A Thesis on News Recommendation

Abhik Naskar[1], Sudeshna Sarkar[2]

Scholar, Bachelor + Master of Technology, Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur

Professor, Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur

Abstract: ***An integral part of any individualised news service is the news recommendation technique. Research on news recommendation is much more limited than that on product and movie recommendations, primarily due to the absence of a benchmark dataset of sufficiently high quality. In this paper, we introduce a massive dataset for news recommendation called MIND. MIND is built from Microsoft News user click logs and contains over 160k English news articles with rich textual content like titles, abstracts, and body paragraphs. We show that MIND is a useful testbed for news recommendation by comparing several cutting-edge methods that were developed on various private datasets. Our findings demonstrate that the success of news recommendation is heavily dependent on the accuracy with which content is understood and user interests are modelled. Several NLP techniques, including efficient text representation methods and pre-trained language models, have been shown to significantly boost news recommendation performance. Because the internet allows access to news stories from millions of sources throughout the world, online news reading has grown in popularity. The ability of news websites to assist readers in finding stories that are interesting to read is a major difficulty. We share our work on constructing a personalised news recommendation engine in the Microsoft News Dataset in this thesis (MIND). The recommendation engine creates profiles of users' news interests based on their prior click activity for users who are signed in and have expressly enabled web history. To better understand how users' news interests vary over time, we employ semantic analysis to anticipate users' current news interests based on their behaviours and the news trend seen across all users' activities. Moreover to capture the knowledge aware concept of different news, their clicked behaviours etc. we also construct knowledge graphs to capture that information and generate personalised news recommendations. To access the MIND dataset, visit https://msnews.github.io.***

Keywords: Content-Based Filtering, Collaborative Filtering , News Recommendation Section, Pre-trained Language Model-empowered News Recommendation, News encoder, Microsoft News Dataset, MIND

1. Introduction

In this time of information consume of data has increased significantly in contemporary years. In modern society people are inundated with data, which makes it difficult to choose from a huge number of possibilities. Recommender systems determine the desired product to be offered to the consumer so that they may evaluate and pick the required item [Ricci et al., 2011].

Since introduction of online commercial platforms like Flipkart, Amazon, the importance of recommender systems has become increasingly apparent. In 2006, the relevance of recommender systems was also highlighted. Netflix, a worldwide supplier of TV shows and movies, held a worldwide competition in 2006 in order forecast ratings given by user for movies depending on ratings given prior without knowing anything about the films or the users. The prize was awarded depending on how well the algorithm developed by Netflix performed. The responsible team eventually winning the competition boosted Netflix's algorithm by roughly 10% [Ricci et al., 2011], thus awarded a prize money of 1 million US$.

A recommender system comprises of:
- **Users:** Users are persons in the system who have preferences for things and who can also be a data source [Ricci et al., 2011]. Every one of the users may have a collection of user attributes; for example, demography (age, gender, and so on) is a user attribute. A model may be inferred for each user. For instance, their preferred film genre or the types of literature they (the users) enjoy reading.
- **Items**: Items are the products that the algorithm chooses to recommend [Ricci et al., 2011]. We may have a collection of item characteristics or properties for each item. An actor in a film, the author of a news piece, or the colour of an appliance, for example.
- **Preferences**: These are the preferences of users [Ricci et al., 2011]. A user can score a movie 5 on a scale of 5 stars if they encounter it in the preferences space.

The components outlined above are employed in a variety of ways in various algorithms. The following types of algorithms can be broadly classified:
- **Non-personalised systems**: It includes summary of the statistics and also in certain cases, association with products [Poriya et al., 2014] based on community information, such as the best-selling, most significant, or hot and trending product. It might also offer a brief about the community ratings, such as how popular a dine-out is among the general public, or a general brief about the ratings by community that is converted in the list format, such as the particular hotel in town is the best.
- **Content Based Filtering:** The rating of things by users, and a model of preferences by the user based on element characteristics is created as a result. A good example would be regarding the field of film. Assume a person is liking scientific fiction, fantasies, and action films but dislikes romance films. Over time, the programme may compile this information and determine that the person has high ratings for scientific fiction, fantasies, and action films, but low levels for romantic movies. The computer may discover this thing the user loves or hates certain actors. For example, the user may enjoy films starring Bruce Willis but not those starring Ben Stiller. This information is used in content-based filtering to connect user ratings to product attributes - in this case, movie attributes. Because we're aiming to construct a model based on user comments on news stories, our

technique falls within this sort of algorithm. Content Based Filtering is well-represented in Figure 1.1.

