This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

MASER: Multi-Order Attention and Semantic-Enhanced Representation Model for Complex Text Recommendation

Pei-Yuan Lai, Qing-Yun Dai, De-Zhang Liao, Zhe-Rui Yang, Xiao-Dong Liao, Chang-Dong Wang, *Senior Member, IEEE*, and Min Chen, *Fellow, IEEE*

***Abstract*—In some recommendation platforms, the recommended items are composed of the complex text, and the target users are also described by the complex text. These texts are usually long, highly specialized, logically structured, and have significant differences, such as recommending technical demands of enterprises to technology researchers. Although some recommendation methods based on text representation can be used to solve this problem, such as Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM), they may encounter challenges from different perspectives, e.g., path connectivity of representations, and the relationship between representations and recommended items. The complex text recommendation is an important problem that remains largely unsolved. In order to overcome the aforementioned challenges, by taking the technology commercialization as an example, which aims to recommend demands to researchers, we propose a novel complex text recommendation model called Multi-order Attention and Semantic Enhanced Representation (MASER). By integrating additional information into text vector representationsuch as structural relationship information for extended keywords, and semantic information for entity description texts the proposed model enhances complex text recommendation effectiveness significantly. Extensive experiments have been conducted on real datasets, confirming the advantages of the MASER model and the attention mechanism's effectiveness on complex text recommendation.**

***Index Terms*—Text recommendation, semantic enhancement, multi-order attention, knowledge graph.**

Manuscript received 7 September 2023; revised 24 May 2024; accepted 20 July 2024. This work was supported in part by NSFC under Grant 62276277, in part by the Guangdong Provincial Engineering Research Center of Intelligent Matching for Technology Commercialization under Grant 2022A175, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022B1515120059, and in part by the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data under Grant 2018B030322016. *(Corresponding author: Qing-Yun Dai.)*

Pei-Yuan Lai is with the School of Information Engineering, Guangdong University of Technology, Guangzhou 510000, China, and also with the South China Technology Commercialization Center, Guangzhou 510000, China (e-mail: peiyuanlai@hotmail.com).

Qing-Yun Dai is with the School of Information Engineering, Guangdong University of Technology, Guangzhou 510000, China, also with the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data, Guangzhou 510000, China, and also with the Guangdong Polytechnic Normal University, Guangzhou 510000, China (e-mail: dqy@gpnu.edu.cn).

De-Zhang Liao and Xiao-Dong Liao are with the South China Technology Commercialization Center, Guangzhou 510000, China (e-mail: ldz@sctcc.cn; lxd@sctcc.cn).

Zhe-Rui Yang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: yangzhr9@mail2.sysu.edu.cn).

Chang-Dong Wang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China, and also with the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data, Guangzhou 510000, China (e-mail: chandongwang@hotmail.com).

Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510000, China (e-mail: minchen@ieee.org).

Recommended for acceptance by Z. Wang.

Digital Object Identifier 10.1109/TETCI.2024.3442872

I. Introduction

COMPLEX text recommendation is an important task in some situations because it is difficult to condense all the recommended item description into a short sentence. One typical example is the technology commercialization, which recommends technical demands (demands hereafter) of enterprises to technology researchers (researchers hereafter) [1]. In such task, the demands described by the complex text [2] have the characteristics of long text, dense information content, many technical keywords and abstract concepts. Moreover, the target users (i.e., researchers) also have complex text descriptions. Therefore, compared with the conventional item recommendation, the complex text recommendation is quite different. In the complex text recommendation, the challenging issues that must be considered usually include the feature characterizing, the path connectivity of representations of the texts, the attention weights of the representations, and the relationship weights between representations and recommended items.

There are currently some text-based recommendation methods [3], [4], [5], which analyze and represent text through Convolutional Neural Networks (CNNs), and then make recommendation based on representation. However, these methods still need to work on solving the problem of feature connection paths between complex texts and the relationship between representations and recommended items. On the one hand, these methods only leverage part of the information from users and items, yet ignore that the fusion of various enhanced semantic and structural information may bring better recommendation performance. On the other hand, the aforementioned models are mostly learned for generic recommendation tasks, which lack sufficient considerations for the specific task of complex text recommendation. The complex text refers to descriptive information with long content, numerous academic terms, and complex logic between texts, such as academic papers, technical patents, research achievements, etc.

In this paper, by taking the technology commercialization as an example, we propose a novel model termed **M**ulti-order **A**ttention and **S**emantic **E**nhanced **R**epresentation (MASER), which integrates four types of representations into a unified deep learning framework for complex text recommendation. The four types of representations include the structural and semantic representations of knowledge graph entities, and two types of attention weight based technical representations. As the preliminary of the MASER model, a technical keyword enhanced knowledge graph is firstly constructed from the detailed information of researchers and demands along with external data such as encyclopedia data. Then four modules are designed. The first module is the structural embedding learning module, which is used to learn the structural representation vectors of all the entities and relationships in the knowledge graph. The second module is the semantic embedding learning module, which is used to learn the semantic representation vectors of all the entities by their textual information. The third module is the relationship learning module based on multi-order attention mechanism, which can learn and train the relationship weights between researchers and demands' entities, resulting in the extended-keyword representation vectors and the original-keyword representation vectors for researchers and demands. The fourth module is the embedding fusion and recommendation prediction training. Experiments on real-world datasets demonstrate the highly competitive performance of our proposed approach over the state-of-the-art methods. It should be noticed that the proposed model is not limited to the technology commercialization, but is also applicable to other complex text recommendation tasks.

The main contributions of this paper are summarized as follows.

1) In this paper, a new recommendation problem called complex text recommendation is proposed, in which the target users and the items to be recommended are described by the complex text. Specifically, complex texts usually have long content, multiple logic levels, and numerous academic terms, making them difficult to read and represent.
2) A novel model termed **M**ulti-order **A**ttention and **S**emantic **E**nhanced **R**epresentation (MASER) is proposed, which integrates four types of information into a unified deep learning framework, including structural and semantic information of knowledge graph entities and two types of technical information with relationship weights.
3) Experiments on the real-world complex text dataset demonstrate the superior performances of our MASER approach in comparison with the state-of-the-art.

The rest of this paper is organized as follows. In Section II, we will briefly review the related work on text-based, semantic-enhanced and attention-based recommendations. In Section III, we will introduce the background of problem and emphasize its challenges. Further, we will describe in detail the proposed MASER method in Section IV. The experimental results will be reported in Section V. Finally, the paper will be concluded and the future work will be presented in Section VI.

II. Related Work

Because of the information overload, recommender systems have been applied to improve the user experience in different kinds of online platforms, such as product recommendation in e-commerce platforms [6], paper recommendation [7], music recommendation [8], and movie recommendation [9]. Collaborative filtering is one of the most popular approaches for recommender systems [10], which utilizes the similarity between users [11] or items [12] to predict the user preferences. Despite some success in providing relevant recommendations, these traditional models suffer from the data sparsity and cold start problems [10]. In order to achieve better recommendation performance, many methods have been explored by utilizing the multiple types of side information and interaction data, such as user profiles [13], item attributes [14], [15], item reviews [16], [17], and interactive information in the sequential context [18], [19]. Different kinds of recommendation methods based on text [20], semantics [21], knowledge graph [22], and attention mechanism [23] are proposed. It can be observed that recommendation algorithms exhibit their own advantages in different scenarios. In this section, we briefly review the literature related to this study, which can be divided into three sub-categories, i.e., text-based recommendation, semantic-enhanced recommendation and attention-based recommendation.

A. *Text-Based Recommendation*

There are many related studies on text-based recommendation, such as news recommendation and paper recommendation, where the recommended items are document-like or long text. In [3], Liu et al. deploy Bert [24] to extract the text-level features of representing papers and the AutoEncoder network is adopted to obtain the feature representation of each journal from the relationship matrix of the paper-journal bipartite graph. In [4], Ali et al. present a weighted probabilistic paper recommendation model termed PR-HNE by learning researchers' and papers' dynamics information network. Yao et al. [5] propose a knowledge graph entity embedding method based on text information, which adopts Bert [24] to learn the textual representation of entity text description, and triplets are used to train and predict the model. In [16], Du et al. propose the Hierarchical Attention Cooperative Neural Networks (HACN) model for recommendation, which adopts a hierarchical attention mechanism to enrich user's and item's feature representation from review texts.

