KRED: News Recommendation System with Data Enrichment and Domain Adaptation Extensions

1st Matteo Garbarino
Politecnico di Torino
Data Science and Engineering
Turin, Italy
s265386@studenti.polito.it

2nd Micol Rosini
Politecnico di Torino
Data Science and Engineering
Turin, Italy
s302935@studenti.polito.it

3rd Gaia Sabbatini
Politecnico di Torino
Data Science and Engineering
Turin, Italy
s291532@studenti.polito.it

***Abstract*—This paper presents an enhanced online news recommendation system that incorporates KRED, a knowledge graph-based recommendation model. KRED utilizes a knowledge graph system, taking into account information from the entity's neighbourhood and refining the entity embedding with dynamic insights based on the news context. In addition, we propose two extensions to the baseline model: data enrichment and domain adaptation. The first extension enables the system to exploit news reading time to model users' behaviour. The second extension expands the system's functionality to include movie recommendations, offering a more comprehensive and personalized user experience beyond news content. By integrating the KRED model with the aforementioned techniques, the enhanced recommendation system delivers accurate and context-aware recommendations. It leverages the knowledge graph's interdependencies and dynamic insights to refine entity embeddings, resulting in improved performance and user satisfaction. The code is available at:**
https://github.com/micolrosini/KRED-Recommendation-System

I. Problem Statement

In today's digital age, the consumption of online news has become an integral part of people's daily lives. With the abundance of news sources and articles available, personalized recommendation systems play a huge role in helping users navigate through this vast sea of information. This paper focuses on the development and extension of a recommendation system for online news, utilizing the Knowledge-Aware Representation Enhancement model for news Documents (KRED) [1].

KRED is a powerful framework designed to enhance the representation of news articles by incorporating knowledge entities mentioned within the articles. This knowledge-aware approach provides valuable insights that can improve the accuracy and relevance of recommendations. The versatility of KRED is showcased by its ability to perform various recommendation tasks, including item-to-item and user-to-item recommendations, news popularity prediction, news category prediction, and local news detection.

Due to limited computational resources, this paper primarily focuses on the user-to-item recommendation task within the KRED framework. By analyzing users' reading history and comparing it with the characteristics of news articles, the goal is to recommend news articles that are most likely to be of interest to individual users.

In addition to the baseline KRED model, this paper proposes two extensions that further enhance its capabilities.

- **Data enrichment**: by extending the model to accommodate a dataset containing information about the reading time spent by each user on each news article clicked. The recommendation system can therefore learn from a more informative dataset leading to better user preference modelling.
- **Domain adaptation for Movies recommendations**: by leveraging Mind Reader dataset, that contains film-related information and entities, the extended KRED model can recommend movies that are highly likely to be enjoyed by users based on their preferences and interests.

The contributions of this paper lie in the adaptation and extension of the KRED model for user-to-item recommendations in the online news domain. By incorporating data enrichment and movie recommendations, the enhanced model strives to provide a more tailored and personalized user experience. The following sections will delve into the methodology, experimental setup, and evaluation results to demonstrate the effectiveness and benefits of these extensions in the recommendation system for online news using the KRED model.

II. Methodology

In this paper, we propose a Knowledge-aware Representation Enhancement model for news documents (KRED) that utilizes the red backbone to create a recommendation system for news and other extensions. The methodology of our approach consists of three key layers: an entity representation layer, a context embedding layer, and an information distillation layer.

1) **Entity Representation Layer**: the assumption used is that entities in news articles can be linked to corresponding entities in a knowledge graph. TransE is used to learn embedding vectors for each entity and relationship in the knowledge graph. To represent an entity, we employ the Knowledge Graph Attention (KGAT) Network [2], which considers both the entity's own embeddings and its neighbours' representations.
2) **Context Embedding Layer**: to reduce computational costs, the conclusive information of entities is extracted from the original document. There are three context

embedding features: position encoding, frequency encoding, and category encoding. The position encoding distinguishes between entities appearing in the title and body of the news, while the frequency encoding captures the importance of entities based on their occurrence frequency. The category encoding helps the model understand content more accurately by revealing the category of entities.