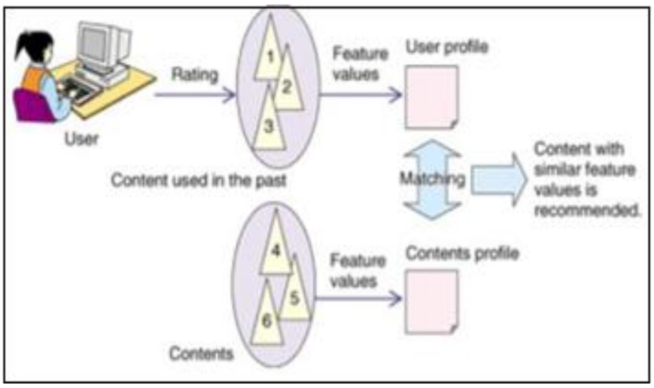

Figure 1.1: Content based Filtering

Collaborative Filtering: In collaborative filtering, rather than attribute data, user evaluations of other individuals are utilised to forecast and suggest [Ricci et al., 2011]. The collaborative filtering is based of concept of a model by user, which is collection of reviews, and a model of the item, which is also a collection of ratings. When the two models are combined, we get a matrix of ratings which is sparse, with the majority empty and the few cells are filled. Thus, the first goal is to mark the empty slots or estimate the rating, the next thing is to select a full cell or propose an item. Figure 1.2 gives an excellent summary of Collaborative Filtering.

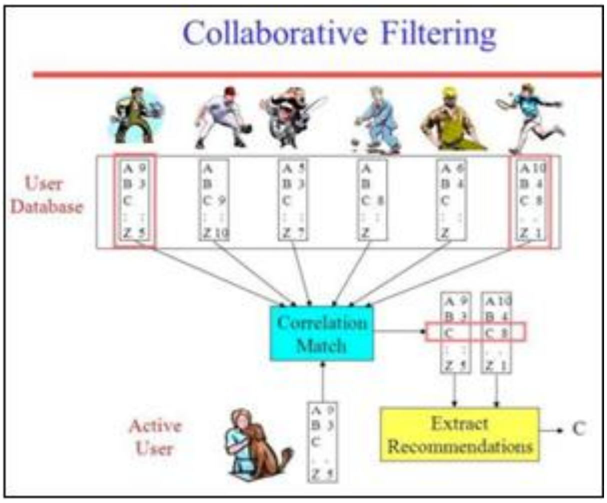

Figure 1.2: Collaborative Filtering.

The fundamental distinction between collaborative filtering and content based filtering is collaborative filtering recommends new things based on the requirements of users (the users who have likewise tastes of particular products), whereas Content Based Filtering does not.

2. Problem Definition and Thesis Objective

Public have a preference for news which are printed sources such as newspapers and magazines. In the internet age, however, this reality changes dramatically; users are bombarded with information from many sources, and it is very unusual for users to switch among portals for news or news read from forums that combine news pieces among several sources. With so much information available, it's tough to choose which news story a consumer would enjoy. As a result, people either discontinue or reduce their news intake.

Figure 2.1: The New York Times' recommendation section

Recommender systems can be used to solve the problem involving overloading of information (problems faced in making a decision due to availability of extra information). These materials rely on a variety of user input to reduce the number of alternatives available to the user to a manageable quantity. Because the user is not requested to offer implicit input, one can experiment with a variety of options. One example of implicit feedback is the user click history, in which the user's clicked news articles at a certain timestamp may be utilised to propose the next news article that can be provided to the user, increasing the likelihood that the news will be clicked.

The purpose of this thesis is to look at the influence of characteristics taken from feedback classes' (categories to which news articles belong, such as sports, politics, and so on) on recommendations and other features. For different themes in the dataset, we employed multiple deep learning models and attention mechanisms to achieve this aim. We also suggested a novel architecture consisting of pre-trained language models and knowledge graphs to recommend news items to users from a subset of a large dataset based on the user's clicked history. Finally, we compared the performance of our novel design to other current models which are state-of-art.

