Graph Convolutional Networks and Text Integration for Recommender Systems

Anonymous ACL submission

Abstract

We present a new model KeywordSage which consists of integration in text data and Graph Convolutional Networks for recommender systems. This model extracts keyword in most efficient way from user reviews text using language model based on Transformer and then Graph Convolutaionl Networks is efficiently trained to learn about user-item interactions by utilizing extracted keywords. This makes it possible to reflect meaningful information from users and utilize it for representing the user-item interaction. We prove that our model is more efficient showing that KeywordSage result in better performance even with significantly fewer learning steps compared to existing models. Our approach is to be a meaningful contribution in that it proposes a new recommender systems by combining Natural Language Processing and a graph-based neural networks, suggesting a direction for covering research in both fields.

1 Introduction

In the field of recommender systems, Collaborative Filtering, Content-based filtering, and Matrix Factorization have been studied actively as traditional methodologies. These trends show that recommendation systems have been recognized as a very important task in many real-worlds. Recently, as deep learning has been revolutionizing dramatically in the overall field of society, the research on personalized recommender systems based on artificial neural networks has also been actively studied (Sun et al., 2019; He et al., 2017; Sarwar et al., 2001; Hidasi et al., 2015). In this study, we propose a new recommender systems model using artificial neural networks that can provide personalized recommendation services by accurately understanding users and items characteristics, and complex interactions.

KeywordSage, which we propose, effectively extracts meaningful keywords from user review text through language model based on Transformer. Afterwards, the extracted keywords and interactions between users and items are learned using an artificial neural network based on Graph Convolutional Neural Networks. With this structure, our model learned meaningful interactions between users and item characteristics, which resulted in providing more accurate and faster personalized recommendation services.

Graph Convolutional Networks (GCN), based on graph structure, is one of the key research areas that has recently attracted attention in the field of artificial neural networks. Hamilton et al. (2017b) presented that expressing data with a complex connection structure in a graph structure consist of nodes and edges, and learning this with a Graph Neural Network (GNN), is a effective methodology for learning the representation. And this increases the applicability of GNNs in various fields such as social networks, molecular structure prediction, etc. Monti et al. (2017) present the potential of graph structured artificial neural network showing that multi-graph neural networks can effectively model the geometric characteristics of data. Berg et al. (2017) deal with the Matrix Completion using a graph structure and show the results of effectively solving the problem in the recommender systems by implementing a system that predicts the rating a user gives to an item. These research show the potential for development of Graph Neural Network for recommendder systems and suggest directions for implementing better recommendation algorithms and personalized services.

The field of Natural Language Processing (NLP) is also one of the areas that has recently attracted attention in the field of artificial neural networks. The possibilities of language models are attracting more attention appearing Large Language Models such as GPT (Radford et al., 2018), LLAMA (Touvron et al., 2023). From the traditional methodology, Term Frequency-Inverse Document Frequency

(TF-IDF), to the artificial neural network models Word2Vev (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) models, previous methodologies mainly consider only the frequency of words or only the context around words when they learn the representation. On the other hand, a language model based on Transformer (Vaswani et al., 2017) understands the entire context and learns representation much more effectively through Attention mechanism. Therefore, in this paper, we seek to extract meaningful keyword information from user review through a Transformer-based language model and implement a recommendation system using this.

The contributions of this study are as follows:

- Proposal of new model for recommender systems based on keyword-centered and graph structure: Effectively extracts keywords from user review through Transformer-based methodology and uses this to propose a model for recommender systems that learns user-item interaction through Graph Convolutional Neural Networks.

- Combining various research fields: This methodology, developed by combining the latest research in natural language processing and graph neural networks in the field of recommender systems, suggests the possibility of convergence between various research fields and suggests a new direction for recommendation algorithms.

- Establishing a research foundation for recommender systems based on text: By proposing a method to approach a recommender systems using text, it can serve as a cornerstone for future research on recommender systems using text.

In Section 2, we review previous research on reccomender systems and Transformer-based language model. In Section 3, we explain in detail the overall structure and pipeline of our proposed model. In Section 4, we discuss experiments and results and, In Section 5, we discuss future research.