Except for the above-mentioned text representation based on entities, some research works focus on the path-based relationship representation between text entities [25], [26]. Moreover, the texts of review in the e-commerce business are also used for recommendations. In [27], Liu et al. propose a hybrid neural recommendation model to learn the deep representations for users and items from both ratings and reviews, and then introduce a novel review-level attention mechanism incorporating with rating-based representation as query vector to select useful reviews. In [28], Yao et al. propose a deep learning recommendation model which integrates textual review sentiments and rating matrix, and combines cross-grained sentiment of reviews and user-item rating-based matrix factorization. Large language models have become pivotal in natural language processing, significantly impacting recommendation systems [29]. These models enhance recommendation quality by leveraging high-quality textual representations and comprehensive external knowledge to establish correlations between items and users. However, a fundamental challenge lies in accurately extracting and representing information from texts enhanced by large language models. This work addresses the critical issue, proposing solutions to improve the precision and effectiveness of recommendation systems through advanced text analysis. Although these methods have exhibited good performance in text-based recommendation, they differ significantly from the problem studied in this paper, e.g., their target users do not have textual descriptions.

B. Semantic-Enhanced Recommendation

Semantic-enhanced recommendation utilizes semantic information to improve the accuracy and relationship of recommendations, by using Natural Language Processing (NLP) and machine learning technology to understand the meaning and context of users, items, and interactions. Some efforts have been made in improving the recommendation performance from the perspective of semantic similarity [30]. It is a special type of recommendation with hybrid feature fusion, which is an enhanced strategy for better data representations of entity information, and widely used in various types of application. For example, Cantador et al. [31] propose a semantic enhanced knowledge representation for news recommendation, which incorporates the semantic-rich domain knowledge between user and item spaces. In [21], Yang proposes a novel recommender system based on semantic analysis and semantic awareness, which are responsible for solving the narrowness problem and the sensitivity to the semantic problem, respectively. Recently, Zhou et al. [32] address the problem of high-quality data representations in the conversational recommender systems by semantic fusion in word-level and entity-level semantic spaces. Similarly, Sun et al. [33] present a model termed Req2Lib, which is the first to explore the semantic information of requirement descriptions for software library recommendation. Specifically, Req2Lib adopts a sequence-to-sequence model to learn semantics and a pre-trained word2vec model for word embedding. However, Req2Lib fails to take into account attention mechanisms, which might better represent entity embedding information for the recommendation tasks.

Besides, many researchers also explore the recommendation algorithms based on the context, but they mainly extract the text information in the reviews and fuse the score and review information for prediction, so as to improve the performance of recommendation. For example, Chen et al. [34] propose a review-based recommendation model by separating user reviews into different sentiment orientations, and then generating the recommendation set through assigning voting rights according to similarities. In [35] Liu et al. extract the enhanced representation between the text reviews through the local and mutual attention within convolutional neural networks for the text-based recommendation. In [20] Khan et al. propose a novel recommender model which enriches the items content embedding with contextual features extracted through CNNs, to overcome both the issues of sparseness and loss due to negative values. In order to achieve high-performance recommendations in text-based recommendation, Wang et al. [25] combine the description of the entity text and the semantic information of the relationship into the path representation, and encode the expression by mining the sequence dependence of the path, so as to make the recommendation.

C. Attention-Based Recommendation

Besides, attention mechanisms have been widely used in recommender systems. For example, Wang et al. [36] propose a method termed knowledge graph attention network (KGAT), which updates a node's embedding by recursively embedding propagates from its neighbors with an attention weighted mechanism, and uses the attention weights to interpret the result of recommendation. Similarly, Wang et al. [37] aggregate the information of neighbors on the knowledge graph and propose a model termed RippleNet for recommendation. In [38], attention mechanism is adopted for fusing user-item latent vectors across different domains for cross-domain recommendation.

Although the above-mentioned methods have shown promising advantages in selecting important representation by attention mechanisms, its single attention layer or module is still under-represented and has room for improving performance in the recommendation tasks. To better explore the important information, many researchers have attempted to solve this problem by multi-attention mechanism. For instance, Chen et al. [39] propose an attention model to address the challenges of implicit feedback in multimedia recommendation, which is based on item-level and component-level attention mechanism. In addition, Sun et al. [40] present a network for temporal social recommendation by modeling users' complex preferences with the dynamic and static attention network. Recently, Ni et al. [41] apply multi-attention-based convolutional neural networks to represent user and item feature vectors for recommendation. Especially the multi-attention mechanism consists of self-attention and cross-attention, which refers to exploring the internal and mutual attention between users and items, respectively. Moreover, In [42] Wang et al. propose a hybrid deep collaborative filtering model that jointly learns rating embedding and textural feature from ratings and reviews respectively, and introduce two attention-based Gate Recurrent Unit (GRU) networks to learn context-aware representation as textural feature for users and items from reviews.

As recommendation system technologies advance and proliferate, the methods for extracting features from data have seen rapid development, driven by the need to tackle emerging challenges in various application scenarios. This evolution has led to the creation of numerous recommendation system models, which leverage features from the textual content and semantics of users and items, the relationships and weights of entities and attributes, and the fusion of features derived from interaction matrices. Despite these advancements, the challenge

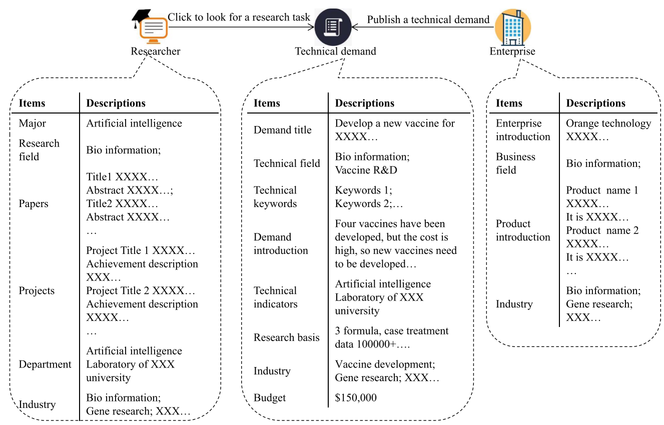


Fig. 1. Examples of description and schematic overview of the relationships among researchers, enterprises and demands.

of recommending based on complex texts remains unaddressed. However, in the above researches, the text-based models and the path-based models mostly ignore the important role of user interaction information in recommendations. Moreover, the text information and semantic enhancement are rarely used in the design of attention mechanism recommendations focusing on interactive information. The advent of large language models has introduced more sophisticated techniques for expanding the descriptive texts of users and items. However, while complex texts enrich entity features, they simultaneously introduce substantial noise. Accurately representing users and items within complex texts to enhance the performance of recommendation systems is a pressing new challenge. In this work, we present the problem of complex text recommendation and explore recommendation methods based on complex text, utilizing knowledge graphs and multi-layer attention mechanisms.

III. Background and Challenges

In this section, we will describe the background of research problem and the main challenges of complex text recommendation.

A. Problem Description

In some platforms, such as technology commercialization platforms, researchers usually fill in their technical achievements in detail to get accurate recommendations of demands from the platform. On the other hand, enterprise users usually fill in their demands in detail for seeking the most suitable researchers. As a result, the users leave a large amount of textual information on the platform, providing valuable information for recommending demands to researchers. Taking SCTCC.CN[1] as an example, as shown in Fig. 1, it contains two user roles: researcher and enterprise user. Researchers can publish various information in the SCTCC platform, including their major research fields, the technical achievements they have obtained (such as papers, research projects and patents), and their laboratories or departments, which can attract the attention of enterprises and hence promote their cooperation with enterprise users. As for enterprise users, they first provide their basic information, including identity authentication, enterprise profile, main product introduction and so on, and then release demands in the SCTCC platform. The detailed information of the demands includes the general description, technical indicators, economic indicators, research basis, etc. It is worth noting that they also publish the payment budget for each demand, which is essential for promoting cooperation with researchers.

In order to recommend demands to researchers accurately, we need to extract the technical keywords from the complex texts of the enterprises, their demands and the researchers, some of which are very critical while some are useless or even harmful. And the technical keywords may be the same or synonymous. Meanwhile, the importance of each extracted word to the different users may vary from one user to another. Therefore, in this scenario, a new problem called complex text recommendation arises, which focuses on important and valuable keywords within complex text representations, and ensures that the recommendation results meet the complex demands of different users.

[1]https://www.sctcc.cn/. It is a service platform for technology commercialization.