3) **Information Distillation Layer**: the importance of an entity is not solely determined by its own message but also influenced by other co-occurring entities and the topics of the article. An attentive mechanism merges all entities' information into one output vector. The attention weight is computed using a two-layer fully connected neural network, and the key and value are entity representations obtained from TransE. The attention mechanism allows for information propagation from neighbour nodes to the current entity.

Finally, the entity vector and the original document vector are concatenated and fed into a fully-connected feed-forward network to obtain the Knowledge-aware Document Vector (KDV) [3]. By employing the KRED model, we can generate Knowledge-enhanced Document Vectors (KDV) that can be consumed by various downstream applications, including recommendation systems for news. Our framework has the advantage of accommodating different document representation methods, utilizing all available data in news documents, and achieving computational efficiency. The KRED model is highly effective for news recommendations due to its ability to cover a broad range of applications. It goes beyond personalized recommendations and includes item-to-item recommendation, news popularity prediction, new category prediction, and local news detection. This wide coverage makes KRED a versatile and comprehensive solution for various aspects of news recommendation systems. Overall, the proposed methodology of the KRED model enables the development of other extensions related to recommendation systems. Fig.1 shows the proposed KRED model.

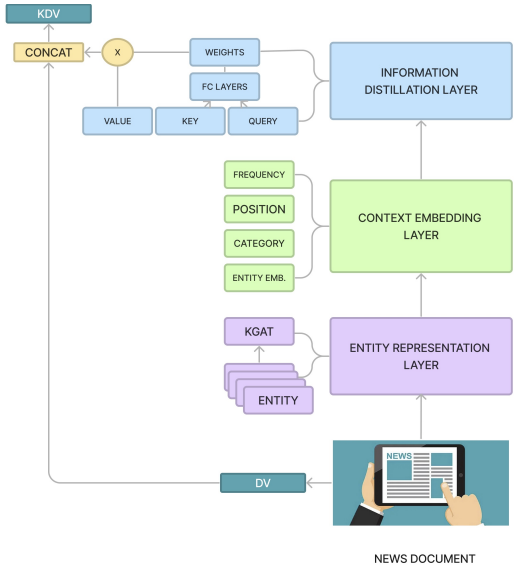

Fig. 1: An overview of the proposed KRED model. FC-LAYERS indicates fully connected layers. DV indicates the (original) document vector. KDV indicates the knowledge-enhanced document vector produced by KRED.

A. First extension: Domain adaptation with Movies recommendations

The *Mind-reader dataset* [4] used in the training process contains movie entities and related entities such as actors, companies, film categories, release years, and directors. Additionally, there is a rating file where users have provided positive or negative ratings for movies or entities related to movies. The dataset also includes a knowledge graph that connects entities with specific relations.

To recreate the files necessary for training the recommender system, the following steps were performed:

1) **Manual analysis and mapping of Wikidata IDs**: since not all entities in the dataset had associated Wikidata IDs, a manual analysis was conducted on the Wikidata database to find the corresponding IDs for those entities. This mapping process ensured that each entity had a unique identifier.
2) **Querying Wikidata for entity embeddings**: the textual description of the entity ID is retrieved from the Wikidata database. Then it is passed to the BERT [5] model to obtain the entity embedding.
3) **Creating movie embeddings**: since movies descriptions were not provided, in order to create movie embeddings and capture all relevant information, we gathered entities associated with each movie from the knowledge graph. This allowed us to enrich the movie embeddings with details about actors, directors, year of publication, and genre. Additionally, we encoded additional information such as position, frequency, and category for each entity, following the approach used in the original training. The entity category was determined by analyzing the relations present in the knowledge graph. For example, if two distinct entities were connected to other entities using the same relation, such as *Starred by*, it indicated that these second linked entities shared the category of *actor*. Since we didn't have an abstract or summary available for the movies, we treated all entities as part of the "title." Fig. 2 shows the difference between the original news entity embedding and the movies entity embedding.
4) **User history creation**: the user history was generated by selecting movies that were positively rated by each user. These movies were considered part of the user's history since they served as a reflection of the user's preferences and viewing experiences.
5) **Positive and negative lists**: Using the rating file, posi-

tive and negative lists were created for each user. These lists contained movies and entities that were liked or disliked by the user. By including entities like actors or directors in the lists, the model would be able to recommend movies featuring those entities to users.