2 Related Works

In this part, we will briefly review previous studies. Topics related to this study include general recommender systems, Graph Convolutional Networks for recommender systems, and Transformer-based language model.

2.1 General Recommender Systems

Collaborative Filtering, Content-based Filtering, and Matrix Factorization have traditionally been studied as methodologies in the field of recommender systems. Collaborative Filtering analyzes user behavior records to group users with similar preferences and provides recommendations based on group preferences. Content-based filtering analyzes the characteristics of items and recommends new items based on the user's previous preferences. Matrix Factorization decomposes the user-item matrix to discover hidden factors and provide recommendations to the user. These traditional methodology mainly analyzes existing data and models it using statistical methodology.

Artificial neural network-based methodologies have attracted attention because they can increase non-linearity and model more complex interactions. Hidasi et al. (2015) present Recurrent Neural Networks (RNN) model using the online session data of users to model the user's session in chronological order and uses this to recommend items. He et al. (2017) propose that by combining matrix decomposition techniques and neural network models in a collaborative filtering-based recommendation system, recommendation accuracy has improved and a personalized recommendation system can be implemented. Therefore, recommender systems based on artificial neural network have overcame the limitations and limitations of traditional recommendation systems and presented new perspectives and possibilities to provide more personalized recommendations to users.

2.2 Graph Convolutional Netowrks for Recommender System

As proven in numerous studies, Graph Convolutional Networks show excellent performance in efficiently extracting and representing information using convolutional operations in a graph structure.

Hamilton et al. (2017b) present study focusing on efficient representation learning of GNN using graph structure and applicability in various fields, and Monti et al. (2017) has proven to effectively model data geometric characteristics through multiple neural networks. Chen et al. (2018) announce the FastGCN model, a model based on Graph Convolutional Networks that introduce the Importance Sampling method, which is a graph synthesis tool that achieves high computational efficiency at a faster speed by selectively selecting the nodes necessary for neural network learning.

GraphSAGE (Hamilton et al., 2017a) and PinSage (Ying et al., 2018) models are notable studies that implemented a personalized recommendation system considering user-item interactions using Graph Convolutional Neural Networks. PinSage follows the basic structure of GraphSAGE and proposes a method of learning large-scale graph structures and optimize GPU using effective sampling and learning techniques such as Importance Sampling and Curriculum Training.

LightGCN (He et al., 2020)is a model that made meaningful contributions by proposing a graph-structured recommendation system with good performance while simplifying the graph convolutional neural networks. By using a single layer to consider only user-item interaction and implementing an embedding learning method without scaling, LightGCN simplified the model and showed similar performance to other complex models. Also the self-connection effect is derived through an embedding parameter learning method and the possibility of improving the over-smoothing problem is presented.

The model proposed in this paper follows a basic structure to the Graph Convolutional Networks of the GraphSage and PinSage models.

2.3 Transformer in Natural Language Processing

Language models based on Transformer(Vaswani et al., 2017) have brought innovation to the field of Natural Language Processing (NLP). Starting with Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), various Transformer-based language models such as DistilBert (Sanh et al., 2019), Roberta (Liu et al., 2019), and sentence–Bert (**?**) have emerged. The efficiency and accuracy of the language model have been proven through a lot of research. These models effectively calculate word embedding within sentences through Token Embedding, Segment Embedding, and Positioning Embedding that take context into account. Additionally, through Attention mechanism, interactions between words within a sequence of sentence are understood and important information is extracted. Therefore, in this paper, Sentence-Bert, which is a Transformer-based language model, is used to calculate keywords from text.

3 Method

In this Section, we introduce the overall structure and pipeline of the KeywordSage model, which we suggest. The structure of KeywordSage consists of a total of two stages. Keyword extraction based on Transformer is preformed in Stage-I, and Graph Convolutional Networks is trained based on the bipartite graph structure of the data reflecting the extracted keyword in Stage-II.