B. Challenges

In some recommendation scenarios, the long textual description can be processed by the existing methods, such as NLP for word segmentation and feature extraction. However, the technical keywords extracted from the context are usually messy, or even contain some harmful information, which results in the following challenges:

1) The first challenge is feature diversity. The description of researchers usually comes from their research achievements, i.e., papers or projects, and the relationship between these attribute words is primarily abstract and challenging to quantify. For example, when selecting one keyword for the researcher from `Computer Technology` and `Planting Technology`, the former should be selected since the keywords `Agricultural Big Data`, `Artificial Intelligence`, and `Data Analysis` are extracted respectively from the researcher's two papers. Therefore, How to choose more representative technical keywords among the extracted technical keywords is one of the most critical challenges.
2) The second challenge is path diversity. Most technical keywords in the researchers' descriptions have diverse meanings, especially for interdisciplinary researchers. For example, the term `Big Data` may mean medical Big Data or news Big Data, and the former may refer to researchers with doctor-related majors or computer science-related majors. The latter may refer to public opinion analysis. In most cases, these two technologies will not be mastered by one researcher. Therefore, it is not easy to accurately describe the semantics of `Big Data`, which makes the technical description of researchers face significant challenges.
3) The third challenge is relationship diversity. Since the descriptive text provides various information from

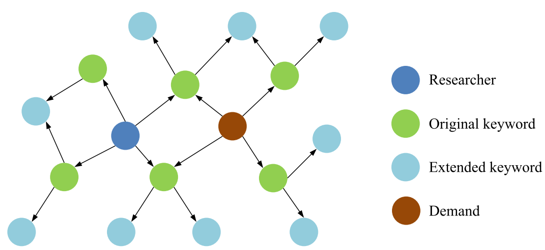


Fig. 2. The schematic diagram of the constructed knowledge graph.

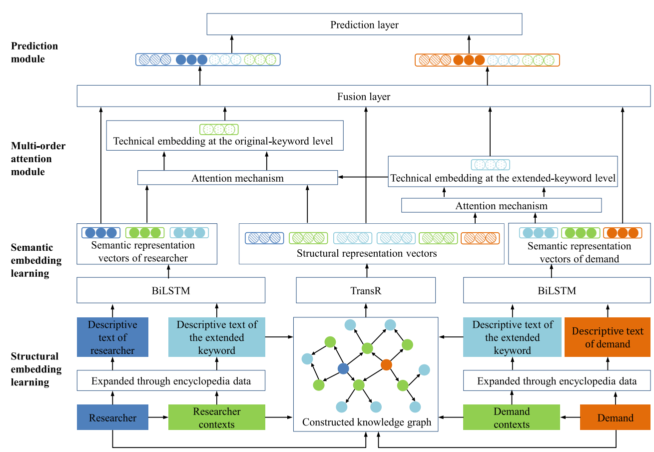


Fig. 3. Overview of the proposed MASER framework. It consists of the following four main modules, i.e. structural embedding learning module, semantic embedding learning module, relationship learning module and prediction module.

TABLE I
THE MAIN SYMBOLS USED IN THIS PAPER

Symbols	Definitions and descriptions
$\tilde{\mathbf{e}}_h$	Structural embedding vector of head entity h
$\tilde{\mathbf{e}}_t$	Structural embedding vector of tail entity t
$\bar{\mathbf{e}}_h$	Semantic embedding vector of head entity i
$\bar{\mathbf{e}}_t$	Semantic embedding vector of tail entity i
$\mathbf{e}_r$	Embedding vector of relationship r
W_i	The detailed textual information associated with entity i
$w_{i,j}$	The j-th word of the descriptive text of entity i
(h, r, t)	The triplet existing in the knowledge graph
(h, r, t')	The negative sample triplet that does not exist in the knowledge graph
$\mathbf{e}_i$	The embedding vector of entity i
$\mathbf{e}_{i,t,k}^{\text{ext}}$	The embedding of the researcher i's k-th extended technical keyword
$\mathbf{e}_{i,r,k}^{\text{ext}}$	The embedding of the relationship of the researcher i's and its k-th extended technical keyword
$\mathbf{e}_{i,t,k}^{\text{orig}}$	The embedding of the researcher i's k-th original technical keyword
$\mathbf{e}_{i,r,k}^{\text{orig}}$	The embedding of the relationship of the corresponding triplet of the researcher i's k-th original technical keyword
$\tilde{\mathbf{W}}$	The parameter matrix in structural embedding learning
$\bar{\mathbf{W}}$	The parameter matrix in semantic embedding learning
$\mathbf{A}_{i,j}$	Correlation of each extended keyword pair between the i-th researcher and the j-th demand
$\mathbf{f}_i$	The embedding of entity i's relationship at the extended-keyword level obtained by scaled dot-product attention mechanism, e.g. researcher or demand
$\mathbf{c}_i$	The embedding of entity i's at the extended-keyword level obtained by multi-order attention mechanism, e.g. researcher or demand
$\mathbf{o}_i$	The embedding of entity i's at the original level obtained by multi-order attention mechanism, e.g. researcher or demand
$\mathbf{m}_i$	The embedding of researcher i
$\mathbf{n}_j$	The embedding of demand j
Θ	Set of parameters to be trained in the model
λ	The regularization coefficient

multiple perspectives, the relationship between keywords and users is complex and diverse. In the SCTCC platform, both of the demands and the researchers usually contain multiple technical keywords. For example, one demand contains the keywords of `Big Data`, `Cancer Treatment`, and `Intelligent Prediction Technology`. Moreover, another demand contains `Remote Control`, `Unmanned Aerial Vehicles`, and `Agricultural Big Data`. Besides, the keywords of researchers are `Computers`, `Artificial Intelligence` and `Data Analysis`. Therefore, which demand should be recommended to the researcher first? This is the third challenge.

IV. THE PROPOSED MASER MODEL

In this section, we will first introduce the knowledge graph we construct as shown in Fig. 2, and then present the proposed MASER model in detail, which is illustrated in Fig. 3. It consists of the following four main modules. The first module is a structural embedding learning module, which is used to learn the structural representation vectors of all the entities and relationships in the knowledge graph. The second module is a semantic embedding learning module, which is used to learn the semantic representation vectors of all the entities by their textual information. The third module is a relationship learning module based on multi-order attention mechanism, which can learn and train the relationship weights between researchers and demands' entities, resulting in the extended-keyword representation vectors and the original-keyword representation vectors for researchers and demands. And the fourth module is a prediction module which calculates the matching score between researchers and demands by fusing the four vectors obtained previously (i.e. structural representation vectors, semantic representation vectors, extended-keyword representation vectors, and original-keyword representation vectors).

For clarity, the main symbols used in this paper are summarized in Table I.

A. Construction of Knowledge Graph

In order to make full use of the detailed information of researchers and demands, we construct a technical keyword enhanced knowledge graph from the detailed information of researchers and demands along with external data such as encyclopedia data. Suppose the detailed textual information associated with a researcher is represented as $\{w_1, w_2, ..., w_l\}$, where l indicates the number of words in the textural information, which is obtained from the researcher's publications such as papers, projects, patents, and so on. We employ keyword extraction tools such as Jieba[2] to extract keywords from the researcher's detailed

[2]https://github.com/fxsjy/jieba

textual information. Similarly, we extract technical keywords from the descriptive texts associated with demands.

To explore the relationship between researchers and demands, we treat these extracted keywords as the original technical keywords (original keywords hereafter) and use them as indices to extract related keywords from encyclopedia data. Specifically, we link the aforementioned original keywords to encyclopedia data and retrieve extended descriptive texts corresponding to these original keywords. After that, we utilize keyword extraction tools to further extract technical keywords from these extended descriptive texts. These extracted keywords from the extended descriptive texts are regarded as the extended technical keywords (extended keywords hereafter). Finally, we construct the knowledge graph, as shown in Fig. 2, using triples of the form (*researcher*, *relation*, *original keyword*), (*demand*, *relation*, *original keyword*) and (*original keyword*, *relation*, *extended keyword*). Additionally, we crawl descriptive texts for each keyword from encyclopedia data, so there is a corresponding descriptive text for each entity on the knowledge graph.

B. Structural Embedding Learning

To learn the representations of researchers and demands, we utilize knowledge graph embedding techniques to parameterize entities and relationships in the knowledge graph as vector representations. Specifically, we adopt TransR [43] to capture the structural information of entities and relationships in the knowledge graph. Assume that $\tilde{\mathbf{e}}_h + \mathbf{e}_r \approx \tilde{\mathbf{e}}_t$ if the triplet (h, r, t) exists in the knowledge graph, where $\tilde{\mathbf{e}}_h$ and $\tilde{\mathbf{e}}_t$ are the structural embedding vectors of the head entity h and the tail entity t respectively, and $\mathbf{e}_r$ is the embedding vector of relationship r. Since entities and relationships are different objects, the embedding vectors of entities and relationships may be in different embedding spaces. Therefore, TransR projects the entity embeddings into the relationship embedding space as follows,

$$\tilde{\mathbf{W}}\tilde{\mathbf{e}}_h + \mathbf{e}_r \approx \tilde{\mathbf{W}}\tilde{\mathbf{e}}_t \tag{1}$$

where $\tilde{\mathbf{e}}_h \in \mathbb{R}^{d_1\times 1}$, $\tilde{\mathbf{e}}_t \in \mathbb{R}^{d_1\times 1}$, $\mathbf{e}_r \in \mathbb{R}^{d_2\times 1}$, and $\tilde{\mathbf{W}} \in \mathbb{R}^{d_2\times d_1}$ is a learnable matrix projecting the entity structural embedding into the relationship embedding space.

