KRED: Knowledge-Aware Document Representation for News Recommendation

Nasirimajd, Amirshayan
Polytechnico di Torino
amirshayan.nasirimajdi@studenti.polito.it

Karimizandi, Bardia
Polytechnico di Torino
bardia.karimizandi@studenti.polito.it

Behkish, Arman
Polytechnico di Torino
arman.behkish@studenti.polito.it

***Abstract*—This report presents a study on the KRED (Knowledge-Aware Document Representation for News Recommendations) [1] method. The study's objective is to analyze the impact of different KRED model components on the classification head's overall performance. The KRED model is a knowledge-aware document representation approach that incorporates external knowledge sources to enhance news recommendations. It contains three layers: entity representation, context embedding, and information distillation layer. In this study, we examined the effects of context embedding layer components' position, category, and frequency over the KRED model. Also, we explored the impact of different natural language understanding (NLU) models and robustness in classifying detailed classes. We evaluated the performance of the modified models using standard classification metrics, including accuracy and F1 score. The results of our experiments show that the KRED model's performance improves with the change of the NLU model. Additionally, we observed that modifying the context embedding component had a minor effect on the model's performance. In contrast, the model's ability to compare a detailed (higher) number of topic classes dropped significantly. This report highlights the importance of incorporating KRED model components in document representation models, which can improve their effectiveness in news recommendation tasks. Furthermore, the model performance has been tested over MIND small dataset from Microsoft. Github repo: https://github.com/bardiakzzzz/KRED**

***Index Terms*—knowledge graph, News Recommendations, Ablation study, Representation Learning, Text classification**

I. Introduction

News recommendation has been widely used for assisting users in digesting the extensive flow of daily news in diverse media by providing related articles based on user preferences.

In traditional recommendation systems, User preferences are learned using specific (e.g., ratings) or implied (e.g., browsing memory) feedback. These systems can be categorized into collaborative filtering, content-based, and hybrid methods. Collaborative filtering systems recommend items that acquired the user's interest in the past with similar preferences to the current user. In contrast, content-based algorithms suggest items with similar characteristics to the ones the user prefers. Hybrid models combine one or more types of recommendation approaches to address weaknesses such as the cold-start problem and the over-specialization issue. [2]

The news-related problems have specific challenges for traditional recommendation systems due to the particular factors that news items present. News articles are highly time-sensitive and soon become obsolete; thus, collaborating filtering methods might be less practical. This is because news content is critical, making the use of the utility-matrix less effective. Incorporating external knowledge sources has been proposed to capture information and patterns not contained in the text and metadata of the article. The entities, e.g., names, places, events, etc., usually convey vital messages of the article. Knowledge graphs are one of the important ways to encode real world information and their relations so that algorithms can utilize this knowledge for better-comprehending news articles. One of the most important works to integrate knowledge graph information into news recommendations is DKN [3]. It is a framework for click-through rate predictions that uses directed graphs in which nodes are entities and edges are relations. It uses TransE [4] for knowledge graph embeddings. TransE is a fast algorithm that creates a low-dimensional vector for entity relations by leveraging (head, label, tail) triplets. It utilizes heterogeneous multi-relational knowledge graphs. In DKN, Word embedding, entity embedding from TransE, and context embedding are combined using a CNN network to create a compact embedding for the news document. In this setup, Context embeddings are extracted using the nodes in a one-hop radius of the knowledge graph. This technique only uses news titles and merges entity vector to word embedding, which is not very efficient. Consequently, the main pitfall is the inability to use news bodies due to representation that could be used as the model's input. The main goal of KRED is to develop a way of fusing entity information in the news article into latent representation without having any limitation on the NLU model used to obtain the document representation.

The KRED model is a model for document representation. It consists of three main layers to represent the information of documents to a multi-task framework to improve generalization ability and efficiency. The KRED proposed an entity representation layer, a context embedding layer, and an information distillation layer. Considering these layers, we tried in this work to comprehend the component-wise workings of this model by doing an ablation study over the context embedding layer, changing the NLU model, and exploring the ability of the model to improve the performance by leveraging the subclasses. Furthermore, we present the model's performance and