3.1 Keyword Extraction (Stage-I)

The Stage-I of the KeywordSage model which is the first stage of the methodology is described in this section. This includes the process of efficiently extracting keywords from customer review text data.

In order to extract meaningful keywords from text data, keywords are selected by considering the similarity between words in a sentence. Based on the N-gram, trigram is set as the basic unit of text and the most frequent trigram in the review text is calculated. It is assumed that frequently appearing keywords are representative of whole review. One set is created with the most frequent keywords, and another set is created by dividing the entire text review into trigram units. These two sets are used as encoder input values for Sentence-Bert, and the similarity is compared. Sentence-Bert has a Siamese Network structure and it help that allows different sentences A and B to pass through each encoder layer and then calculates the similarity of the results. The output of the encoder layers pass through the mean pooling layer to calculate sentence embedding, and the similarity is calculated using cosine similarity for each of these values u and v. Cosine Similarity is defined as the following equation (1) and Table 1 shows sample keyword extraction for a customer review.

$$CosineSimilarity = \frac{u \cdot v}{\|u\| \, \|v\|} \tag{1}$$

3.2 Graph Convolutional Networks (Stage-II)

In Stage-II, The structure and learning process of Graph Convolutional Networks is explained.

The Stage-II begins by importing the keywords finally output in the previous Stage-I as attributes of the user node. This forms a complete bipartite graph structure to be used as input to the convolution layer. The formed graph passes through the projection layer and becomes the input of the Graph Convolutional Layer.