TransR is able to distinguish whether a triplet (h, r, t) exists in the knowledge graph, and thus we define the score function as follows,

$$\tilde{g}(h, r, t) = \left\|\tilde{\mathbf{W}}\tilde{\mathbf{e}}_h + \mathbf{e}_r - \tilde{\mathbf{W}}\tilde{\mathbf{e}}_t\right\|_2^{ext} \tag{2}$$

According to the hypothesis, if the triplet (h, r, t) exists in the knowledge graph, the value $\tilde{g}(h, r, t)$ will be relatively small. Otherwise, it will be relatively large.

C. Semantic Embedding Learning

The entity structural embeddings learned through TransR only contain structural information. However, considering only the structural information of researchers and demands may not achieve satisfactory recommendation performance. For each entity i in the knowledge graph, there is a corresponding descriptive text $W_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,l}]$, where $w_{i,j}$ is the j-th word of the descriptive text of entity i. Therefore, we utilize the semantic information of entities to enhance their embeddings. Specifically, we use the pre-trained BERT [24] to initialize each word in the descriptive text and obtain word embeddings as follows,

$$\mathbf{W}_i = [\mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \ldots, \mathbf{w}_{i,l}] \tag{3}$$

where $\mathbf{w}_{i,j} \in \mathbb{R}^{d_3}$ is the embedding vector of $w_{i,j}$ obtained by BERT.

To learn the semantic information of the text, we use BiLSTM to learn the semantic embeddings. Specifically, we utilize the word embeddings as input to BiLSTM and obtain the semantic embeddings of entities as follows,

$$\bar{\mathbf{e}}_i = \text{BiLSTM}(\mathbf{W}_i) \tag{4}$$

where $\bar{\mathbf{e}}_i \in \mathbb{R}^{d_1}$ is the semantic embedding of entity i. Similarly, we adopt TransR to train the semantic embeddings of entities, which assumes that $\bar{\mathbf{W}}\bar{\mathbf{e}}_h + \mathbf{e}_r \approx \bar{\mathbf{W}}\bar{\mathbf{e}}_t$ if the triplet (h, r, t) exists in the knowledge graph, where $\bar{\mathbf{W}} \in \mathbb{R}^{d_2\times d_1}$ is a learnable matrix projecting the entity semantic embedding into the relationship embedding space.

Similarly, we encourage distinguishing whether a triplet exists in the knowledge graph and define the score function as follows,

$$\bar{g}(h, r, t) = \left\|\bar{\mathbf{W}}\bar{\mathbf{e}}_h + \mathbf{e}_r - \bar{\mathbf{W}}\bar{\mathbf{e}}_t\right\|_2^{\text{ext}} \tag{5}$$

According to the hypothesis, if there are triples (h, r, t) in the knowledge graph, the value $\bar{g}(h, r, t)$ will be relatively small. Otherwise, the value $\bar{g}(h, r, t)$ is relatively large, which means that the triplet does not exist and refers to a negative sample. In the representation learning process of knowledge graph, we need to consider both structural information and semantic information. Therefore, the objective score function is defined as follows,

$$g(h, r, t) = \tilde{g}(h, r, t) + \bar{g}(h, r, t) \tag{6}$$

If the triplet (h, r, t) exists in the knowledge graph, the score value of $g(h, r, t)$ is small. Otherwise, the score value is large. To encourage the method to distinguish whether the triplet (h, r, t) exists in the knowledge graph, we define the loss function as follows,

$$\mathcal{L}_{KG} = \sum_{(h,r,t,t')\in\Gamma} -\ln\sigma\left(g\left(h, r, t'\right) - g(h, r, t)\right) \tag{7}$$

where Γ indicates the training set, (h, r, t) is the triplet existing in the knowledge graph, and (h, r, t') denotes the constructed negative sample, i.e., the triplet that does not exist in the knowledge graph. In general, a negative sample is an invalid triplet constructed by randomly replacing one of the entities in the valid triple.

After training, the structural embedding and semantic embedding of entity are concatenated as the entity's embedding representation. Therefore, for a triplet (h, r, t) in the knowledge graph, its embedding representation is expressed as follows,

$$(\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t) = \left([\tilde{\mathbf{e}}_h; \bar{\mathbf{e}}_h], \mathbf{e}_r, [\tilde{\mathbf{e}}_t; \bar{\mathbf{e}}_t]\right) \tag{8}$$

where $\mathbf{e}_h \in \mathbb{R}^{2d_1 \times 1}$, $\mathbf{e}_r \in \mathbb{R}^{d_2 \times 1}$ and $\mathbf{e}_t \in \mathbb{R}^{2d_1 \times 1}$ are the embeddings of the head entity h, the relationship r and the tail entity t respectively.

D. Relationship Learning Module Based on Multi-Order Attention Mechanism

In the recommendation platform, in addition to considering the preferences of users, it is even more important to consider the relationship of their embeddings between users and items by historical interaction data. Therefore, a multi-order attention mechanism is designed to capture the relationship between researchers and demands. For clarity, we take researcher i and demand j as examples.

1) Scaled Dot-Product Attention Mechanism: An attention mechanism is designed to capture the extended-keyword level of relationship between researchers and demands. Firstly, we calculate the similarity between the extended keywords of researchers and the extended keywords of demands. Specifically, we concatenate the embeddings of the extended keywords of researchers along with the embeddings of their corresponding relationships as follows,

$$\mathbf{X}_i = [\mathbf{E}_i; \mathbf{R}_i] \in \mathbb{R}^{(2d_1+d_2)\times p} \tag{9}$$

$$\mathbf{E}_i = \left[\mathbf{e}^{\text{ext}}_{i,t,1}, \mathbf{e}^{\text{ext}}_{i,t,2}, \ldots, \mathbf{e}^{\text{ext}}_{i,t,p}\right] \in \mathbb{R}^{2d_1\times p} \tag{10}$$

$$\mathbf{R}_i = \left[\mathbf{e}^{\text{ext}}_{i,r,1}, \mathbf{e}^{\text{ext}}_{i,r,2}, \ldots, \mathbf{e}^{\text{ext}}_{i,r,p}\right] \in \mathbb{R}^{d_2\times p} \tag{11}$$

where p is the number of the researcher i's extended keywords, $\mathbf{e}^{\text{ext}}_{i,t,k}$ is the embedding of the researcher i's k-th extended keyword, and $\mathbf{e}^{\text{ext}}_{i,r,k}$ is the embedding of the relationship in the triplet connecting the researcher i's k-th extended keyword and the corresponding original keyword. For the case with more than one triplets, the relationship in one randomly selected triplet will be used.

Similarly, we concatenate the embeddings of the extended keywords of demands along with the embeddings of their corresponding relationships as follows,

$$\mathbf{Y}_j = [\mathbf{E}_j; \mathbf{R}_j] \in \mathbb{R}^{(2d_1+d_2)\times q} \tag{12}$$

$$\mathbf{E}_j = \left[\mathbf{e}^{\text{ext}}_{j,t,1}, \mathbf{e}^{\text{ext}}_{j,t,2}, \ldots, \mathbf{e}^{\text{ext}}_{j,t,q}\right] \in \mathbb{R}^{2d_1\times q} \tag{13}$$

$$\mathbf{R}_j = \left[\mathbf{e}^{\text{ext}}_{j,r,1}, \mathbf{e}^{\text{ext}}_{j,r,2}, \ldots, \mathbf{e}^{\text{ext}}_{j,r,q}\right] \in \mathbb{R}^{d_2\times q} \tag{14}$$

where q is the number of the demand j's extended keywords, $\mathbf{e}^{ext}_{j,t,k}$ is the embedding of the demand j's k-th extended keyword, and $\mathbf{e}^{\text{ext}}_{j,r,k}$ is the embedding of the relationship in the triplet connecting the demand j's k-th extended keyword and the corresponding original keyword. For the case with more than one triplets, the relationship in one randomly selected triplet will be used.