3. Related Work

Therefore their high skill in text modelling, pre-trained language models (PLMs) have had a lot of success in NLP. PLMs are often trained earlier on a corpus which are large and not labelled using supervision by itself to encode text information, as opposed to standard models that are not unusually explicitly trained by labelled data which are specialised tasks. As a result, PLMs are often a superior starting point for fine adjustment in downstream operations. Furthermore, unlike many classic NLP approaches that use shallow models, our method uses deep models. PLMs are often substantially more complex, having a large number of variables. The base model of BERT, for example, has 109M parameters and 12 Transformer layers.

Personalised news recommendations can improve a user's reading experience significantly. To develop news and user representations, traditional news recommendation techniques mostly rely on manual feature engineering. Deep learning-based approaches and pre-trained language models have recently gotten a lot of attention and have shown to be more effective. Pre-trained language models are used to model news, which is then fine-tuned using the news recommendation task. These approaches, on the other hand, simply employ implicit information on various texts to learn appropriate presentations that might not be enough modelling consumers and news. Unlike previous approaches, ours focus on entities which are explicit in user modelling to understand various links between news and entities. It combines relational entity graph data with textual data. In the meantime, with advancement in the field deep learning, neural networks and graph neural networks (RGCNN) have become employed in the natural language processing tasks, such as categorization of text, task for machine-translation, and question- answering in news recommendation [3].

Above two methods are implemented on a real world dataset, MIND, to achieve a great performance on news recommendation than existing methods and their performances are compared against each other.

4. Problem Definition

The following is an illustration of the news suggestion problem in our article. The model seeks to forecast whether the user u would click the candidate news given a candidate news d_c and a user u with Q clicked news d_h. Then, depending on the ordering of news-user pair scores, suggestions are made.

5. Methodology

5.1 Model 1: News Recommendation using CNN

In the following section, we bring in the details of CNN-empowered recommendation of news. We discuss how we incorporated convolutions into this model to enhance the news recommendation modelling.

5.1.1 CNN Empowered News Recommendation

We discuss the model of text classification using convolutional neural networks as shown in figure 5.1.

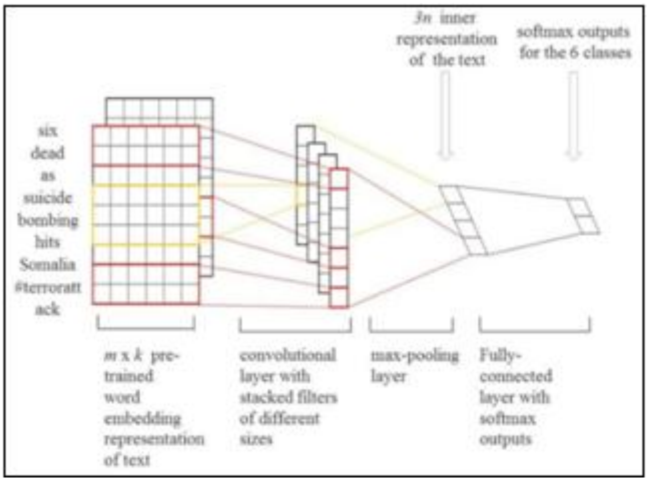

Figure 5.1: CNN Architecture for text classification

The news encoder is initialised with a word embedding which is trained in an earlier model to capture the various contexts involving neighbouring words and phrases in news texts and a convolutional network to get the output of classification.

Input: News tokens for each news consisting of title, abstract and body for each news.

Processing:
Step1: We use a dictionary based embedding to convert each token into its index based representation. When we dot product of text vectors, they may return zero even though they belong to the same class, but if we dot product of those embedded word vectors to identify similarity between them, we may find the interrelation of words for a given class.

Step2: The Max Pooling layer then slides the filter/ kernel across these embeddings to discover convolutions, which are then further dimensionally reduced to minimise complexity and processing. Finally, the completely linked layers and the activation function on the outputs will provide values for each class.

5.1.2 Primary Task

We must estimate if a user would click a candidate news Dc given a set of previously clicked news D = [D1, D2,..., Dr], which were previously clicked by users for a particular timestamp or impression log, and a set of previously clicked news D = [D1, D2,..., Dr].

