BERT [3] with embedded sentences of news articles for a news recommendation system. BERT is a language model designed to learn from textual data. It captures contextual information within the text and embeddings can be created for recommendation purposes. While BERT can benefit from article data to understand the content, GNNs offer an additional advantage by taking into account editors' recommendations to learn directly from them. Recent research shows some new implementations of recommendation systems for news recommendation using GNN models [30]. GNNs have the ability to learn from manually curated recommendations. Liu et al. [11] found that a PMGT (a GNN-based method) outperformed the graph BERT approach for recommendations based on various information sources, including text.

2.2 Link Prediction

There are three main traditional approaches to link prediction: heuristic methods, latent-feature methods, and content-based methods [30]. Heuristic methods compute similarity scores as link probabilities (for example Common Neighbors (CN) [13]). Latent-feature methods factorize matrix representations of a graph to learn node embeddings (for example Node2vec [5]). Content-based methods focus only on node attributes and not on the structure of a graph. Zhao et al. [9] showed that link prediction performance can be improved by combining graph features with node attributes. GNNs have become increasingly popular in link prediction and are a powerful tool for learning from both graph structure and node information together [30]. It has shown great advantages over traditional link prediction methods. A GNN typically consists of graph convolution layers, which extract substructure features for nodes, and a graph aggregation layer, which uses node-level features to aggregate them into a graph layer feature. Mainly two paradigms exist for predicting links using GNNs. The first is a node-based method that aggregates the node embeddings of connected nodes learned by a GNN. An example of such a method is the Variational Graph AutoEncoder (VGAE) [8]. The second method is a subgraph-based method, which extracts a local subgraph around a link and uses the subgraph representation learned by a GNN to predict a link. The state-of-the-art subgraph-based method is SEAL [31]. This method uses an enclosing subgraph technique for a link that needs to be predicted and then uses GNN to predict the existence of the link.

2.3 Link Prediction using SEAL

As explained in the previous section, the SEAL method extracts local subgraphs to learn subgraph representations by a GNN for link prediction [31]. Zhang and Chen have proposed the SEAL framework to predict the probability of whether two nodes in a network are likely to have a link. The SEAL framework can learn together from enclosed subgraphs, node embeddings, and node attributes for link prediction. Node embedding is a latent feature method that factorizes matrix representations of a network to learn a low-dimensional vector representation of a node in the graph, Zhang and Chen used Node2vec [5] to define node embeddings. Explicit features are available in the form of node attributes and describe any kind of side information about individual nodes such as metadata. The novelty of the SEAL framework is that it includes the local subgraph of a link and uses it to train the GNN instead of looking at the whole graph. This is possible because a subgraph already contains enough information to learn good graph structure features. The SEAL framework consists of three steps. The first step is to extract enclosing subgraphs, the second step is to construct a node information matrix from the three input features and the third step is to train the GNN. h-hop enclosing subgraphs enclose local subgraphs around two target nodes within h-hops of the target nodes, this number of hops defines the h-order heuristics. Learning first-order heuristics such as CN [13] as graph structure features tells us something about how likely two nodes are connected. It is also shown that higher-order heuristics have better performance than first- and second-order [12]; these higher-order heuristics require knowledge of the whole network rather than subgraphs, but also come with high time and memory consumption for most networks. However, the theory of the SEAL framework shows that it is possible to learn higher-order heuristics with a small value of h. Local subgraphs already contain enough information to learn good graph structure features, it is possible to accurately compute first- and second-order heuristics and a wide range of high-order heuristics with a small error. The SEAL framework is evaluated on eight datasets and has a higher AUC (average AUC = 94.20) for link prediction on seven out of eight different datasets compared to different latent feature methods. For example, the VGAE [8] method, which is a latent feature method that uses a GNN to learn the node embeddings that best construct the network (average AUC = 84.15). Furthermore, the AUC is higher for all datasets compared to heuristic methods such as CN (average AUC = 82.96)[13] or Wisfeiler-Lehman graph kernel (average AUC = 92.67)[17], where the latter also uses enclosing subgraphs but not GNNs.

2.4 Text-based Network Embedding

SEAL learns from node embeddings among other information to predict links. Yang et al. [29] mention that graph embedding methods like Node2vec are great for low-dimensional dense vectors, but not for using textual information like text from news articles. Tu et al. [22] propose the Context-Aware Network Embeddings (CANE) framework. This embedding method uses the text of connected nodes as information to learn node embeddings. Unlike Node2vec, which is a structure-only embedding method, it can model more semantic relationships between nodes by looking at the text. The AUC values of CANE for link prediction were higher than baselines that used node embeddings based on graph structure features only. Furthermore, the AUC values of CANE were also significantly higher compared to other textual node embedding methods (for example [28] and [18]).

2.5 Research Gap

The present research will utilize a previously untapped dataset obtained from the NOS, which contains extensive information regarding Dutch news articles and their interrelatedness. Such a dataset in the Dutch language has not been researched before. It is worth noting that the utilization of the SEAL framework on the NOS dataset or on a Dutch dataset in general has not been investigated before, which makes this research project unique in its methodology. In addition, the study will introduce a novel approach by integrating the SEAL framework with CANE embeddings and using this combination for link prediction on the NOS dataset. This integration will bring a new perspective to the study.

3 METHODOLOGY

In this section, first of all, a description of the new dataset is given and an exploratory data analysis is performed. Secondly, a new method for link prediction based on textual data embedded into node embeddings is proposed.

3.1 Data Description

This section describes two datasets related to the NOS: the NOS links data and the NOS article data.

3.1.1 NOS Links data. The first dataset presented contains all links between articles, the total number of links is 208,610. This dataset is hand-labeled by the editorial teams of the NOS, which has typically assigned 0, 1, 2, or 3 articles as recommendations for a given article. The links represent the recommended news articles below the text of a news article. The dataset contains two attributes. These are as follows:

- `parent_article`: the article id of the article for which recommendations are begin made
- `child_article`: the article id of the article which is the recommended article for the parent_article

An example of the hand-labeled links for a specific article can be found in Appendix A. Exploring the data showed that before the year 2014, none of the articles had any labeled recommended articles (Figure 1). From 2014 onwards, the number of articles with linked articles increased over time. However, a significant proportion of articles in 2022 still have no recommended articles. The average number of recommended articles for articles with at least one recommendation is 2.17. In terms of graphs, the average node degree is 2.17. The average node degrees of datasets used by Zhang and Chen [31] vary from 2.49 to 27.36 (for example Power [25] has an average node degree of 2.67 and Yeast [23] of 14.46). Figure 7 in Appendix B shows the difference in hours between when a `parent_article` is published and when the `child_article` is published. For most of the links, there is only a maximum of one week difference between when the two articles were published.