Next, we adopt the scaled dot-product attention mechanism [44] to calculate the similarity between $\mathbf{X}_i$ and $\mathbf{Y}_j$ as the weight of attention as follows,

$$\mathbf{Q}_i = \mathbf{W}_Q \mathbf{X}_i \tag{15}$$

$$\mathbf{K}_j = \mathbf{W}_K \mathbf{Y}_j \tag{16}$$

$$\mathbf{A}_{i,j} = \frac{\mathbf{Q}_i^T \mathbf{K}_j}{\sqrt{d_k}} \tag{17}$$

where $\mathbf{W}_Q \in \mathbb{R}^{d_k\times(2d_1+d_2)}$ and $\mathbf{W}_K \in \mathbb{R}^{d_k\times(2d_1+d_2)}$ are learnable parameter matrices. Besides, $\mathbf{A}_{i,j} \in \mathbb{R}^{p\times q}$ refers to the correlation of each extended keyword pair between the i-th researcher and the j-th demand.

Finally, we aggregate the embeddings of the extended keywords of researchers based on the attention weights to obtain the embeddings of researchers' relationship as follows,

$$\mathbf{V}_i = \mathbf{W}_v^1 \mathbf{X}_i \tag{18}$$

$$\mathbf{F}_i = \text{Softmax}\left(\mathbf{A}_{i,j}^T\right)\mathbf{V}_i^T \tag{19}$$

$$\mathbf{f}_i = \text{MeanPooling}\left(\mathbf{F}_i\right) \tag{20}$$

where $\mathbf{W}_v^1 \in \mathbb{R}^{d_k\times(2d_1+d_2)}$ is the learnable parameter matrix, Softmax() indicates the softmax normalization performed on each row of the matrix, MeanPooling() indicates the average pooling layer, and $\mathbf{f}_i$ indicates the embedding of researcher i's relationship.

Similarly, we aggregate the embeddings of the extended keywords of demands based on the attention weights to obtain the embeddings of demands' relationship as follows,

$$\mathbf{V}_j = \mathbf{W}_v^2 \mathbf{Y}_j \tag{21}$$

$$\mathbf{F}_j = \text{Softmax}\left(\mathbf{A}_{i,j}\right)\mathbf{V}_j^T \tag{22}$$

$$\mathbf{f}_j = \text{MeanPooling}\left(\mathbf{F}_j\right) \tag{23}$$

where $\mathbf{W}_v^2 \in \mathbb{R}^{d_k\times(2d_1+d_2)}$ and $\mathbf{f}_j$ indicates the embedding of demand j's relationship.

2) Multi-Order Attention Mechanism: To further explore the correlation between researchers and demands, a multi-order attention mechanism is designed to capture the technical embeddings of researchers and demands. Specifically, we utilize the $\mathbf{f}_i$ obtained by the attention mechanism in the previous section as guidance to calculate the attention weights for each extended technical keyword of researcher i. We then aggregate the embeddings of the extended technical keywords of researcher i to obtain the technical embedding of researcher i at the extended-keyword level as follows,

$$p^{\text{ext}}_{i,k} = \frac{\exp\left(\mathbf{f}_i \mathbf{W}_f \left[\mathbf{e}^{\text{ext}}_{i,t,k}; \mathbf{e}^{\text{ext}}_{i,r,k}\right]\right)}{\sum_{k=1}^{p} \exp\left(\mathbf{f}_i \mathbf{W}_f \left[\mathbf{e}^{\text{ext}}_{i,t,k}; \mathbf{e}^{\text{ext}}_{i,r,k}\right]\right)} \tag{24}$$

$$\mathbf{c}_i = \sum_{k=1}^{p} p^{\text{ext}}_{i,k} \left[\mathbf{e}^{\text{ext}}_{i,t,k}; \mathbf{e}^{\text{ext}}_{i,r,k}\right] \tag{25}$$

where $\mathbf{W}_f \in \mathbb{R}^{d_k\times(2d_1+d_2)}$ is the parameter matrix. $p^{\text{ext}}_{i,k}$ indicates the weight of the k-th extended technical keyword for researcher i and $\mathbf{c}_i$ indicates the technical embedding of researcher i at the extended-keyword level.

Next, we utilize $\mathbf{c}_i$ as guidance to calculate the attention weights for each original technical keyword of researcher i. We then aggregate the embeddings of researcher i's original technical keywords to obtain the technical embedding of researcher i

at the original level as follows,

$$p_{i,k}^{\text{orig}} = \frac{\exp\left(\mathbf{c}_i \mathbf{W}_c \left[\mathbf{e}_{i,t,k}^{\text{orig}}; \mathbf{e}_{i,r,k}^{\text{orig}}\right]\right)}{\sum_{k=1}^{m} \exp\left(\mathbf{c}_i \mathbf{W}_c \left[\mathbf{e}_{i,t,k}^{\text{orig}}; \mathbf{e}_{i,r,k}^{\text{orig}}\right]\right)} \tag{26}$$

$$\mathbf{o}_i = \sum_{k=1}^{m} p_{i,k}^{\text{orig}} \left[\mathbf{e}_{i,t,k}^{\text{orig}}; \mathbf{e}_{i,r,k}^{\text{orig}}\right] \tag{27}$$

where m indicates the number of researcher i's original technical keywords, $\mathbf{e}_{i,t,k}^{\text{orig}}$ is the embedding of the researcher i's k-th original technical keyword, $\mathbf{e}_{i,r,k}^{\text{orig}}$ is the embedding of the relationship of the corresponding triplet of the researcher i's k-th original technical keyword, $\mathbf{W}_c \in \mathbb{R}^{(2d_1+d_2)\times(2d_1+d_2)}$ is the parameter matrix, $p_{i,k}^{\text{orig}}$ indicates the weight of the k-th original technical keyword for researcher i, and $\mathbf{o}_i$ indicates the technical embedding of researcher i at the original-keyword level.

Similarly, for the demand j, using the same approach, we obtain the technical embedding at the extended-keyword level denoted as $\mathbf{c}_j$, and the technical embedding at the original level denoted as $\mathbf{o}_j$.

E. Prediction and Training

To achieve high-performance recommendation, we integrate the learned information from multiple aspects to obtain the embedding of researchers and demands. Specifically, we concatenate the structural embedding and semantic embedding of researchers and demands, along with their technical embeddings at both the extended-keyword level and the original level. We then apply a linear layer mapping to obtain the embedding representations of researchers and demands as follows,

$$\mathbf{m}_i = \mathbf{W}_m \left[\mathbf{e}_{i,s}; \mathbf{e}_{i,d}; \mathbf{c}_i; \mathbf{o}_i\right] \tag{28}$$

$$\mathbf{n}_j = \mathbf{W}_n \left[\mathbf{e}_{j,s}; \mathbf{e}_{j,d}; \mathbf{c}_j; \mathbf{o}_j\right] \tag{29}$$

where $\mathbf{W}_m \in \mathbb{R}^{d'\times(6d_1+2d_2)}$ and $\mathbf{W}_n \in \mathbb{R}^{d'\times(6d_1+2d_2)}$ are parameter matrices, $\mathbf{m}_i$ indicates the embedding of the researcher i, and $\mathbf{n}_j$ indicates the embedding of the demand j.

Next, we calculate the inner product of the embeddings of researchers and demands to predict their matching scores as follows,

$$\hat{r}(i,j) = \mathbf{m}_i^T \mathbf{n}_j \tag{30}$$

And the objective function is defined as follows,

$$\mathcal{L}_{CF} = \sum_{(i,j^+,j^-)\in O} -\ln \sigma\left(\hat{r}\left(i,j^+\right) - \hat{r}\left(i,j^-\right)\right) \tag{31}$$

where $O = \{(i,j^+,j^-)\}$ denotes the training set consisting of $(i,j^+) \in$ positive examples and $(i,j^-) \in$ negative examples.

Finally, we define the final objective function as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{KG} + \lambda_2 \mathcal{L}_{CF} + \lambda \left\|\Theta\right\|_2^1 \tag{32}$$

where Θ indicates the set of parameters of the model, λ indicates the regularization coefficient, and the last term of the loss function is adopted to prevent overfitting.

TABLE II
STATISTICS OF THE SCTCC DATASET

	Researchers	Enterprises	Demands
#Entities	2,882	1,880	1,410
#Original keywords	299,766	61,161	61,906
#External keywords	1,184,883	757,079	921,698
#Interactions	16,107	-	-

V. EXPERIMENTS

In this section, we will conduct extensive experiments on real-world historical interaction data between researchers and enterprises in the SCTCC platform to evaluate the proposed MASER model. First, we will introduce the setting of our experiments, including the real-world datasets, implementation details and evaluation measures. Then, we will conduct the parameter analysis and analyse the comparison results. Besides, we will conduct the ablation study to evaluate the effectiveness of our model.