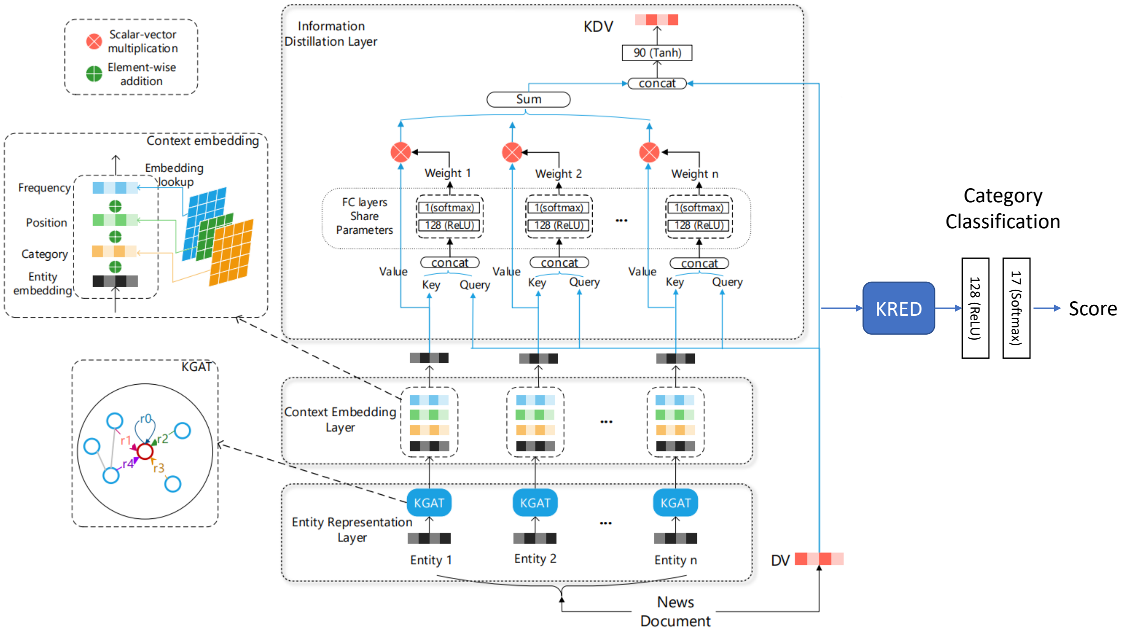

Fig. 1. An overview of the KRED model. DV indicates the (original) document vector. KDV indicates the knowledge-enhanced document vector produced by KRED. UV indicates the user vector. In the end, also there is a simple category classification head.

understanding in the Experiment section after testing each part on the MIND [5] dataset.

II. Methodology

In this section, we will introduce the KERD model in more detail, and related implementation changes in architecture for each experiment.

A. KRED: Overview of the Pipeline

The Knowledge-aware Representation Enhancement model for news Documents (KRED), is illustrated in Fig 1 Given an arbitrary Document Vector, which is produced by BERT, denoted as v_d, produces a Knowledge-enhanced Document Vector (KDV) to tackle different applications. However, in our case, we use a single-task model for category classification due to the lack of computational resources. Moreover, the KRED architecture consists of three layers Entity Representation, Context Embedding, and Information Distillation which will be discussed in the following parts.

1) Entity Representation Layer: In the first layer, TransE is used as a fast and efficient entity embedding method. Moreover, Knowledge Graph Attention Network [6] is used to further enrich entity embedding. KGAT is a knowledge propagation mechanism that leverages attention to detect the importance of item-user nodes in the heterogeneous graph and merge it into the final embedding. Moreover, KRED uses TransE to learn embedding vectors for each entity and relationship, and it is trained on the knowledge graph. In addition, the embeddings of entities/relationships are fixed as features for the KRED framework. considering that an entity is represented by its own embeddings, and also partially by its neighbors, KRED uses Knowledge Graph Attention (KGAT) Network to produce an entity representation.

2) Context Embedding Layer: In the second layer, the dynamic context of the entities is fused into the vectors. It contains positional encoding, which determines whether the entity is in the body or the title; frequency encoding which adds the frequency of the word (with the upper bound of 20); and category embedding based on the category of the entity in the knowledge graph.

3) Information Distillation Layer: The third layer is an attention layer that combines all entities using the original document representation as a query and entities as keys and values. The final embedding is obtained by concatenating the attention layer output by the initial vector and going through a dense layer.

B. Extensions

After Understanding the three main layers of the KRED model, we implement the following study by changing the architecture of data manipulation of the KRED model.

1) Model Investigates: The model has meaningful performance improvements over similar methods, and the document embedding could be performed with any NLU model. However, in the original implementation in the paper, the model uses BERT to represent the embedding of document vectors. Therefore, we replaced it with two other NLU models to get the document embedding.

- The first one is RoBERTa: A Robustly Optimized BERT Pretraining Approach [7]. RoBERTa is another transformer-based language model that uses self-attention to process input sequences and generate contextualized representations. In addition, it was trained on a much larger dataset compared to BERT and used a more effective training procedure. This model has been fine-tuned with a Sentence similarity approach to produce semantically meaningful sentence representations. As a result, we can get the same embedding size with better performance.

- The second one is MiniLM [8], a simple language model with fewer parameters than BERT. Therefore, it is a faster model for language understanding and generation.

2) General Category vs Sub Category: In this part, we were eager to see the model's ability to understand the class of documents in a more detailed number of classes. While in the main implementation, the focus was on 17 topics to classify the news, we decided also to use a more detailed class set to assess the robustness of the model. Therefore, we used subvert, which consists of subclasses of the 17 main topics, which is 263 different classes.

3) Context Embedding Ablation Study: Since the context embedding layer is created from three main parts position, frequency, and Category encoding, we decided to test how much each of these components can enhance the model's performance in classification. To do so, we tried the impact and performance of each element separately as an ablation study.