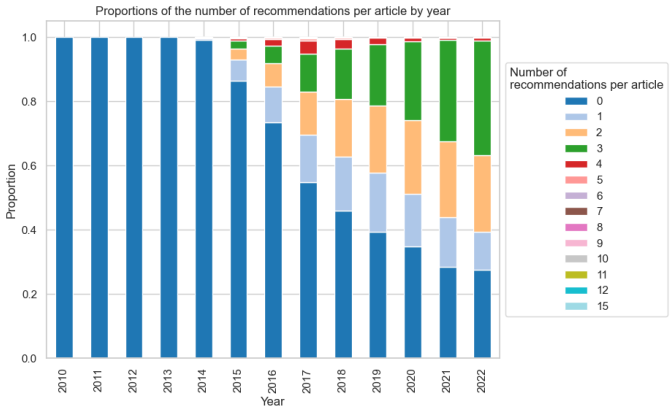

Figure 1: Proportions of the numbers recommendations per news article by year

3.1.2 NOS Article data. The second dataset contains data from all news articles from the NOS from 2010 to the end of 2022. In total, the dataset contains 468,695 news articles. The data consists of the following metadata where all information is written in Dutch:

- `article_id`: the unique id for the article
- `article_title`: the title of the article
- `article_description`: the description of the article
- `article_text`: the text of the article
- `published_at`: the date and time the article was published
- `owner`: the editorial team that published the article, for example, news, sports or youth
- `subcategory_list`: a list of predefined categories to which the article belongs, provided by the editorial team
- `system_tag`: an event tag, if the article belongs to a special NOS event page

An example of the metadata of an article can be found in Appendix A. An example of an attribute is the `subcategory_list`, an article can have more than one subcategory. In total, there are 109 different subcategories. The twenty most frequent and twenty least frequent categories are shown in Figure 8 in Appendix B. The categories most frequently assigned to the articles are 'buitenland' (foreign affairs), 'binnenland' (internal affairs), and 'voetbal' (soccer). More than 50,000 articles are assigned to these categories. On the other hand, some categories are only assigned once, such as wheelchair fencing or goalball. To give an indication of the text lengths of the available text corresponding to the article dataset. The mean and median number of words in the `article_title`, `article_description`, and `article_text` in the article dataset are presented in Table 1.

	Title	Description	Article text
Mean	6.70	18.34	242.51
Median	6	18	186

Table 1: Mean and median number of words in the title, description, and text of the articles

3.2 Models

3.2.1 Proposed Method. Following the success of the SEAL method for link prediction[31], we propose a new method based on the SEAL framework that incorporates textual data into node embeddings. This project uses an unpublished dataset consisting of Dutch textual data. Since the SEAL framework predicts links from node embeddings that do not use textual data, we have added the CANE method [22] to the existing SEAL framework. The context-aware network embeddings are used to define the node embeddings of each node in a graph [22]. Figure 2 shows the pipeline of this approach. The nodes in the graph represent the Dutch news articles, while the edges represent the links to the recommended articles. The proposed method uses three types of input: local enclosed subgraphs, node embeddings generated by the CANE method, and optional explicit features representing node attributes. These inputs are then used to train a GNN for link prediction. The CANE framework concatenates the text embeddings and the structural node embeddings to the CANE node embedding. All this information is fed into a GNN to make predictions about the likelihood of an existing link between two nodes.

Looking at the new approach in more detail, the GNN uses three different types of input of the graph from which it learns to predict links: local enclosed subgraphs, latent features (node embeddings), and explicit features (node attributes). The locally enclosed subgraphs around two nodes are defined by the parameter h (number of hops). The explicit features are node attributes that represent any side information of the node, they must be discrete or continuous vectors. We used CANE embeddings as node embeddings, these embeddings are context-aware of their neighbors. The CANE embedding is a concatenation of the structure-based embedding, which captures information about the network structure, and the text-based embedding, which captures the textual meaning of itself and the surrounding nodes. The structural embedding is generated by the LINE method [20], which computes a conditional probability of a node being generated by another node. The text-based embedding uses mutual attention to enable the pooling layer in a convolutional neural network (CNN) to be aware of the node pair, so that the text can influence the embedding of the other node and vice versa. The process of generating a CANE embedding is as follows and is shown in Figure 3. The text belonging to a node goes through a convolutional layer and the output is a matrix. From these matrices of two connected nodes, a correlation matrix is computed. Then a mean-pooling operation is performed (row- and column-wise) to produce vectors, these vectors are transformed into attention vectors by a softmax function. Finally, the text embedding is computed by multiplying the attention vector with its corresponding matrix. The CANE embedding can be computed from the text and structure embeddings.

The link prediction process consists of three steps: subgraph extraction, node information matrix construction, and GNN training. The GNN takes as input both the adjacency matrix (A) and the node information matrix (X). The node information matrix (X) contains structural node labels, CANE embeddings, and node attributes. In order to indicate the different roles of a node in the enclosed subgraph, we have marked the nodes with a structural node label. The GNN needs to know which nodes to predict the link between, and it provides additional information about the relative positions of the other nodes to the target nodes. During training, we used negative links in addition to positive links to generalize the GNN model. In this way, it learns to distinguish between the features of nodes that are linked and those that are not. The framework outputs a score between zero and one that represents the probability of an existing link between two nodes.

3.2.2 Models. We trained and evaluated the following baseline and proposed methods to compare their performance. The models are compared on one month of the data and on all data.

- **Random predictor** In order to have a simple baseline, we used a random link predictor, where the test links are assigned a random score between 0 and 1 as the probability of an existing link.
- **TF-IDF** A second method is not based on link prediction but is a textual approach for recommendation systems. The text used in an article is vectorized by a TfidfVectorizer [14]. The cosine similarity between two articles can be calculated from these vectors. This model is used to evaluate the link prediction model with a text-only model.
- **SEAL (subgraphs only)** The state-of-the-art SEAL [31] implementation for link prediction is used as a baseline, this method only learns from the enclosed subgraphs.
- **SEAL + Node2vec** In the original implementation of SEAL, it is optional to use the Node2vec [5] embeddings, which capture the structure of a graph. The node information matrix from the previous model is extended with Node2vec embeddings.
- **SEAL + Node2vec + attributes** This is the same model as the previous one, but with the attribute vectors of the nodes added, which is also optional in the original SEAL implementation.
- **SEAL + CANE** This model is the proposed method of SEAL with the CANE [22] embeddings.
- **SEAL + CANE + attributes** This model is the proposed method that learns from enclosed subgraphs, CANE embeddings, and node attributes.

To compare the performance of the proposed method SEAL + CANE with the text-only method TF-IDF in terms of different parts of the available article text. These models are trained on the title, the title and description, and the title, description and full article text on half a year of the data.

4 EXPERIMENTAL SETUP

This section describes how the above methods are used in this study and how the new dataset introduced in section 3.1 is pre-processed to be used in our research.