A. Experimental Setting

1) Datasets: The SCTCC dataset is a researcher-demand recommendation dataset that is obtained from the SCTCC platform. Based on the operational data of the SCTCC platform, researchers need to fill in accurate materials in the platform and accept the review and revision requirements, which can ensure the accuracy of information. Similarly, enterprises and their demands also need to be filled in accurately and reviewed multiple times. As shown in Fig. 1, the dataset of the SCTCC platform contains some descriptions of researchers, enterprises, and their demands.

The statistics of the SCTCC dataset are described in Table II. The dataset contains 2,882 researchers, 1,880 enterprises, and 1,410 demands. By extracting information from the content filled by all the users, many original entities are formed, including 299,766 attribute entities of researchers, 61,161 attribute entities of enterprises and 61,906 attribute entities of demands, respectively. Next, each original entity is expanded by the encyclopedia data, and we further extract information from the encyclopedia text to form a large number of extended entities, containing 1,184,883 entities for researchers, 757,079 entities for enterprises and 921,698 entities for demands respectively. In addition, we have 16,107 interaction data that researchers browse demands of the enterprises.

2) Implementation Details: As shown in Fig. 4, we illustrate the implementation details of the data initialization and processing. The entities of the target users (i.e. researchers) and the recommended items (i.e. demands) are related to a large number of original technical keywords and extended technical keywords. Therefore, after data initialization, we randomly adopt the technical keywords as input to the knowledge graph. That is, m original technical keywords are sampled for each researcher and demand, and then p extended technical keywords are adopted for each original technical keyword. In our experiment, we set m to 3 and p to 5.

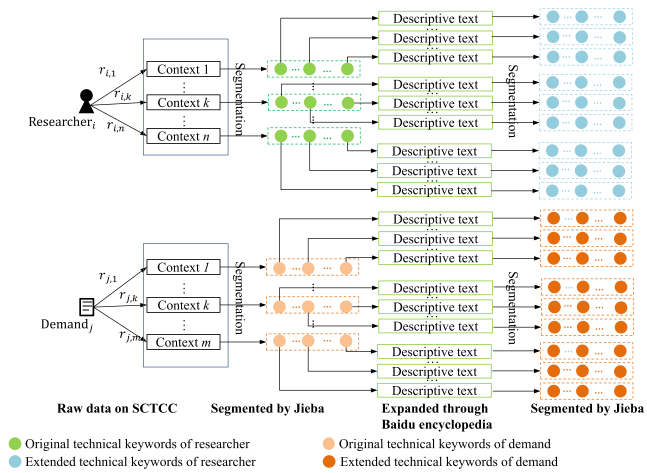
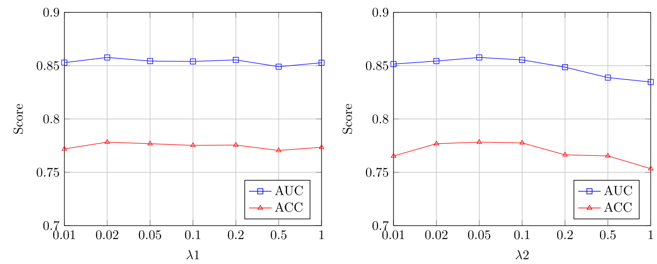


Fig. 4. Implementation details of the data initialization and processing.



(a) AUC (b) ACC

Fig. 5. Parameter analysis: The impact of the parameters λ_1 and λ_2 on the performance of MASER in terms of AUC and ACC on the SCTCC dataset.

TABLE III

PARAMETER ANALYSIS: THE IMPACT OF THE DIMENSION (DIM) ON THE PERFORMANCE OF MASER, INCLUDING ENTITY DIMENSION, RELATIONSHIP DIMENSION, HIDDEN DIMENSION AND OUTPUT DIMENSION

#Dim	Entity dim		Relation dim		Hidden dim		Output dim	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
16	0.7902	0.7229	0.8398	0.7650	0.8536	0.7805	0.8199	0.7414
32	0.8049	0.7316	0.8367	0.7645	0.8522	0.7713	0.8297	0.7509
64	0.8195	0.7449	0.8459	0.7757	0.8523	**0.7833**	0.8386	0.7632
128	0.8385	0.7620	0.8460	**0.7775**	0.8460	0.7775	0.8460	0.7775
256	**0.8459**	**0.7757**	**0.8518**	0.7694	**0.8541**	0.7786	0.8517	**0.7811**
512	0.8439	0.7629	0.8481	0.7637	0.8533	0.7735	**0.8536**	0.7675

During the training, we set the embedding dimension of the entity and relationship of the knowledge graph, hidden layer dimension and output layer dimension to 256. In addition, the learning rates of knowledge embedding and recommendation tasks are set to 0.001 and 0.0001, respectively. For the setting of weight parameter value in the loss function, we set λ_1 and λ_2 as 0.02 and 0.05. Besides, we set λ as 0.001in our experiments. All experiments are conducted on a single NVIDIA Tesla A100.

3) Evaluation Measures: In the experiments, we adopt two widely used evaluation measures, namely, the Area Under the ROC Curve (AUC) [45] and Accuracy (ACC) [46]. AUC measures all operating points while ACC is a measure at a single operating point or decision threshold on the model's ROC curve. Note that larger values of AUC and ACC indicate better recommendation results. In addition, in order to rule out the factor of getting occasionally, we conduct five times for every test method in each experiment, and then report their average AUC and ACC.

B. Parameter Analysis

In this subsection, we will analyze the impact of some parameters, namely, λ_1 and λ_2 of the loss function and the dimension of entity, relationship, hidden layer and output layer, on the performance of the proposed MASER model.

The parameters λ_1 and λ_2 denote the weight of the module represented by knowledge graph embedding and the recommendation matching loss, respectively. By trial and error, we analyze their values and tune them in {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1}. As shown in Fig. 5, it is found that the proposed MASER model can obtain the best AUC and ACC when $\lambda_1 = 0.02$ and $\lambda_2 = 0.05$. Therefore, in the following experiments, we set λ_1 and λ_2 as 0.02 and 0.05, respectively.

For the parameter dimension, we conduct experiments on the dimension of entity, relationship, hidden layer and output layer, respectively. In the experiment, we run the MASER model on the SCTCC dataset by selecting the dimension from {16, 32, 64, 128, 256, 512} while fixing the other parameters. The experimental results are shown in Table III.

The entity dimension and relationship dimension are important parameters in entity embedding representation of knowledge graph. The other two dimensions also affect the final model performance. From the results in Table III, on the one hand, it is clear that the model can achieve the best result when the entity dimension is set to 256. On the other hand, for the other three dimensions, although setting $dim = 256$ can not achieve the best performance in terms of both two evaluation measures, on the whole, it can ensure that one of the measures is the best with the other measure being close to the best effect. In summary, we set the four dimension parameters to 256 on the SCTCC dataset in our experiments.

C. Comparison Experiments

In this subsection, we evaluate the performance of the proposed MASER method against five baseline methods.[3]

1) **Knowledge graph attention network (KGAT)** [36]: It is an algorithm which updates a node's embedding by recursively propagating embedding from its neighbors with an attention weighted mechanism.
2) **Knowledge graph convolutional networks (KGCN)** [47]: It is an end-to-end recommendation method, which

[3]The codes of the baselines are downloaded from the following websites:
- KGCN: https://github.com/hwwang55/KGCN;
- KGAT: https://github.com/xiangwang1223/knowledge_graph_attention_network;
- LightGCN: https://github.com/kuandeng/LightGCN;
- SGL: https://github.com/AoiDragon/SGL_reproduction;
- NCF: https://github.com/hexiangnan/neural_collaborative_filtering;
- RippleNet: https://github.com/hwwang55/RippleNet;
- LightGCL: https://github.com/HKUDS/LightGCL.

TABLE IV
COMPARISON RESULTS ON THE SCTCC DATASET

Methods	SCTCC AUC	SCTCC ACC
KGAT [36]	<u>0.8207</u>	0.6803
KGCN [47]	0.6450	0.6030
LightGCN [48]	0.8068	<u>0.6967</u>
SGL [49]	0.5634	0.5572
NCF [50]	0.7292	0.6607
RippleNet [37]	0.7395	0.6879
LightGCL [51]	0.6429	0.5572
MASER	**0.8577**	**0.7783**
%Improv.	4.51%	11.71%

The best scores of AUC and ACC are in bold, and the second best scores are underlined. %improv. $= \frac{\text{best score}-\text{second best score}}{\text{second best score}} \times 100\%$.