Following these steps, all the necessary files were recreated to train the recommender system.

B. Second extension: Data enrichment

The reason for developing this extension is to model more accurately the user preferences. In the news context, the assumption that a click on an article always expresses a real interest may be an oversimplification. Users may quickly close an article due to a "clickbait" title or lose interest while reading.

The goal is to train the KRED model by exploiting users' article ***reading time*** to better understand their behaviours, using a richer dataset that includes reading time information. For this purpose, we utilized the *Adressa Norwegian News Dataset* [6].

To recreate the files necessary for training the recommender system, the following steps were performed:

1) **Data loading**: the *Adressa dataset* was originally a chronologically ordered log of user clicks on news articles. Due to the dataset's large size (16GB) and its irregular schema, a significant effort was invested in this step to efficiently load the data into a suitable data structure. Despite utilizing an ad-hoc environment like Google Colab, the limited disk space and RAM posed as constraining factors. Consequently, careful selection of data structures and implementation of memory management operations were essential to obtain a Pandas DataFrame [7] for further processing.
2) **Dataset preprocessing - user modelling**: the key aspect in generating the `behaviours.tsv` file, which includes user preferences, was to extract both the user history and their behaviours, namely positive and negative clicks. While the user history is a subset of the positive clicks, the definitions of positive and negative clicks had to be renewed to consider the ***reading time***. For each user, we computed the mean and standard deviation of its reading time for all the news articles he clicked on. If the reading time for each user click was within one standard deviation from the mean user reading time, it was classified as a positive click. Otherwise, it was considered a negative click. The interpretation behind this classification is that if a user closes an article too quickly or takes too long compared to their usual reading habits (for example, losing interest in the article and leaving the page open), it is reasonable to assume that there was no genuine interest in that particular article.
3) **Dataset preprocessing - news modelling**: the entities embeddings and the knowledge graph files were not provided by the dataset and had to be generated. As the news corpora were not available, the extraction of entities was carried out using the dataset's metadata, which included information such as news title, category, keywords, and the "profile" field containing a list of potential entities. These extracted entities were then mapped to their corresponding WikidataID, and the same process as described in II.A.22 and II.A.33 was applied. To construct the Knowledge Graph from the ground up, requests were made to the Wikidata API, and an iterative process was carried out using the previously obtained list of Wikidata ID entities.

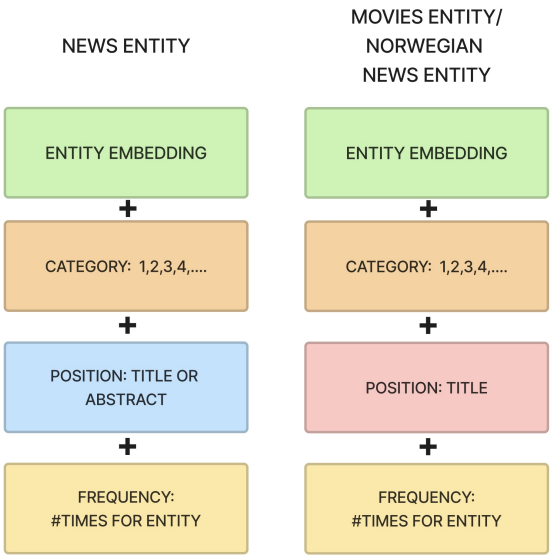

Fig. 2: Difference of the original news entity embedding and the movies/addressa entity embedding: the only position considered is title, there is no abstract's entities.

III. Experiments

A. Dataset Description

1) Microsoft News Dataset (MIND): The KRED framework utilizes the Microsoft News Dataset (MIND), a large-scale collection of anonymized behaviour logs from the Microsoft News website. MIND consists of 1 million users and over 160,000 English news articles. Additionally, a smaller version called MINDsmall is available, which includes a random sample of 50,000 users and their corresponding behaviour logs.

It contains the following files:

- `news.tsv`: for each news article it contains the News Id, category, title, abstract, url, wikidataIDs of the entities mentioned in the title and in the abstract.
- `behaviours.tsv`: for each user's log it contains User-Id, Time stamp of the log, user's news reading history, a list with the news along with a label that is equal to 1 if the news shown to the user was clicked on, 0 otherwise.