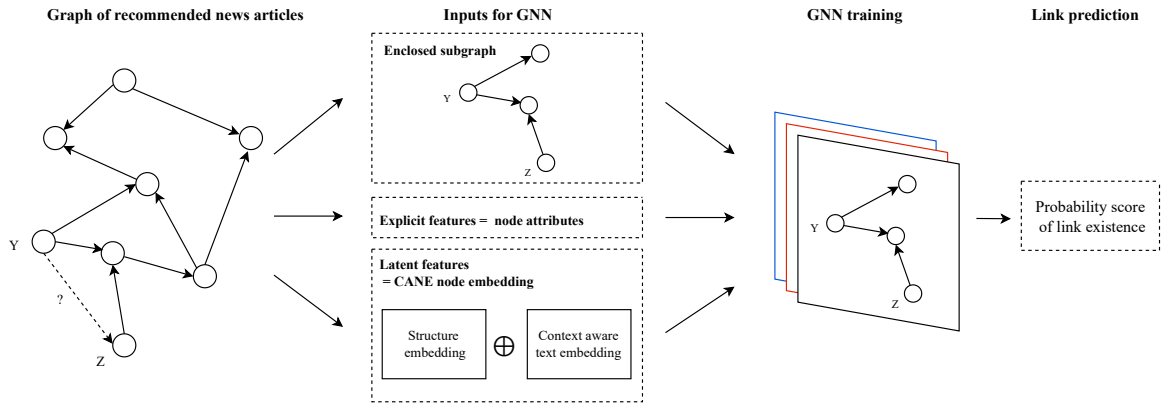

Figure 2: Overview of the new approach for link prediction using textual data in the form of node embeddings

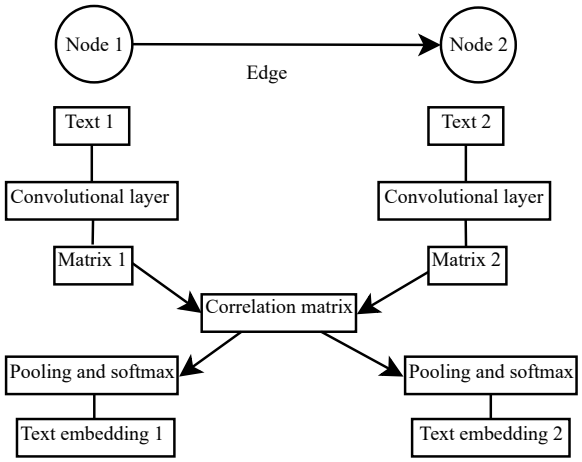

Figure 3: Overview of the process of creating the Context-Aware Network Embedding

4.1 Data Preparation and Pre-processing

In order to use the NOS datasets for this research, some pre-processing steps are taken. To use this data for a link prediction, only articles that have one or more recommended articles are used. This resulted in a total of 208,610 links and 131,887 news articles. The models described in section 3.2.2 are tested on one month (December 2022) of data, and on all filtered data. TF-IDF and SEAL + CANE are evaluated six months of data (January 2022 to the end of June 2022). The articles are filtered within the selected time frame that have at least one recommendation to an article also within the selected time frame. Links are also filtered if both the `parent_article` and `child_article` articles are in the selected time frame. The title, description and text for the CANE embeddings is prepared in the same way as the text in the original research [22]. We tokenized the text into words, removed all punctuation and accents, and lower-cased each word. For TF-IDF, the title, description and text is also pre-processed but in a different way than for the CANE embeddings. It is tokenized into words, punctuation is removed, all words are lower-cased, stopwords are removed and the Porter stemmer [15] is used to stem the words.

	Train	Test	Val	Tot. links	Tot. articles
One month	1233	155	154	1542	1535
Half a year	10988	1373	1373	13735	7866
All	166888	20861	20861	208610	131887

Table 2: Overview of the number of links and total number of articles of the split data per selected time frame

4.1.1 Validation and Test set. The links are split into a training, validation, and test set. As in the implementation of Zhung and Chen [32], a split of 80% training, 10% validation, and 10% test is used. All these different splits of the dataset must have negative examples for training. This is done by the model itself, which randomly selects the same number of non-existent links. An overview of the number of links for each selected time frame used in this research is given in Table 2. It also shows the number of corresponding articles, which provide the text for the CANE embeddings and the metadata for the node attributes.

4.1.2 Data Preparation of Explicit Features. One of the three different types of input for the GNN to train on is the explicit features, which must be given in a continuous or discrete vector [31]. Any kind of side information about the nodes can be used, except their structure. The node attributes used are `published_time`, `owner`, `system_tag`, and `subcategory_list`. The `published_time` is converted to a vector by assigning a numeric value to each time interval (year, month, day, hour, minutes, seconds), this still captures the natural order of the timestamps. The variable `owner` is one-hot encoded into a discrete vector, the attribute consists of seven different categories. The variable `system_tag` is one-hot encoded in the same way as the variable `owner`, there are 27 different system_tags, and most articles do not contain any. The `subcategory_list` attribute is multi-hot encoded. This attribute consists of 109 different categories, where it is also possible for an article to have multiple subcategories. Only the top twenty subcategories are used to keep

the vector small, as the other categories are assigned less frequently (see Figure 8).

4.2 Experiment

The code for the SEAL implementation and the code for the CANE implementation are available on GitHub. In this subsection, all settings of the experiments and the calculation of the evaluation metrics are described. All models except for TF-IDF are trained five times with different random initializations.

4.2.1 SEAL. Zhung and Chen explained that SEAL is flexible with which GNN method to use. As in their implementation, the Deep Graph Convolutional Neural Network (DGCNN) architecture [32] is used as GNN. We used the default setting of DGCNN, which consists of four convolutional layers of 32, 32, 32, 1 channels, a SortPooling layer, two 1-D convolutional layers (16 and 32 output channels), and a dense layer (128 neurons) [32]. The parameters are set to the values given in the original research. The DCGNN is trained for 50 epochs, and the model with the lowest loss for the validation set is selected to predict the scores for the links in the test set. The batch size is 50. The Adam optimizer is used with a learning rate of 1e-4. An important hyperparameter in the SEAL framework is the number of hops (h) [31]. This number selects the enclosed subgraph around the target nodes. In the SEAL framework, h is only selected from {1,2} because it has been verified that these subgraphs already contain most of the useful information. The selection procedure for the number of hops is as follows: if the second-order heuristic Adamic-Adar [1] is better than the first-order heuristic CN [13] on the validation set, then $h = 2$ is chosen, otherwise $h = 1$. For the methods using Node2vec embeddings, the parameters are set as follows. An embedding size of 128 and a window size of 10. The number of walks per node is 10 and the walk length is 80.

4.2.2 CANE. The hyperparameters used for the CANE embeddings are almost identical to those used in the original CANE research [22]. These are the Adam optimizer with a learning rate of 1e-3, a batch size of 64, an embedding size of 200, and a number of epochs of 200. Due to time and computational constraints, the SEAL + CANE and SEAL + CANE + attributes models for all data were trained with 50 epochs. The max_len parameter is changed according to the maximum length of a document in terms of tokenized words. When trained on title only, the value is 20; when trained on title + description, the value is 35; and when trained on title + description + article text, the value is 1944.

4.3 Evaluation Metrics

Some evaluation metrics are chosen to evaluate the models in section 3.2.2. Two standard metrics are often used to quantify the accuracy of a predictive link model, these are the Area Under the Curve (AUC) and precision [12]. The AUC can be calculated from the area under the ROC-curve, which plots the True Positive Rate (TPR, equation 1) against the False Positive Rate (FPR, equation 2) at different thresholds. The AUC can be seen as the probability that a randomly selected missing link is more likely to appear than a randomly selected non-existent link. Precision represents the proportion of correctly predicted links among all predicted links. However, it can be a limited evaluation metric in terms of the dataset. The dataset of hand-labeled links contains possible recommended articles. This does not directly imply that these are the only correct articles. False positives could also be good recommendations. A metric that does not directly penalize false positives is recall. Recall is also a common metric in link prediction and is used when the goal is to minimize false negatives [2, 21]. The formula for recall is the same as for TPR and is written in equation 1. The number of true positives is divided by the number of true positives plus the number of false negatives. The recall is the number of correct positive link predictions as a percentage of the number of positive links. [6].