5.1.3 Model Training

For a particle candidate news, we collect all the news information of the previously clicked news i.e. title, body and abstract for the previously clicked news and convert the texts into embeddings and thereby pass through a cnn filter to perform classification. By categorising clicking the particular candidate news, we utilise cross entropy loss function to train the model. We fine tune parameters of the recommendation model and CNNs for the news recommendation job by optimising the back-propagation of the loss function.

5.2 Model 2: News Recommendation with Pre-trained Language Models

We will go through the specifics of PLM-powered news recommendations in this part. We go through how we used PLMs to empower news modelling in this framework.

5.2.1 News Recommendation Empowered by PLM

We discuss the model for PLM empowered news recommendation as depicted in Fig 5.2.

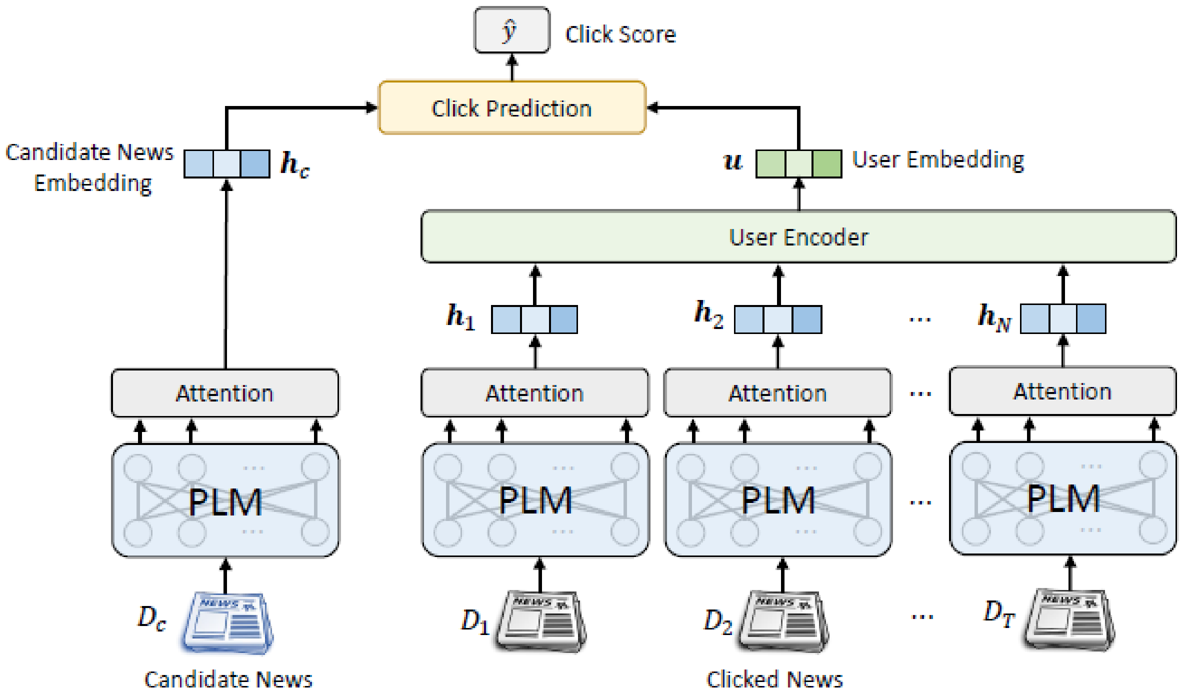

Figure 5.2: The Framework of PLM empowered news recommendation

To get hold of the deep contexts of textual news, the news encoder is started with a pre-trained language model and then we apply an attention network to get the output of PLM.

Input: [w1, w2,......, wm] are news texts with M tokens apiece.

Processing:

Step1: Through numerous transformer layers, the PLM turns these tokens into embeddings and learns hidden representations of these words.

Step2: To combine these concealed token representations into a single news embedding, an attention network is utilised.

5.2.2 Primary Task

Given a set of clicked news D = [D1, D2,…, Dr] which are clicked by users previously for a given timestamp or impression log, and a candidate news Dc, we have to evaluate if the news would be clicked by the user or not.

5.2.3 Model Training

Using news impression records to create samples to be tagged, we employ negative sampling techniques. By categorising the candidate news to be clicked, we utilise the cross entropy loss function to train the model. The parameters of the recommendation model and PLMs may be fine-tuned for the dedicated job by optimising the back-propagation of the loss function.