In the provided dataset, along with the news articles, the embeddings of entities and relations are included. Additionally, a knowledge graph is provided that establishes connections between entities through relations.

2) Mind Reader Dataset: MindReader is a novel dataset providing explicit user ratings over a knowledge graph within the movie domain, it contains 218,794 ratings from 2,316 users over 12,206 entities, and an associated knowledge graph consisting of 18,133 movie-related entities. The dataset is collected from an online movie recommendation game, MindReader, where users are pseudo-randomly asked to provide preferences for both movie and non-movie entities (e.g., genres, actors, and directors). For each entity, users can either like it, dislike it, or state that they do not know it. It contains the file

- `ratings.csv` containing for each item the UserId, the WikidataId of the entity, a boolean indicator that states if the item is a film or not, the value of the rating(+1/-1/0)
- `entities.csv`: for each entity it has the wikidataId, the name of the entity, and the category.

In the provided dataset, entity and relations embeddings were not explicitly given but the knowledge graph is provided.

3) Adressa Norwegian News Dataset: Adressa is an anonymized user behaviour log collected from Adresseavisen, a local newspaper company in Trondheim (Norway), for Recommendation Technology research purposes. The dataset version used for the experiments is AdressaSMALL, collected over a full week (Jan 1st to Jan 7th 2017) and consists of over 280.000 users, 40.000 news articles and 10.000.000 clicks. The dataset is structured with one file for each day, and each file contains:

- `Adressa_day.txt`: for each log entry it contains the UserId, the timestamp of the click, the URL of the clicked news, news title, category, keywords, a list of possible entities and other features not relevant to our study.

The dataset's missing values have been dropped, except for the reading times where the missing value meant less than 1 second. The embeddings of the entities and the Knowledge Graph were not provided, necessitating their computation from scratch.

B. Experimental Design

Considering the limitations in memory space and processing time, both on our local machines and Google's Colab platform with the Tesla T4 GPU (16GB memory), it became apparent that processing the entire MIND dataset was not feasible. Consequently, we decided to work exclusively on the MINDsmall dataset (50000 users). During the preprocessing stage, which encompassed the majority of data preparation for both the original task and its extension, we primarily relied on our local machines. However, when it came to the training phase, we leveraged the powerful computational resources offered by the Google Colab platform.

Due to computational power limitations, we chose to concentrate solely on the user-to-item task. Consequently, the evaluation metrics utilized in this work are:

- the **AUC (Area Under the Curve)** score: commonly employed in binary classification tasks. It measures the performance of a classifier. This metric ranges from 0 to 1, where a score of 1 indicates a perfect classifier. [8]
- the **NDCG@10** score: it stands for Normalized Discounted Cumulative Gain, sums the true scores ranked in the order induced by the predicted scores, after applying a logarithmic discount; then it is divided by the best possible score (Ideal DCG, obtained for a perfect ranking) to obtain a score between 0 and 1. [9]

Table I showcases the optimal hyper-parameter configurations for each model obtained through a grid-search process.

Hyper-parameters	**Configuration**	**Values**
Epoch	KRED-News	5
	KRED-Movies	5
	KRED-Enriched News	5
Batch size	KRED-News	64
	KRED-Movies	64
	KRED-Enriched News	128
Learning rate	KRED-News	$2e^{-5}$
	KRED-Movies	$5e^{-5}$
	KRED-Enriched News	$1e^{-5}$
Learning rate decay	KRED-News	$1e^{-6}$
	KRED-Movies	$2e^{-6}$
	KRED-Enriched News	$1e^{-6}$
Optimizer	KRED-News	Adam
	KRED-Movies	Adam
	KRED-Enriched News	Adam

TABLE I: Best hyper-parameters configuration

C. Execution Times

For the original news recommendation task, training the KRED model took approximately 20 minutes per epoch. When using the Adressa dataset for news recommendation, the training process extended to around 25 minutes for each epoch. Due to computational constraints, we made a decision to restrict the training of these two models to only 5 epochs. On the other hand, for the Movies Recommendation task, the KRED model demonstrated faster training times, with each epoch requiring only 5 minutes. Interestingly, even in this scenario, the optimal configuration for the model was achieved after 5 epochs.