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{TP + FN} \tag{2}$$

To evaluate the performance of the models as a recommendation system, a top-k recommendation is made and evaluated by recall@k. The formula is given in equation 3. The number of correctly predicted links in the top k is divided by the total number of positives (TP+FN).

$$recall@k = \frac{TP@k}{TP + FN} \tag{3}$$

The models are evaluated in two different ways. The first is to evaluate the models in terms of link prediction. The test set contains only positive links and the model outputs scores for these links. The recall is calculated from these using different values as thresholds, where scores above this value are considered a positively predicted link. The AUC is calculated from the positive links in the test set and the sampled negative links from the model. The second setup is to evaluate the performance of the proposed model as a recommendation system using recall@k. To select the top-k recommended articles, a score is predicted for each article to every other article.

5 RESULTS

In this section, the results of our experiments are displayed. The average scores for AUC and recall are shown with their standard deviation. The average running time is also shown. The AUC values are multiplied by 100. Recall is calculated for three different thresholds (0.5, 0.7, and 0.9) for considering a link as a predicted positive link. For methods using CANE embeddings, the title, description, and article text are used unless stated otherwise.

5.1 Models

Displayed in Table 3, the results of the test set with one month of the data. All models had a higher AUC than the simple random predictor baseline. It can be observed that the non-GNN-baseline method TF-IDF retrieves the highest AUC with an AUC of 98.12. The second highest AUC score (AUC = 87.96) is obtained by the SEAL method in combination with the CANE embeddings. The SEAL method with Node2vec embeddings performed only slightly better than SEAL (subgraphs only). The introduction of CANE embeddings, which replace the structure-only Node2vec embeddings in the SEAL framework, led to a significant increase in performance. Extending the models of SEAL + embedding with node attributes did not seem to provide any benefits. The AUC was lower and the standard deviation was higher for the same model without attributes. The

Method	AUC	recall (0.5)	recall (0.7)	recall (0.9)	avg. running time (hh:mm:ss)
Random predictor	51.20 ± 0.02	0.49 ± 0.03	0.30 ± 0.03	0.09 ± 0.01	00:00:01
TF-IDF	98.12	0.11	0.01	0.00	00:00:07
SEAL (subgraphs only)	81.20 ± 1.31	0.65 ± 0.06	0.50 ± 0.05	0.50 ± 0.05	00:00:35
SEAL + Node2vec	83.02 ± 1.24	0.68 ± 0.10	0.61 ± 0.09	0.50 ± 0.04	00:01:17
SEAL + Node2vec + attributes	69.89 ± 4.37	0.74 ± 0.06	0.07 ± 0.07	0.00 ± 0.00	00:01:22
SEAL + CANE	87.96 ± 1.69	0.64 ± 0.08	0.54 ± 0.07	0.48 ± 0.06	00:25:33
SEAL + CANE + attributes	69.85 ± 9.16	0.63 ± 0.23	0.16 ± 0.11	0.00 ± 0.00	00:28:04

Table 3: Performance of the various models on one month of the data. AUC and recall at different thresholds are averaged over five runs and accompanied by their standard deviation. Also displayed is the mean duration of the running sessions.

SEAL method with CANE embeddings had the highest recall score with a threshold of 0.5 and 0.7. The SEAL method (subgraphs only) had the highest recall with a threshold of 0.9. The recall scores for the TF-IDF model were much lower than for the link prediction models, and the recall scores for the models that use node attributes were also lower than for the link prediction models that do not use node attributes. Figure 9 in Appendix C shows the corresponding distributions of the test scores of all models from Table 3. The TF-IDF model and the models with attributes have relatively lower scores than the other models, which showed more link prediction scores of 1.0 on the test links.

Table 4 shows the results of the same various models on all data. The results are comparable to those obtained using one month of the data. With the highest AUC score for TF-IDF (AUC = 99.44) and the second highest score for SEAL + CANE (AUC = 89.09). It is observed that introducing attributes in the SEAL + Node2vec method led to an increase in performance in terms of AUC compared to the same model without attributes. However, the use of attributes in the SEAL + CANE model decreased the performance.

5.2 Text Data Analysis

The AUC and recall values presented in Table 5 shows the results for TF-IDF and SEAL + CANE using different parts of the available textual data. Both models were trained only on the title, title + description, and title + description + full article text. The AUC for both models was the highest when using the title, description, and full article text. TF-IDF had an AUC value of 98.72 and SEAL + CANE had an AUC of 93.62. When only the title was considered, the AUC for SEAL + CANE was much higher than for TF-IDF. When also using the description, the AUC was slightly higher for SEAL + CANE than for TF-IDF. The addition of training with more text seemed to have improved performance more effectively on TF-IDF than on SEAL + CANE. This was observed while the running time for TF-IDF remained low, whereas for SEAL + CANE it increased significantly as more text was added.

5.3 Link Prediction as Recommendation System

For each model, we extracted the top k scores representing the recommended articles for values of k between 1-50. The mean recall@k scores of all articles in one month of the data were calculated and shown in Figure 4. The highest recall@k values for every value between 1 and 50 was found for TF-IDF. The second best model was SEAL with CANE embeddings, which had a much better performance than all the other link prediction models. The titles of the top-5 recommended articles for each method for the article shown in Appendix A can be found in Appendix D.

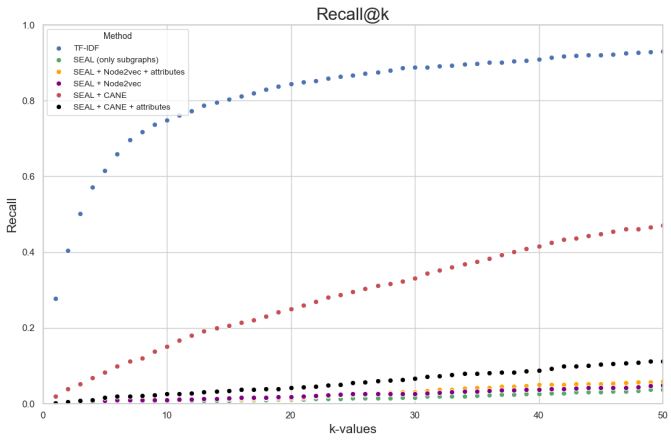

Figure 4: Recall@k for k between 1-50 for different methods on one month of the data. The mean over all articles is shown.