5.3 Model 3: Entity Graph and PLM for News Recommendation

Existing techniques typically model potential news based on the content of the text and conclusion of the interest of the user based on the history of news clicked by the user. We learned news representation via a word-level application which is personalised for attention networks and representing interest of users via a user-level personalised attention network, not dependent on each other, as in the previous model, and like previous model we used the inner product of interest of user representation and representation of candidate news to perform matching of interest.

The candidate news story, on the other hand, may have several characteristics and elements, and the user might be having various interests. As a result, independent interest of the user modelling might be weaker for matching of interest.

As a result, we suggest using graph convolutional neural networks (RGCNN) to represent user interests across various news and entities explicitly. To improve user representation, we mix text information and graph entities explicitly derived from pre-trained language models. We ran extensive tests on a subset of a real-world dataset called the Microsoft News Dataset (MIND).

5.3.1 Architecture

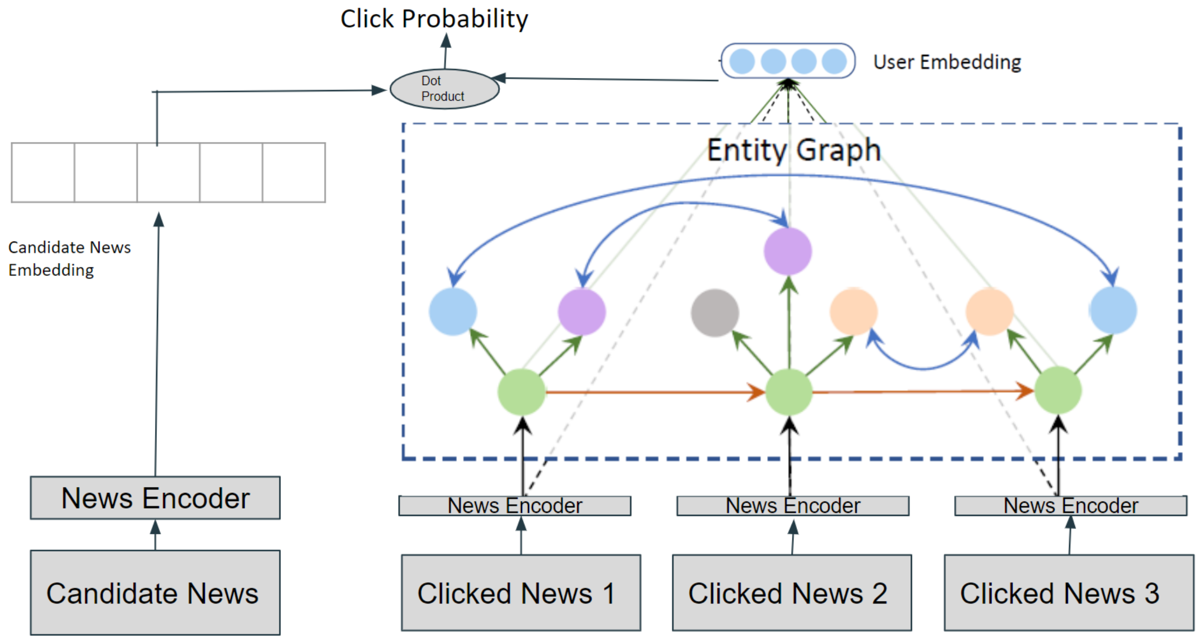

Figure 5.3: The framework of our model. In the graph entities, green nodes denote news. Nodes in other colours are represented by entities

Step1: We begin by calculating implicit text representations based on several news attributes.
Step2: The graph entities and textual information are then reasoned using RGCNN.
Step3: Finally we predict the probability of clicking the particular news according to representations of news and user, integrating the results of the first two stages.

We will discuss above steps in details:

5.3.2 Text representation

This information may be gleaned immediately from the body, title and abstract of each news item. These semantic characteristics are extracted using the "News Encoder." The first layer consists of language models which are trained earlier that take the title and abstract text of the clicked news as input and produce a distributed representation of each word in the text. Then, on top of that, we employ multi-head attention and additive attention to generate a title and abstract sequence that may be expressed as $[w_1^t, w_2^t, \ldots, w_M^t]$, $[w_1^a, w_2^a, \ldots, w_N^a]$ respectively. Figure 5.3.1 depicts the design of a news encoder.