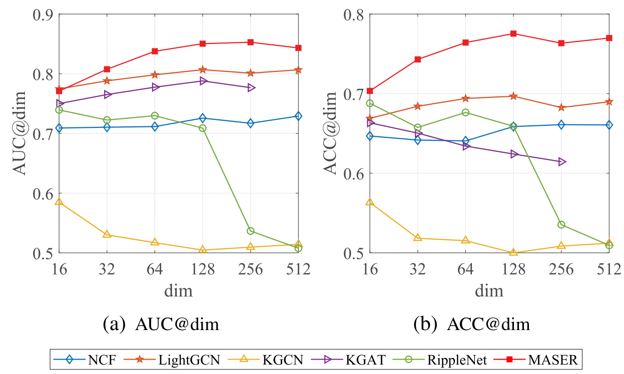


Fig. 6. Comparison results on the SCTCC dataset in terms of AUC and ACC in different entity dimensions. Note that KGAT [36] does not support 512 dimensions.

first samples the neighbors of each entity as receptive field, and then represents the embedding of given entity by combining neighborhood information with bias.

3) **Light graph convolutional network (LightGCN)** [48]: It is a simple and linear model which only aggregates neighborhood with weighted embeddings learned based on graph convolution network for recommendation.
4) **Self-supervised Graph Learning (SGL)** [49]: The method is the enhanced model of LightGCN. It explores self-supervised learning on user-item graph by generating multiple views of a node, so as to maximize the agreement between different views of the same node compared to that of other nodes.
5) **Neural collaborative filtering (NCF)** [50]: It is a method which learns the function from user-item data by replacing the inner product via a neural network.
6) **RippleNet** [37]: It is an end-to-end framework for knowledge-graph-aware recommender system. In RippleNet, the user preferences are propagated by iteratively extending the user's potential interest in knowledge graph, which is motivated by the way of ripples propagating on the water.
7) **Light Graph Contrastive Learning (LightGCL)** [51]: The method is a simple graph contrastive learning paradigm that exclusively utilizes singular value decomposition for contrastive augmentation, which enables the unconstrained structural refinement with global collaborative relation modeling.

In order to fairly conduct the performance comparison of all methods, including the above baselines and our MASER method, we adopt the same settings for the common parameters and tune the method-specific parameters as suggested by their corresponding papers. In addition, we also fine tune some common parameters for the baselines to achieve better competitive results. The experimental results *w.r.t.* AUC and ACC by different methods are reported in Table IV.

The performances are reported in terms of AUC and ACC on the SCTCC dataset. As shown in Table IV, we can see that the proposed MASER model outperforms all the baselines with a large margin. In particular, our model has achieved at least 4.51% improvement in terms of AUC, and a significant improvement as large as 11.71% in terms of ACC than the best baseline. This is because our model integrates richer semantic information compared with other models, namely, the structural information, semantic information and attention weight information, so as to provide more accurate embedding representations and obtain more significant improvement for recommendation. In light of the intricate semantic nuances and information extraction challenges in complex textual data, the performance of many models remains suboptimal. This deficiency extends to SGL, notwithstanding its utilization of different views of the same node compared to that of other nodes for learning, as it also yields merely moderate efficacy.

In addition, we also conduct comparison experiments on the entity dimension parameters of various baseline methods, and the experimental results are shown in Fig. 6. It can be seen from the experimental results that our model is superior to other baseline models in all kinds of entity dimension, which also indicates that our model does not depend on the limitation of entity dimension and is superior to other models as a whole.

D. Ablation Study

In order to further investigate the effectiveness of the key components in our MASER model, we conduct ablation study on the impact of using semantic information and the extended keyword information, respectively. First, we remove the semantic representation vector learning module, which is named MASER-Semantic. It directly uses the semantic information of entity for learning semantic representation. Another ablation study is conducted on the extended keyword information module, namely MASER-Extended keyword, which learns the extended word embedding for recommendation.

The results are listed in Table V. The results show that MASER significantly outperforms MASER-Semantic and MASER-Extended keyword, and MASER-Extended keyword performs better than MASER-Semantic. On the one hand, by comparing MASER-Semantic and MASER, MASER-Semantic has obtained 13.23% decrease on AUC and 13.06% decrease on ACC compared with MASER, which confirms the necessity of

TABLE V
ABLATION STUDY ON THE SCTCC DATASET

Methods	SCTCC	
	AUC	ACC
MASER-Semantic	0.7575 $\downarrow_{13.23\%}$	0.6884 $\downarrow_{13.06\%}$
MASER-Extended keyword	0.8497 $\downarrow_{0.94\%}$	0.7738 $\downarrow_{0.58\%}$
MASER	**0.8577**	**0.7783**

utilizing the semantic representation vector learning for making a better recommendation. On the other hand, by comparing MASER-Extended keyword and MASER, although the performance degradation is not as obvious as the former, it is also verified that the extended keyword information embedding can further improve the semantic representation, so as to achieve better recommendation performance.

VI. Conclusion and Future Work

Complex text recommendation is quite different from the conventional recommendation. In particular in the case of technology commercialization, which recommends technical demands of enterprises to technology researchers, both of which are composed of complex text. To address the unsolved challenges associated with complex text recommendation, in this paper, we have proposed a new model termed **M**ulti-order **A**ttention and **S**emantic **E**nhanced **R**epresentation (MASER), which integrates four types of information into a unified deep learning framework, namely structural and semantic information of knowledge graph entities and two types of technical information. By integrating additional information into text vector representationsuch as structural relationship information for extended keywords, and semantic information for entity description textsthe proposed model enhances complex text recommendation effectiveness significantly. Extensive experiments have been conducted on real-world complex text datasets to confirm the superiority of the proposed model in technology commercialization.

Although MASER has achieved promising performance by fusing semantic-enhanced representation and multi-order attention mechanism, it has not yet provided explanations for recommendations due to the complexity arising from keywords extraction and representation, as well as their relationship and weights. In the future, we can consider combining the latest research achievement of the NLP technology to further improve the performance of the model. For example, on the one hand, we can use word semantic disambiguation or word semantic clustering technology to remove non-technical keywords, or use heterogeneous information networks or multimodal representation techniques to further enhance the representations of the complex text. On the other hand, the matching performance of complex texts can be improved through knowledge augmentation by introducing more external knowledge.

References

[1] Z.-R. Yang, Z.-Y. He, C.-D. Wang, P.-Y. Lai, D.-Z. Liao, and Z.-Z. Wang, "A bi-directional recommender system for online recruitment," in *Proc. IEEE Int. Conf. Data Mining*, 2022, pp. 628–637.

[2] Y. Li, B. Hu, Y. Ding, L. Ma, and M. Zhang, "A neural divide-and-conquer reasoning framework for image retrieval from linguistically complex text," in *Proc. Assoc. Comput. Linguistics*, 2023, pp. 16464–16476.

[3] C. Liu, X. Wang, H. Liu, X. Zou, S. Cen, and G. Dai, "Learning to recommend journals for submission based on embedding models," *Neurocomputing*, vol. 508, pp. 242–253, 2022.

[4] Z. Ali, G. Qi, K. Muhammad, B. Ali, and W. A. Abro, "Paper recommendation based on heterogeneous network embedding," *Knowl. Based Syst.*, vol. 210, 2020, Art. no. 106438.

[5] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," 2019, *arXiv:1909.03193*.

[6] Z. Zhan and B. Xu, "Analyzing review sentiments and product images by parallel deep nets for personalized recommendation," *Inf. Process. Manag.*, vol. 60, no. 1, 2023, Art. no. 103166.

[7] A. Chaudhuri, M. Sarma, and D. Samanta, "SHARE: Designing multiple criteria-based personalized research paper recommendation system," *Inf. Sci.*, vol. 617, pp. 41–64, 2022.

[8] P. D. V. Chaves, B. L. Pereira, and R. L. T. Santos, "Efficient online learning to rank for sequential music recommendation," in *Proc. 25th Int. Conf. World Wide Web Conf.*, 2022, pp. 2442–2450.

[9] Z. Wang, H. Chen, Z. Li, K. Lin, N. Jiang, and F. Xia, "VRConvMF: Visual recurrent convolutional matrix factorization for movie recommendation," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 6, no. 3, pp. 519–529, Jun. 2022.

[10] Q. Guo et al., "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022.

[11] J. Chen, T. Zhu, M. Gong, and Z. Wang, "A game-based evolutionary clustering with historical information aggregation for personal recommendation," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 2, pp. 552–564, Apr. 2023.

[12] J. Guo, J. Deng, X. Ran, Y. Wang, and H. Jin, "An efficient and accurate recommendation strategy using degree classification criteria for item-based collaborative filtering," *Expert Syst. Appl.*, vol. 164, 2021, Art. no. 113756.

[13] H. Liang, "DRprofiling: Deep reinforcement user profiling for recommendations in heterogenous information networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1723–1734, Apr. 2022.

[14] H. Zhang, Y. Sun, M. Zhao, T. W. S. Chow, and Q. M. J. Wu, "Bridging user interest to item content for recommender systems: An optimization model," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4268–4280, Oct. 2020.