5.4 Articles Outside the Network

Articles outside the network have no links and thus no ground truth. Therefore, distributions of scores are presented instead of standard evaluation metrics. In the data of one month, twenty random articles inside the network (with one or more links) and twenty random articles outside the network (without a link) are selected. From these articles, scores are predicted for every other article within the one-month dataset. Figures 5 (SEAL + Node2vec), 6 (SEAL + CANE) and Figure 10 in Appendix E (SEAL (subgraphs only), SEAL + Node2vec + attributes, and SEAL + CANE + attributes) show the distributions of the scores of the two sets of randomly selected articles to all other articles. It can be seen that for the SEAL + Node2vec model, the scores of articles outside the network do not behave completely differently from the nodes inside the network. Most of the scores for the SEAL + CANE model are relatively lower than for the SEAL + Node2vec model, where the scores for articles outside the network are even lower than the scores for nodes inside the network for the SEAL + CANE model. Figure 10 shows that the scores for SEAL (subgraphs only) for both articles with links and

Method	AUC	recall (0.5)	recall (0.7)	recall (0.9)	avg. running time (hh:mm:ss)
Random predictor	50.16 ± 0.21	0.50 ± 0.00	0.30 ± 0.00	0.10 ± 0.00	00:00:15
TF-IDF	99.44	0.14	0.01	0.00	00:14:27
SEAL (subgraphs only)	80.18 ± 0.65	0.57 ± 0.01	0.48 ± 0.01	0.48 ± 0.01	00:54:53
SEAL + Node2vec	81.39 ± 1.25	0.49 ± 0.01	0.48 ± 0.01	0.47 ± 0.00	01:49:34
SEAL + Node2vec + attributes	85.37 ± 7.82	0.71 ± 0.17	0.57 ± 0.11	0.45 ± 0.02	01:54:27
SEAL + CANE	89.09 ± 1.84	0.64 ± 0.13	0.50 ± 0.08	0.42 ± 0.04	20:37:51
SEAL + CANE + attributes	79.62 ± 5.63	0.56 ± 0.19	0.46 ± 0.09	0.44 ± 0.05	23:36:53

Table 4: Performance of the various models on all data. AUC and recall at different thresholds are averaged over five runs and accompanied by their standard deviation. Also displayed is the mean duration of the running sessions.

	Method	AUC	recall (0.5)	recall (0.7)	recall (0.9)	avg. running time (hh:mm:ss)
TF-IDF	title	77.22	0.01	0.00	0.00	00:00:01
	title + description	88.48	0.01	0.00	0.00	00:00:02
	title + description + text	98.72	0.12	0.00	0.00	00:00:07
SEAL + CANE	title	91.84 ± 1.03	0.69 ± 0.06	0.63 ± 0.07	0.55 ± 0.07	00:11:48
	title + description	93.11 ± 0.89	0.68 ± 0.07	0.63 ± 0.07	0.55 ± 0.08	00:13:59
	title + description + text	93.62 ± 0.75	0.59 ± 0.15	0.54 ± 0.16	0.39 ± 0.15	04:30:22

Table 5: Model performance of TF-IDF and SEAL + CANE on training on title, title + description and title + description + full text. AUC and recall at different thresholds are averaged over five runs and accompanied by their standard deviation. Also displayed is the mean duration of the running sessions.

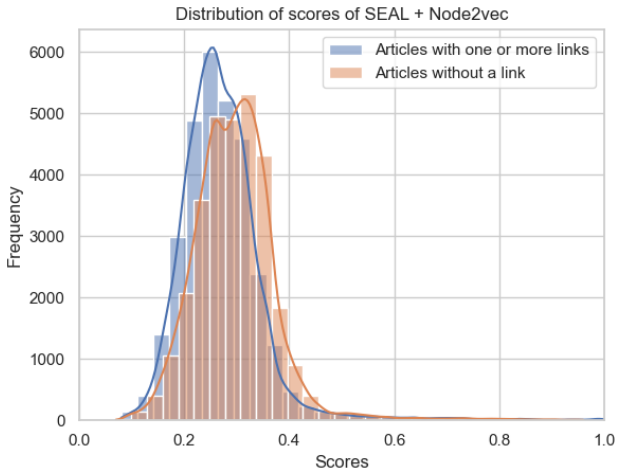

Figure 5: Distributions of scores on SEAL + Node2vec embeddings with articles inside the network and articles outside the network in one month of the data, the scores are from twenty randomly selected articles in each distribution to all other articles in the data.

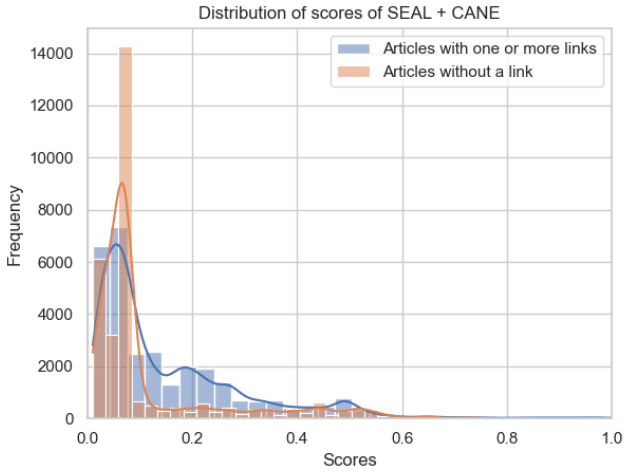

Figure 6: Distributions of scores on SEAL + CANE embeddings with articles inside the network and articles outside the network in one month of the data, the scores are from twenty randomly selected articles in each distribution to all other articles in the data.

without links to nodes in the network have close scores, while the models with attributes have scores that are more spread out.

6 DISCUSSION

6.1 Performance Comparison

Comparing the performance of the proposed method with the state-of-the-art models provides valuable insights into the effectiveness of link prediction. SEAL + CANE (AUC = 87.96 and 89.09) outperforms the baseline SEAL + Node2vec (AUC = 83.02 and 81.39) and SEAL (subgraphs only) (AUC = 81.20 and 80.18), highlighting the effectiveness of the CANE embeddings in link prediction. This indicates that the CANE embeddings provides an improved ability to distinguish positive and negative links when using text in the form of node embeddings and outperforms the current state-of-the-art

GNN SEAL framework. This addresses the research gap by incorporating textual data for node embeddings into a GNN-based link prediction model.

However, it is important to note that the results for the baseline SEAL (subgraphs only) are lower than those reported in the original research by Zhang and Chen [31], although they are within the same order of magnitude. Our average AUC for one month is 81.20 and for all data is 80.18. The average AUC of Zhang and Chen varies between 90.30 and 98.85 for the different datasets. A possible explanation for this is that the average node degrees of all datasets used in the SEAL research are higher (2.49 to 27.36) than the average node degree of the NOS dataset (2.17). This discrepancy in dataset characteristics could affect the overall performance.

6.1.1 TF-IDF Baseline Method and Recommending. The TF-IDF baseline method, which is a text-based model, achieves the highest AUC score compared to the different link prediction models, indicating its strong discriminative power in distinguishing positive and negative links. Moreover, TF-IDF outperforms them in terms of recall@k, demonstrating its effectiveness as a recommendation system. One reason for TF-IDF's strong performance could be attributed to the nature of the dataset and the characteristics of the hand-labeled recommendations. In the dataset used, the recommendations were typically made between articles that shared the same topic or event. As a result, words that are specific to the topic or event may occur infrequently in the overall vocabulary. TF-IDF assigns higher weights to such infrequent words, considering them as more important for capturing the relevance between articles on the same topic. Consequently, TF-IDF can successfully match articles with similar subject matter, leading to its superior performance in this context. Looking at the use of link prediction models as recommendation systems, SEAL + CANE outperforms the state-of-the-art GNN models SEAL (subgraphs only) and SEAL + Node2vec. For future work, a potential improvement could be the use of hard negatives during the training process of SEAL + CANE, instead of randomly sampled negatives. By using hard negatives, which are more challenging negative examples, the model can learn to better discriminate between positive and negative links, potentially improving their predictive capabilities.