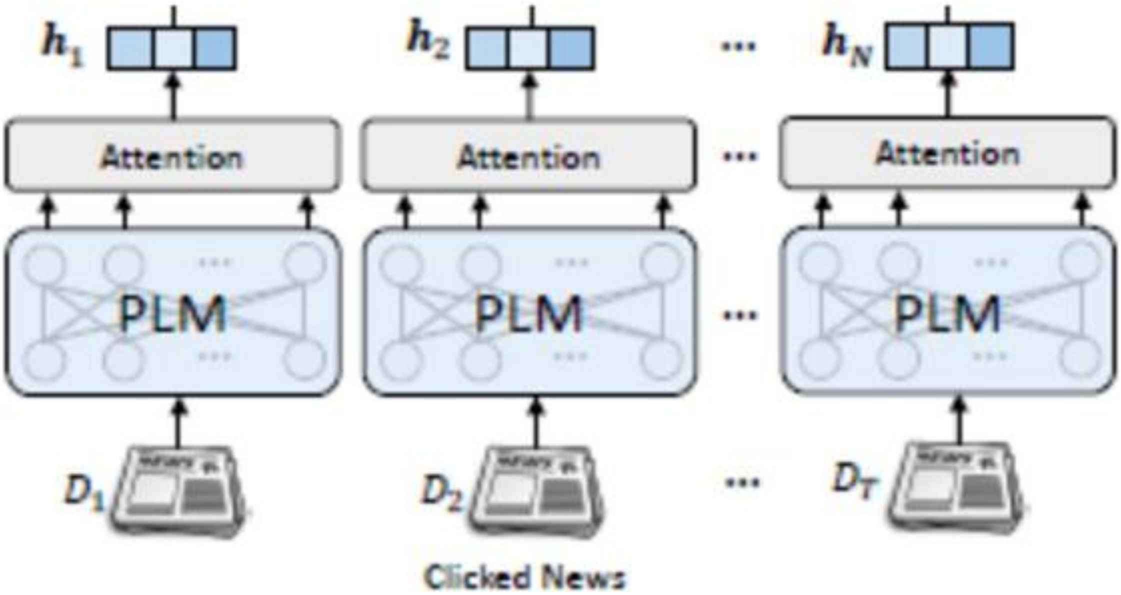

Figure 5.3.1: The framework of the News Encoder Model

We can represent the formula for each word in title, body and abstract as follows:

$$c_i^t = \mathrm{q}_t^T \tanh(\mathrm{V}_t \times \mathrm{w}_i^t + \mathrm{v}_t),$$
$$\alpha_i^t = \frac{e^{c_i^t}}{\sum_{j=1}^{M} e^{c_j^t}} \quad (1)$$

$$c_i^a = \mathrm{q}_a^T \tanh(\mathrm{V}_a \times \mathrm{w}_i^a + \mathrm{v}_a),$$
$$\alpha_i^a = \frac{e^{c_i^a}}{\sum_{j=1}^{N} e^{c_j^a}} \quad (2)$$

Where q_t, q_a, q_b signify the query vector for attention that has to be learnt, and V_t, V_a, v_t, v_b denote the projection parameters. Densely represented title, abstract, and body (d_t and d_a) may then be obtained by cumulating the series of word embedding:

$$\mathrm{d}_t = \sum_{i=1}^{M} \alpha_i^t \mathrm{w}_i^t,$$
$$\mathrm{d}_a = \sum_{i=1}^{N} \alpha_i^a \mathrm{w}_i^a, \quad (4)$$

After that, we get the representation of news by combining the abstract and title representations and running them through a linear layer neural network. This news representation will be fed into the entity graph explained in the following section.

5.3.3 Entity Graph Construction

A graph is made up of nodes (V) and edges (E) (E). GNN = G (V, E). News and entities are the nodes, whereas news-news, news-entities, and entities-entities are the edges. The MIND dataset provides the entities that we can extract from and connect to WikiData using their own NER and entity linking tool. Let's have a look at an example of graph construction:

For example, for a particular user we have three clicked news d1, d2, d3:

Table 1: News Articles and their Entities

News Articles	Entities
News D1	Joe Biden, Catholic Church
News D2	Charles Rogers (American football), 2003 NFL Draft
News D3	United Kingdom, Sky Sports