[15] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang, "Joint item recommendation and attribute inference: An adaptive graph convolutional network approach," in *Proc. ACM SIGIR*, 2020, pp. 679–688.

[16] Y. Du, L. Wang, Z. Peng, and W. Guo, "Review-based hierarchical attention cooperative neural networks for recommendation," *Neurocomputing*, vol. 447, pp. 38–47, 2021.

[17] W.-D. Xi, L. Huang, C.-D. Wang, Y.-Y. Zheng, and J.-H. Lai, "Deep rating and review neural network for item recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6726–6736, Nov. 2022.

[18] W. Zhou et al., "Dynamic multi-objective optimization framework with interactive evolution for sequential recommendation," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 4, pp. 1228–1241, Apr. 2023.

[19] Y.-H. Chen, L. Huang, C.-D. Wang, and J.-H. Lai, "Hybrid-order gated graph neural network for session-based recommendation," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1458–1467, Mar. 2022.

[20] Z. Khan, N. Iltaf, H. Afzal, and H. Abbas, "Enriching non-negative matrix factorization with contextual embeddings for recommender systems," *Neurocomputing*, vol. 380, pp. 246–258, 2020.

[21] Q. Yang, "A novel recommendation system based on semantics and context awareness," *Computing*, vol. 100, no. 8, pp. 809–823, 2018.

[22] X. Wang et al., "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. Web Conf. 2021*, 2021, pp. 878–887.

[23] J. Wang, J. Zhou, Z. Wu, and X. Sun, "MARec: A multi-attention aware paper recommendation method," *Expert Syst. Appl.*, vol. 232, 2023, Art. no. 120847.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[25] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI*, 2019, pp. 5329–5336.

[26] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. ACM SIGKDD*, 2017, pp. 635–644.

[27] H. Liu et al., "Hybrid neural recommendation with joint deep representation learning of ratings and reviews," *Neurocomputing*, vol. 374, pp. 77–85, 2020.
[28] Y. Cai, W. Ke, E. Cui, and F. Yu, "A deep recommendation model of cross-grained sentiments of user reviews and ratings," *Inf. Process. Manag.*, vol. 59, no. 2, 2022, Art. no. 102842.
[29] L. Wu et al., "A survey on large language models for recommendation," CoRR, vol. abs/2305.19860, 2023.
[30] P. M. T. Do and T. T. S. Nguyen, "Semantic-enhanced neural collaborative filtering models in recommender systems," *Knowl. Based Syst.*, vol. 257, 2022, Art. no. 109934.
[31] I. Cantador, P. Castells, and A. Bellogín, "An enhanced semantic layer for hybrid recommender systems: Application to news recommendation," *Int. J. Semantic Web Inf. Syst.*, vol. 7, no. 1, pp. 44–78, 2011.
[32] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, and J. Yu, "Improving conversational recommender systems via knowledge graph based semantic fusion," in *Proc. SIGKDD*, 2020, pp. 1006–1014.
[33] Z. Sun, Y. Liu, Z. Cheng, C. Yang, and P. Che, "Req2lib: A semantic neural model for software library recommendation," in *Proc. IEEE 27th Int. Conf. Softw. Anal. Evol. Reengineering*, 2020, pp. 542–546.
[34] C. Yang, X. Chen, L. Liu, and P. Sweetser, "Leveraging semantic features for recommendation: Sentence-level emotion analysis," *Inf. Process. Manag.*, vol. 58, no. 3, 2021, Art. no. 102543.
[35] D. Liu, J. Li, B. Du, J. Chang, and R. Gao, "Daml: Dual attention mutual learning between ratings and reviews for item recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 344–352.
[36] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov Data Mining KDD*, 2019, pp. 950–958.
[37] H. Wang et al., "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 417–426.
[38] S.-T. Zhong, L. Huang, C.-D. Wang, J.-H. Lai, and P. S. Yu, "An autoencoder framework with attention mechanism for cross-domain recommendation," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5229–5241, Jun, 2022.
[39] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 335–344.
[40] P. Sun, L. Wu, and M. Wang, "Attentive recurrent social recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 185–194.
[41] J. Ni, Z. Huang, C. Yu, D. Lv, and C. Wang, "Comparative convolutional dynamic multi-attention recommendation model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3510–3521, Aug. 2022.
[42] Z. Wang, H. Xia, S. Chen, and G. Chun, "Joint representation learning with ratings and reviews for recommendation," *Neurocomputing*, vol. 425, pp. 181–190, 2021.
[43] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.
[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[45] A. M. Carrington et al., "Deep ROC analysis and AUC as balanced average accuracy to improve model selection, understanding and interpretation," *CoRR*, vol. abs/2103.11357, 2021.
[46] G. Santafé, I. Inza, and J. A. Lozano, "Dealing with the evaluation of supervised classification algorithms," *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 467–508, 2015.
[47] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 3307–3313.
[48] X. He, K. Deng, X. Wang, Y. Li, Y.-D. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[49] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 726–735.
[50] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web Conf.*, 2017, pp. 173–182.
[51] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," in *Proc. Int. Conf. Learn. Representations*, 2023.

Pei-Yuan Lai received the B.S. and master's degrees in electronic information engineering from Tongji University, Shanghai, China, in 2007 and 2010, respectively. He is currently working toward the doctoral degree in Big Data with the Guangdong University of Technology, Guangzhou, China. Since 2019, he has been the CTO with South China Technology Commercialization Center and has led the development and operation of more than 20 platforms. He has authored or coauthored more than 20 papers in international journals and conferences such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON BIG DATA, AAAI, ICDM, KAIS, DASFFA, and IWCMC. His research interest include data mining, in particular, recommender systems.

Qing-Yun Dai received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2001. In 2004 she joined as a Professor with the Guangdong University of Technology, Guangzhou. She is also the Leader of the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data. She has authored or coauthored more than 100 papers in international journals and conferences. Her research interests include large language model, data mining, and the application of intellectual property.

De-Zhang Liao received the bachelor's degree in computer science and technology from the Nanfang College of Sun Yet-sen University, Guangzhou, China, in 2017. He is currently a Senior Algorithm Engineer with the South China Technology Transfer Center, Beijing, China. He has authored or coauthored four papers in IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON BIG DATA, IEEE ICDM, in 2022, and *Knowledge And Information Systems*. His research interests include natural language processing, especially recommendation systems, and large language models.

Zhe-Rui Yang received the bachelor's degree in information and computing science in 2022 from Sun Yat-sen University, Guangzhou, China, where he is currently working toward the master's degree in computer technology. He has authored or coauthored five papers in IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, AAAI, IEEE ICDM and *Knowledge And Information Systems*. His research interests include data mining, especially graph mining, and recommendation systems.

Xiao-Dong Liao received the bachelor's degree from Central South University, Changsha, China, and the master's degree from South China Agricultural University, Guangzhou, China. He is currently working toward the doctoral degree with the Southern Research Institute, Jilin University, Zhuhai, China. His research interests include technology commercialization, especially the mechanisms for researchers matching the technical demands.

Chang-Dong Wang (Senior Member, IEEE) received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2013. From January 2012 to November 2012, he was a Visiting Student with the University of Illinois, Chicago, IL, USA. In 2013, he joined Sun Yat-sen University, where he is currently an Associate Professor with the School of Computer Science and Engineering. His research interests include machine learning and data mining. He has authored or coauthored more than 100 scientific papers in international journals and conferences such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACM TKDD, ACM TIST, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, KDD, AAAI, IJCAI, CVPR, ICDM, CIKM and SDM. He was the recipient of the Honorable Mention for Best Research Paper awards for ICDM 2010, the 2012 Microsoft Research Fellowship Nomination Award, and 2015 Chinese Association for Artificial Intelligence (CAAI) Outstanding Dissertation. He is also an Associate Editor for *Journal of Artificial Intelligence Research* (JAIR) and *Neural Networks* (NN).

Min Chen (Fellow, IEEE) is currently a Full Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University, Seoul, South Korea. He is also the Director of Embedded and Pervasive Computing Laboratory, Huazhong University of Science and Technology, Wuhan, China. He is the Founding Chair of IEEE Computer Society Special Technical Communities on Big Data, and was the Chair of IEEE Globecom in 2022 eHealth Symposium. His Google Scholar Citations reached more than 40,500 with an H-index of 95. His top paper was cited more than 4,100 times. From 2018 to 2022, he was selected as Highly Cited Researcher. He was the recipient of the IEEE Communications Society Fred W. Ellersick Prize in 2017, IEEE Jack Neubauer Memorial Award in 2019, and IEEE ComSoc APB Oustanding Paper Award in 2022.