6.2 Behavior of Unconnected Nodes

We additionally analyzed the behavior of articles without a link in the network. Section 5.4 showed that articles without links received similar scores to those with links when using the SEAL + Node2vec approach. This can be attributed to the assumption of Node2vec embeddings that nodes without a link have similar structural properties as linked nodes [5]. Nodes without links in the network have significantly lower score distributions compared to nodes with links for methods using CANE embeddings. This can be attributed to the use of LINE as the structural embedding method in CANE. LINE constructs node embeddings by considering the local connectivity patterns and similarity of neighborhood structures within the network [20]. However, since unconnected nodes have no neighbors, SEAL + CANE relies solely on the text embedding, leading to lower scores for these nodes. To enhance the model's robustness, future research can explore alternative strategies for handling new nodes without links.

6.3 Attributes

It is possible to use article metadata as node attributes to learn from in the SEAL + Node2vec and SEAL + CANE methods. In both cases, for both time frames, the AUC decreased compared to their model without attributes, except for the SEAL + Node2vec + attributes method on all data. The standard deviations over the different runs were much higher than the models without attributes, indicating more variability. It is worth noting that we used a fixed set of attributes in the training process described in section 4.1.2. This means that the results for the methods with attributes all depend on a single configuration of the metadata of the articles. Therefore, the limitations of the models with attributes should be kept in mind when interpreting these results. Future research can explore the impact of individual attributes, for example through feature selection techniques that identify the most relevant features. Feature engineering can also be considered to enhance the representation of input features. Building on the success of the TF-IDF baseline method, the TF-IDF vector of the article text could be used as a node attribute. In addition, named entity recognition or topic modeling techniques could be explored to further enhance the node attributes in future studies.

6.4 Analysis

The TF-IDF baseline method has a higher AUC but lower recall scores compared to the other methods. Figure 9 in Appendix C shows that the scores for the test links of TF-IDF are lower compared to the SEAL methods, which explains the lower recall scores at different thresholds. The reason for this is that the scores for both frameworks represent something different. The TF-IDF scores are based on text similarity and the scores for the link prediction models represent a probability of link existence.

The results of training on title, title + description, and title + description + full text show the different behaviors of TF-IDF and SEAL + CANE methods. As TF-IDF is a text-only method, the model performance improves as more text is used. SEAL + CANE is a link prediction method that can also learn from existing links and improves relatively less when using more text. Adding more text significantly increases the running time of the SEAL + CANE method, making it less scalable for larger networks. TF-IDF, which solely computes similarities without requiring a training process, exhibits a shorter run time compared to SEAL + CANE, which involves two training processes and consequently leads to a longer running time. As the length of the text or the size of the network increases, the computational cost for generating CANE embeddings in the SEAL + CANE approach becomes more significant. This can lead to longer run times and limit the scalability of the method. The computational demands of generating CANE embeddings are a potential disadvantage when dealing with large amounts of text or large networks.

Overall, the results of the different models are comparable for one month of the data and for the whole dataset. This can be attributed to the fact that most of the links occur between articles published within a week of each other (Figure 7 in Appendix B), indicating that accurate predictions can still be made with a small amount of data from the NOS dataset.

6.5 Dataset and Generalizability

It is important to acknowledge that our study focused solely on the NOS dataset and its generalizability to other text datasets, particularly those in languages other than Dutch, is currently limited. Furthermore, there is room for improvement in the experimental setup, as the links are manually labeled by the NOS editorial team. This may introduce a degree of subjectivity, potentially affecting the recommendations. To mitigate this, future work can include a qualitative evaluation of the recommended articles for the different models, which would provide valuable insights. To increase the reliability and applicability of the results, it is recommended to test the approach on different datasets in different languages. In addition, exploring topics beyond news can contribute to a better understanding of the method's performance in different contexts. This will help to make the research more robust and widely applicable.

7 CONCLUSION

Recent advancements in news recommendation research have highlighted the efficacy of GNN models for link prediction. The subgraph enclosing method SEAL is promising, but has the limitation that it does not include textual information in node embeddings. This research aims to answer to what extent a GNN-based link prediction model can benefit from textual data incorporated in node embeddings in the context of Dutch news articles. In order to answer this, a new method for link prediction is proposed in this research, which is the combination of the state-of-the-art method SEAL for link prediction and CANE for obtaining node embeddings based on textual data. In addition, we introduced a new dataset of Dutch news articles, along with their associated links. We used this dataset to evaluate the proposed method.

The SEAL + CANE method excels in link prediction but falls short in recommending articles compared to the text-only TF-IDF method. The proposed link prediction model performs consistently well in predicting links across different time frames, using the title, description, and full text for optimal results. Additionally, the behavior analysis of unconnected nodes reveals lower scores of link existence compared to connected nodes in the SEAL + CANE framework. In particular, SEAL with CANE embeddings improved performance by 7.7% compared to state-of-the-art SEAL with Node2vec embeddings.

Our integration of textual data into node embeddings enables improved link prediction using GNNs. Future work should validate our findings with diverse datasets and explore improvements in node attributes and strategies for handling unconnected nodes.

REFERENCES

[1] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.

[2] 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.

[3] 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).

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

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

[15] Martin Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems* 14, 3 (1980), 130–137.

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

[17] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12, 9 (2011).

[18] 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).

[19] 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.

[20] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding.. In *WWW*. ACM.

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

[22] 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.

[23] Christian Von Mering, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417, 6887 (2002), 399–403.

[24] 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.

[25] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.

[26] 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.

[27] 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.

[28] 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*.

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

[30] 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.

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

[32] 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.

[33] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. 2020. Revisiting graph neural networks for link prediction. (2020).