As a result, the graph's nodes are D1, D2, D3, and the user's matching entity nodes. Edges: There are two sorts of edges:

a) Unidirectional Edges:
   - News-News: The direction of the edge from news1 to news2 is determined by the sequence in which their click timings were recorded.
   - News-Entity: The direction of the edge from the news to the entity it belongs to.

b) Bi-directional Edges:
   - Entity-Entity: Connect the same entity in different news

5.3.3.1: User Embedding: The representation of graph after training is used as user embedding

5.3.3.2: Candidate News Embedding: The candidate news passed through news encoder, its representation is used as candidate news embedding

5.3.3.3: Click Probability: The inner product of the vectors representing that of the user u and the candidate news d_c, i.e. $y = u^T d_c$, may be used to determine the click probability score y. We may also use a linear layer followed by some mathematical operations to determine if a news item is likely to be clicked or not.

6. Dataset

The Microsoft News Dataset (MIND), a dataset scalably large in size for news recommendation research, is used in our investigations. MIND has over 160K English news items and over 15 million impression logs from 1 million people who clicked on at least 5 news stories in the six weeks between October 12 and November 22, 2019. Each and every news story has a title, abstract, body, category, and entities, as well as extensive textual content. Each impression log contains the user's previous click events, non-clicked events, and historical news click habits. To safeguard user privacy, each user was securely hashed into an anonymised ID and the link from the production system is removed.

Table 2: Details of the dataset statistically presented

	MIND
# Users	1,000,000
# News	161,013
# Impressions	15,777,377
# Click Behaviors	24,155,470

We used the first 1000 examples for our experiments and performed a 90:10 split for training and testing.

7. Experimental Settings

To extract the word embeddings for each unique news text, title, and abstract, we utilised Bert's "Base" version, which is a language model which are pre-trained. As a result, the word embedding dimensions were set to 768. The attention inquiries were set to a size of 200. Adam is a programming language that is used to create optimization algorithms. Our measurements are based on accuracy scores. The weights for the nodes corresponding to the news and entities are randomly initialised using Xavier normal distribution and therefore trained using Relational Graph Convolutional Neural Network in the case of our entity graph (RGCN). We also used a Relational Network Convolutional Neural Network to train each node of the graph using initial embeddings pre-trained from the news encoder (RGCN).

8. Results and Analysis

The performance of our test set can be evaluated on the models we discussed i.e.

1) News Recommendation using PLM
2) News Recommendation using Explicit Entity Graph (random initialisation) and Pre-trained Language Model empowered news recommendation.
3) News Recommendation using Explicit Entity Graph(pre-trained embedding initialisation) and Pre-trained Language Model empowered news recommendation.
4) News Recommendation using CNN as Text Classification

Firstly aggregating the news representation in news extractor and entity graph by attention mechanism, separately.

Table 3: Performance Analysis on Dataset

Model Name	Train Accuracy (%)	Test Accuracy (%)
News Recommendation using PLM	33.28	35.18
News Recommendation using CNN	65.24	72.27
EEG(random initialisation)+PLM Empowered News Recommendation	72.16	73.37
EEG(embeddings initialisation)+PLM Empowered News Recommendation	77.38	77.78

Acknowledgements

I want to thank Dr. Sudeshna Sarkar, Professor, Department of Computer Science and Engineering, IIT Kharagpur, for her guidance throughout my B. Tech project work and for providing me with the necessary access to the facilities for smooth conduct of a wide range of experiments.

I also want to thank Ms. Madiraju SriLakshmi, Research Scholar, for her valuable input and guidance throughout the project.

References

[1] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online.

[2] ChuhanWu1, FangzhaoWu2, Tao Qi1, Yongfeng Huang1. 2021. Empowering News Recommendation with Pre-trained Language Models. In Proceedings of The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.475/123_4.

[3] Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2021. Combining Explicit Entity Graph with Implicit Text Information for News Recommendation. In Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3442442.3452329.

[4] Tao Qi1, FangzhaoWu2, ChuhanWu1, Yongfeng Huang1. 2021. Personalized News Recommendation with Knowledge-aware Interactive Matching. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462861.

[5] Hao Cheng, Xiaoqing Yang, Zang Li, Yanghua Xiao, Yucheng Lin. 2019, Interpretable Text Classification Using CNN and Max-pooling. https://doi.org/10.48550/arXiv.1910.11236