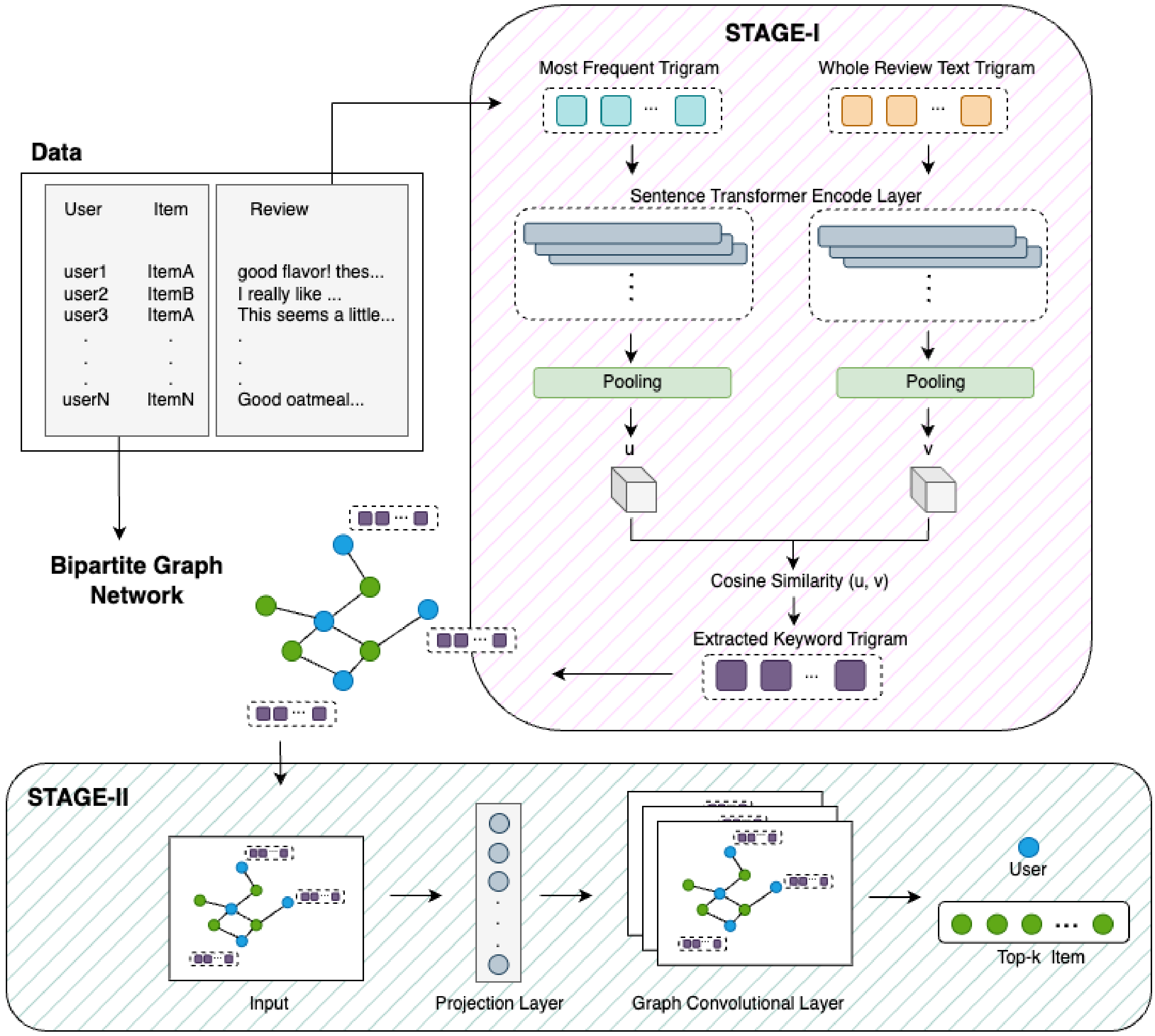

Figure 1: This is the first figure.

Review	I simply fell in love with these chips and refused to share them with my friends :)) My experience with a lot of cheese-favored chips has been too salty and a little over the greasy side. But these NY cheddar from Kettle really changed my mind. Thanks to Amazon for great discounts and I don't have to buy them from the store anymore.
Extracted Keyword	'amazon great discounts', 'experience lot cheese', 'greasy ny cheddar'

Table 1: This shows the examples extracted keyword from review. we use three keywords in trigram units for each review in our experiment.

The purpose of Graph Convolutional Networks is to produce high-quality embedding results for users and items and to use them in recommendation systems. The graph structure is consisted as a bipartite graph with the user node set U and item node set I. An interaction edge exists only between two different types of nodes u and i. The node set of entire graph is denoted as $V = U \cup I$.

Graph Convolutional Networks is trained about node embedding by repeatedly aggregating feature information from each node's neighbors. By stacking multiple convolution layers, it becomes possible to expand the receptor field to extract wider neighborhood information and reflect the structural information of the graph. The input of each convolution layer depends on the representation output from the previous layer. The first layer has the same characteristics as the input node. Model parameters are shared between nodes, but the values have different values for each layer. The output of the last k-th convolution layer passes through a fully-connected neural network to generate the final embedding.

For sampling process, the on-the-fly convolution method is used to sample neighboring nodes, and the Importance-Based Neighborhoods technique is used to sample them using distance information. On-the-fly convolution technique helps the graph structure learn quickly by implementing localized graph convolution that samples locally nearby nodes rather than processing the entire graph node at once. The Importance-Based Neighborhoods, which samples important neighbors by considering connectivity and weights between nodes, helps select and learn items that users interact with or meaningful neighbors of those items.

4 Experiments

In this chapter, we describe the experiments performed to verify our research. We will explain the datasets used in the experiment, the experimental environment, and the experimental results.

4.1 Datasets

The Amazon Product Review datasets(He and McAuley, 2016; McAuley et al., 2015) is used for this experiments. This datasets categorizes products into 29 detailed categories, and the data from the Grocery and Gourmet category is used for this study. This data consists of user information, information about the items the user purchased, text reviews written by the user, product ratings, and information about the time the review was written. We only use the data when the user interacted with at least 10 items.

When this datasets is converted to a bipartite graph, there are 7,323 user nodes, 27,251 item nodes, and a total of 137,696 interactions between users and items.