Appendix A EXAMPLE NOS ARTICLE AND LINK DATA

An example of an article in the article dataset:

- `article_id`: 2456865
- `article_title`: Griezmann hervindt zichzelf in Frans elftal en speelt dit WK voor twee
  (EN: Griezmann reinvents himself in a French team and plays for two this World Cup)
- `article_description`: De 31-jarige Fransman verkeert in een minder aanvallende rol in grootse vorm in Qatar. Zondag speelt hij de WK-finale tegen Argentinië.
  (EN: In a less attacking role, the 31-year-old Frenchman is in great form in Qatar. On Sunday, he will play the World Cup final against Argentina.)
- `article_text`: Antoine Griezmann blinkt uit dit WK voetbal, in een andere rol dan hij gewend is. In 2014 in Brazilië was de 31-jarige Fransman nog linksbuiten en vier jaar later in Rusland werd hij als scorende schaduwspits wereldkampioen. En in Qatar? Eigenlijk is de flegmatieke aanvaller van weleer overal. (truncated)
  (EN: Antoine Griezmann is excelling this World Cup, in a different role than he is used to. In 2014 in Brazil, the 31-year-old Frenchman was still a left-back and four years later in Russia, he became world champion as a scoring shadow striker. And in Qatar? Actually, the phlegmatic striker is everywhere.)
- `published_at`: 2022-12-18 11:19:45
- `owner`: sport
- `subcategory_list`: [voetbal]
- `system_tag`: wk-voetbal-2022

The corresponding recommended articles of the article above (*article_id: 2456865*) in the links dataset:

- `article_id`:2456843
  `article_title`:Onbegrijpelijk dat we Messi nu als beest zien, zéér teleurgesteld in Van Gaal
- `article_id`:2456794
  `article_title`:Kylian Mbappé is rolmodel in de banlieue waar hij opgroeide
- `article_id`:2456424
  `article_title`:WK-droom Marokko voorbij: Frankrijk wint met 2-0 en gaat naar finale

Appendix B DISTRIBUTIONS OF TIME DIFFERENCES OF LINKED ARTICLES AND SUBCATEGORIES

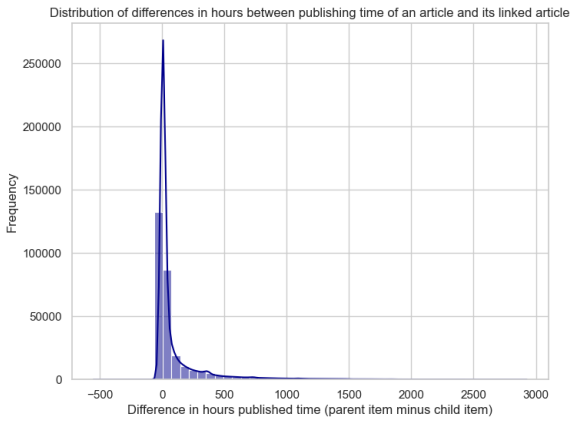

Figure 7: Distribution of differences in hours between the publishing time of an article and its linked article

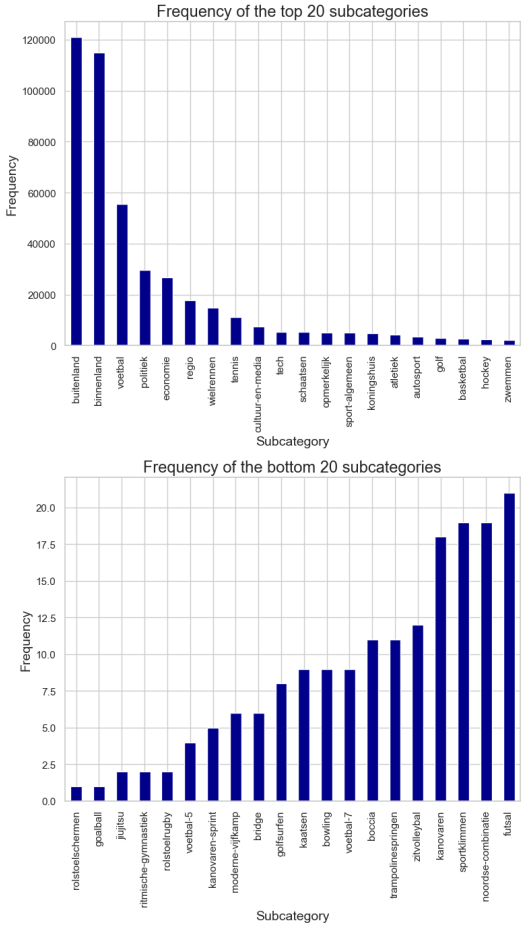

Figure 8: Frequencies of the top and bottom 20 subcategories

Appendix C DISTRIBUTIONS OF THE VARIOUS MODELS OF THE SCORES FROM THE TEST LINKS WITH ONE MONTH OF DATA

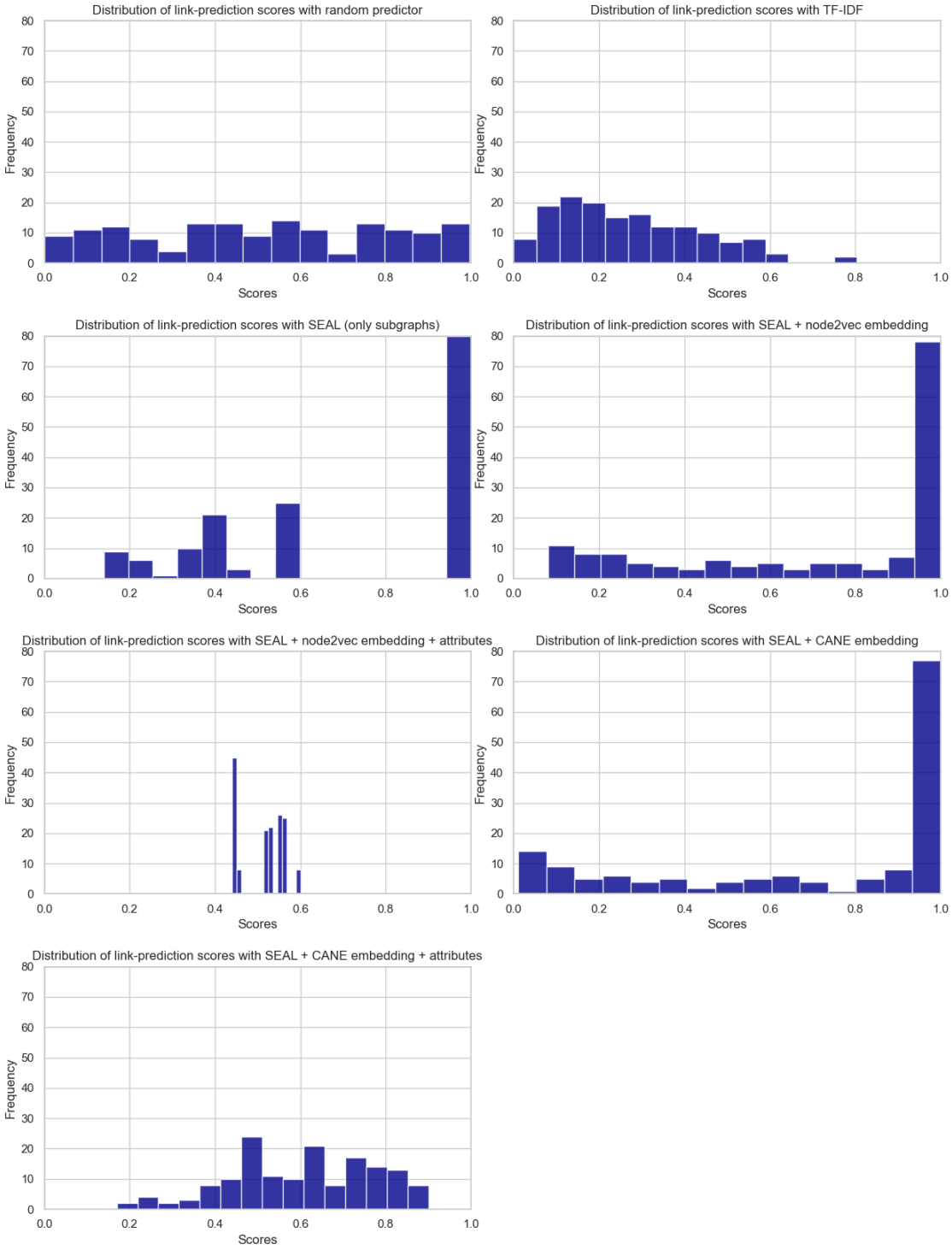

Figure 9: Distributions of the scores on the test set of one month of data for seven different methods

Appendix D EXAMPLES OF TOP-K (K=5) RECOMMENDATIONS FOR ARTICLE 2456865

For each of the following method, the titles of the top 5 recommended articles for the article in appendix A (`article_id` = 2456865) are presented.