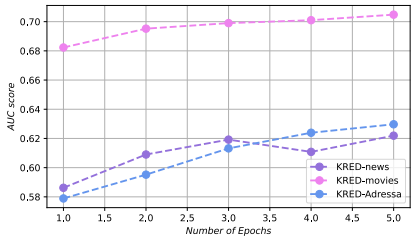

Fig. 3: AUC-score for the different model configurations for each epoch.

D. Results

The performance evaluation of our experiments is presented in Table II. When examining the KRED model for the original

news recommendation task, it is important to note that the reported metrics are lower compared to those in the original paper. This discrepancy can be attributed to the utilization of a smaller dataset and a limited number of training epochs. As for the Adressa data enrichment extension, although it did not show a significant improvement in model performance, the results remained comparable, albeit slightly higher, due to the substantial size of the dataset.

However, the Domain Adaptation Extension emerged as the most successful experiment, achieving remarkably high performance levels. The model's capability to adapt and perform exceptionally well in this scenario is noteworthy. Fig.3 shows the AUC-score for each epoch of these three models.

Configuration	**AUC score**	**NDCG@10**
KRED for NEWS recommendation	0.6219	0.3899
KRED for MOVIES recommendation	0.7048	0.7670
KRED for ENRICHED NEWS recommendation	0.6297	0.3901

TABLE II: AUC and NDCG@10 score for the different models.

IV. Conclusions and Takeaways

We started with KRED as a baseline model, which leverages knowledge graphs to enrich its embeddings. During our research, we developed two extensions with distinct objectives. While the performance of the baseline model was slightly lower, this can be reasonably attributed to the use of the MIND Small version of the dataset. Unfortunately, we couldn't verify this assumption as we lacked the computational resources to work with the full MIND dataset.

The first extension, domain adaptation for movies recommendation, delivered exceptionally high performance, demonstrating its versatility and the effectiveness of its innovations.

On the other hand, the second extension, aimed at data enrichment through reading time utilization, showed lower performance, which may be attributed to various factors. Introducing a new dataset to incorporate reading time information made it challenging to ensure a fair comparison with the baseline. Extracting entities from metadata and rebuilding the knowledge graph proved to be significant obstacles, along with managing computational resources effectively.

For future implementations, we contemplate adding the length of each article to the user's reading time to avoid negatively affecting interesting and lengthy articles. Additionally, translating the entities before querying Wikidata could lead to more accurate Wikidata IDs and potentially improved results.

References

[1] D. Liu, J. Lian, Y. Qiao, J. Chen, G. Sun, and X. Xie, "Fast and accurate knowledge-aware document representation enhancement for news recommendations," *CoRR*, vol. abs/1910.11494, 2019. arXiv: 1910.11494. [Online]. Available: http://arxiv.org/abs/1910.11494.

[2] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*, ACM, Jul. 2019. DOI: 10.1145/3292500.3330989. [Online]. Available: https://doi.org/10.1145%2F3292500.3330989.

[3] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, 2019, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.10.026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705118305124.

[4] Microsoft News Dataset, *MSNews website*, https://msnews.github.io/, 2023.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[6] J. A. Gulla, L. Zhang, P. Liu, Ö. Özgöbek, and X. Su, "The adressa dataset for news recommendation," in *Proceedings of the International Conference on Web Intelligence*, ACM, Aug. 2017, pp. 1042–1048.

[7] T. pandas development team, *Pandas-dev/pandas: Pandas*, version v2.0.3, If you use this software, please cite it as below., Jun. 2023. DOI: 10.5281/zenodo.8092754. [Online]. Available: https://doi.org/10.5281/zenodo.8092754.

[8] Y.-M. Tamm, R. Damdinov, and A. Vasilev, "Quality metrics in recommender systems: Do we calculate metrics consistently?" In *Fifteenth ACM Conference on Recommender Systems*, ACM, Sep. 2021. DOI: 10.1145/3460231.3478848. [Online]. Available: https://doi.org/10.1145%2F3460231.3478848.

[9] Y. Wang, L. Wang, Y. Li, D. He, T.-Y. Liu, and W. Chen, *A theoretical analysis of ndcg type ranking measures*, 2013. arXiv: 1304.6480 [cs.LG].