4.2 Settings Evaluation Metrics

The Max-Margin Ranking Loss is used as loss function for training. This is learned to minimize the inner product of query q and positive samples, and maximize the inner Positive sample(z_i)refers to the interaction with an item that has a purchase record from an actual user, negative sample(z_n) refers to interactions with items for which the actual user has no purchase record. Max-Margin Ranking Loss for the query vector q(z_q) to be predicted is defined as the following equation (2). In this study, we follow the loss function proposed in PinSage model.

$$JG(z_q, z_i) = E_{n_k \sim P_{n(q)}} max\left\{0, z_q \cdot z_{n_k} - z_q \cdot z_i + \Delta\right. \quad (2)$$

To efficiently perform Graph Convolutional Networks training, a producer-consumer mini-batch construction distributed processing pipeline is built in the experiments.Training is performed in a mini-batch using a predefined computational graph to efficiently perform SGD learning. In addition, training is performed using the Efficient MapReduce method and the curriculum training is used during the learning process to enable progressively more difficult learning data to be trained. These techniques help improve the computational efficiency and performance of learning.

4.3 Results

To evaluate the experimental results, we use the evaluation metrics Recall, Precision, and Hit-ratio which are widely used in recommender system research. Recall is indicator of how well the model recommended items are included among actual related items. Precision measures the proportion of actually relevant items among the items recommended by the model. Hit-ratio is an indicator that measures how many items a user actually purchased are included in a specific recommendation list. These three evaluation indicators can complement each other to evaluate the model's performance. For example, Recall can identify whether the model found many related items, but there is a lot of noise among them, also when precision is high, most of the items recommended by the model are highly relevant, but in reality, there are items missed. In this study, we use these indicators to evaluate and analyze the performance of model which we proposed.

4.4 Implementation Details

The experiment is run in an NVIDIA Telsa K80 GPU and Cuda 11.4 version. ADAM Optimizer is set as the optimization. To prevent over-smoothing, where the embedding vector excessively converges to the average of local information when multiple convolution layers are stacked, dropout is set to 0.5. The experimental results introduced in the

Model	Recall@20	Precision@20	Recall@30	Precision@30
PinSage	0.605	0.448	0.676	0.344
KeywordSage*	0.624	0.462	0.677	0.345

Table 2: Comparison table of experiment results. Our proposed model KeywordSage is marked with *.

next section are the results of staking two Graph Convolutional Layers.

Table 2 shows a comparison of the results of the PinSage model and our proposed model, the KeywordSage model. What is important in this table is that the results of the PinSage model proposed in existing research took a much longer learning period. The results of the PinSage model were achieved by running more than 20,000 learning epochs, and the results of the KeywordSage model were achieved by running the learning epochs 3000 times. This verifies that the model we propose is showing better results by significantly reducing the learning time. Additionally, when comparing the evaluation metrics, it can be seen that there is an improvement of 0.019 in Recall@20, 0.014 in Precision@20, and 0.003 in HR@20 of KeywrodSage model. In particular, it is figured out that the performance difference between the two models widens in the Recall score.

5 Conclusion and Future Work

In this study, we propose a model called KeywordSage that provides recommender services using text data based on Graph Convolutional Networks. This successfully models user-item interaction by utilizing Transformer-based keyword extraction and Graph Convolutional Networks. Our experimental results show that this model leads to improved results at a faster rate compared to the existing model PinSage.

Additionally, this study proposes the convergence of various fields such as recommender systems, natural language processing, and graph neural networks. Therefore, we expect that research on the convergence of Natural Language Processing and Graph Neural Networks will be able to expand into broader and more effective collaborative research in the future based on this study. Future research will especially need to study recommendation systems based on text data. With the emergence of Large Language Models(LLM), the speed of language model development is growing rapidly, and the potential to utilize text data and expand application areas in line with these changes is endless. Many industries currently have an enormous amount of text data accumulated. This data exists in many forms, such as consumer reviews, product descriptions, social media opinions, and more. In comparison, the amount of data we currently use is just fragments of a vast amount of information. Therefore, in the future, we expect to be able to propose a model that can increase the accuracy and quality of personalized services by focusing on research that can maximize the value of such data and the application potential of language models.

References

Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*.

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*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017a. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017b. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Federico Monti, Michael Bronstein, and Xavier Bresson. 2017. Geometric matrix completion with recurrent multi-graph neural networks. *Advances in neural information processing systems*, 30.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983.