TF-IDF

- Gemaskerde Kroaat, Marokkaanse superman en Argentijnse kapitein: de WK-revelaties
- Deschamps wil niet praten over toekomst na verloren WK-finale: 'Deceptie is enorm'
- Deschamps na bereiken WK-finale in illuster rijtje: 'Nog één laatste stap te gaan'
- Frankrijk heeft tegen Engeland het geluk van de kampioen, met Giroud als verlosser
- Deschamps waarschuwt Engeland: 'Frankrijk draait niet alleen om Mbappé'

SEAL

- Wenger en Klinsmann staan achter 'speciale' Van Gaal: 'Weet precies wat hij doet'
- 'All I Want for Christmas' verbreekt Spotify-record
- De Nederlandse jaren van Berhalter: 'Bijna knokken met Verbeek bij Cambuur-Heerenveen'
- Een wonder à la Memphis voor Miedema? 'Ik denk het niet, eigenlijk'
- Met sluiting bedreigde centra voor kinderhartzorg blij met advies NZa

SEAL + Node2vec

- Het wonderkind dat Brazilië van een nationaal trauma afhielp
- Zweden staat voor NAVO-dilemma: hoe diep wil het land buigen voor Turkije?
- Argentijnen Messi, Martínez en Fernández krijgen ook individuele prijzen
- Drukte bij vuurwerkwinkels op eerste verkoopdag
- 'Emotionele' Belgische bondscoach Martínez stapt op na WK-uitschakeling

SEAL + Node2vec + attributes

- Hollandse school? Oranje drie keer in top-5 van onsportiefste WK-duels
- Luna klaar voor de JSF-finale: 'Ik kan niet wachten!'
- 'Wintergolf' van Oekraïense vluchtelingen blijft (vooralsnog) uit
- In beeld: leven in het donker, het dagelijks leven in Oekraïne zonder stroom
- Groenen en jarige Martens slechts invaller bij PSG, dat Real Madrid uitschakelt in CL

SEAL + CANE

- Marokko-spelers balen van arbitrage, maar zijn trots op WK-succes: 'Was geen geluk'
- Feest en onrust in grote steden na WK-winst Marokko, tientallen aanhoudingen
- Ambitieus AZ voert druk op, ook richting coach Jansen: 'Contract geen done deal'
- Grote steden voorbereid op WK-duel van Marokko tegen Canada
- Kwetsbaar Duitsland kan ondanks zege op Costa Rica zijn koffers pakken

SEAL + CANE + attributes

- Wenger en Klinsmann staan achter 'speciale' Van Gaal: 'Weet precies wat hij doet'
- Dumfries doet als Cruijff en geeft Van Gaal weer wingbackwapen
- Feest Marokkaanse fans loopt in verschillende steden opnieuw uit de hand
- Dagboek Doha: hoe groots Amerika is in verliezen, weet deze vriend van Berhalter
- Van der Poel wint modderworstelen Grote Drie, maar Van Aert sprint naar winst

Appendix E DISTRIBUTIONS OF SCORES OF THREE METHODS FROM ARTICLES WITH LINKS AND WITHOUT LINKS IN THE NETWORK

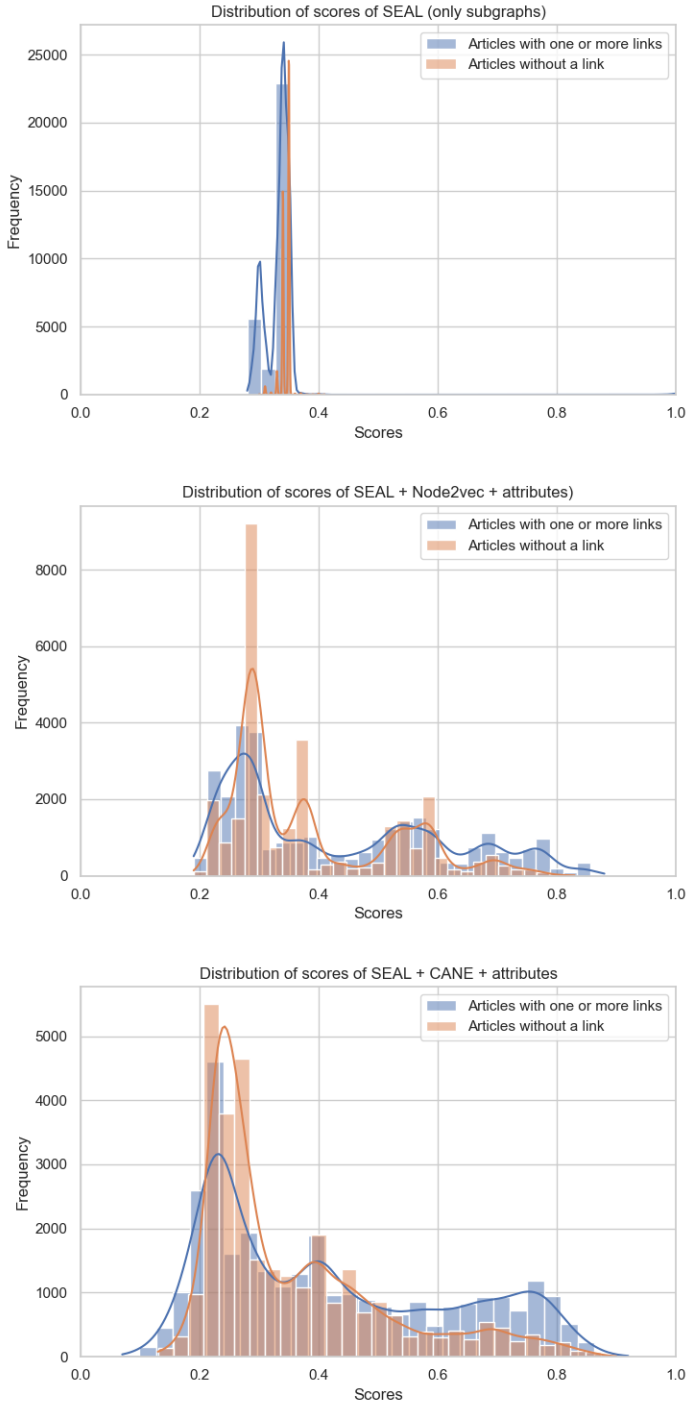

Figure 10: Distributions of the scores of three different methods of twenty random articles with one or more links to other articles and twenty random articles without any links to other articles in one month of the data. The scores are predicted for each article to each other article.