III. Experiments

A. Dataset

The authors of KRED used a real-world dataset provided by Microsoft news, containing the users, news, and user interactions. The average number of words in documents is 700. In addition, an industrial knowledge graph is used to extract the one-hop proximity of entities. MIND [5] is the open-source Microsoft news dataset that we used for training the model. It is sampled from the users with at least five news clicks in 6 weeks. The dataset is further sampled down to 50,000 users into MIND-small. It contains four files. The impression log collects the user's ordered clicked news history and user click behavior (impression) on the displayed news. The news file contains the information for each news item, including the category, subcategory, abstract (title), and item entities. The body of the news is not available. Entity information is provided from the Wikidata knowledge graph. Other files contain the 100-D entity and relation embedding extracted from the Wikidata knowledge graph.

B. Experimental Design

We have done a series of ablation studies to compare the performance of the model in different situations. In order to be able to train the model in a reasonable time on Google Colab, we had to use single-task category classification and discard the prediction of other heads. In the default setup, the model uses all the prediction heads, although their output is not used in single-task classification. This makes the training time intractable, and we could not train the model in this configuration. So we turned off other heads and used the category classification for our tests. Using this configuration, we were able to train the model in 8 to 10 hours, depending on the test being performed.

C. Extensions Test Result

In the following section, we will indicate the results of our experiments for each part. Furthermore, the basic hyperparameters that have been used during the experiments are indicated in TABLE I.

1) Base Line: In the first part, we calculated the baseline which was the presented model in the repository without any change in the default configurations, and the main architecture without any change. To compare the results of the baseline we used two metrics: accuracy, which is the average correct prediction in all classes, and then the F1 score as a balance of precision and recall for our predictions, to have a balance prediction in all classes.

TABLE I
Basic Hyperparameters

Parameter	Value
Batch size	64
epochs	100
Optimizer	Adam
learning rate	0.00002
weight decay	0.000001

2) Model Investigates Reuslts: In TABLE II, we indicated the results of two NLU models. Therefore, we can see that using the RoBERTa which is a more complicated language model, with a higher amount of data training achieves better results in comparison to the baseline, Which used BERT. In addition, we can see the accuracy increased +4.37 percent, while the F1 score increased by 15.99 which also indicates how much performance on NLU can impact the prediction balance between several classes. On the other hand, using a simple fast language model, i.e., MiniLM decreased the performance of KRED in category classification.

TABLE II
NLU model results.

Setting	Accuracy	F1 score	Gain
Baseline	70.81	0.35	-
RoBERTa	75.18	0.51	+4.37
MiniLM	65.08	0.22	-5.37

3) General Category vs Sub Category Results: In this experiment, which results are shown in the TABLE III, we can see not only model fail to do a more accurate prediction over all classes, but also it is clear based on the F1 score that the model prediction is not balanced over all classes. Therefore, when we are looking more into topics that are not general KRED model fails to give good results.

TABLE III
Class Number Impact

Setting	Accuracy	F1 score	Gain
Baseline	70.81	0.35	-
263 Class	34.68	0.02	-36.13

D. Context Embedding Ablation Study Results

Finally, by doing an ablation study on the effect of single components of context embedding layer, we can see at TABLE IV that while type has an important effect in topic categorization and removing it will result in a decrease in accuracy, the position has a reverse effect, and removing it can even improve our results a little bit.

TABLE IV
CONTEXT EMBEDDING ABLATION STUDY RESULTS

Setting	F1 score	Accuracy	Gain
Baseline	70.81	0.35	-
No Type	70.11	0.34	-0.7
No Position	71.10	0.36	+0.29
No Frequency	70.84	0.35	+0.03

IV. CONCLUSION

In this report, we experimented with the KRED model, which is an impactful approach to fuse knowledge obtained from news entities into document representation. This method has successfully beat state-of-the-art in multiple recommendation tasks while being relatively fast and flexible. We conducted our experiments with two main limitations: the much smaller dataset to train on and hardware and time constraints. Despite those limitations, we could obtain very close results on the news classification task to the best of the paper. The gain was mainly due to the essential role of the base language model for document representation. by using RoBERTa, we obtained a significant improvement over our baseline. Moreover, our experiments with the dynamic context layer showed the effect of the lack of each encoding, how much it impacts the performance, and indicated the importance of Type encoding in the KRED framework. In contrast, lack of frequency does not give us any significant difference, and type encoding absence can even improve the model. To conclude, leveraging knowledge entities in document representation is still an exciting and challenging avenue of research to significantly improve the strength of current language models for news recommendation tasks.

REFERENCES

[1] D. Liu, J. Lian, S. Wang, Y. Qiao, J.-H. Chen, G. Sun, and X. Xie, "KRED: Knowledge-aware document representation for news recommendations," in *Fourteenth ACM Conference on Recommender Systems*, ACM, sep 2020.

[2] A. Iana, M. Alam, and H. Paulheim, "A survey on knowledge-aware news recommender systems," *Semantic Web*, no. Preprint, pp. 1–62, 2022.

[3] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *Proceedings of the 2018 world wide web conference*, pp. 1835–1844, 2018.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[5] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, *et al.*, "Mind: A large-scale dataset for news recommendation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3597–3606, 2020.

[6] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 950–958, 2019.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[8] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.