Enhancing Semantic Search of Crowdsourcing IT Services using Knowledge Graph

Duankang Fu, Zhou Shufan, Beijun Shen*, Yuting Chen
School of Electronics, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
{duankangfu, sfzhou, bjshen, chenyt}@sjtu.edu.cn

***Abstract*—Mining search intents in vertical websites like IT service crowdsourcing platform relies heavily on domain knowledge. Meanwhile, it still remains a difficulty of searching services in crowdsourcing platforms, as these platforms do contain much insufficient information, for example, users tend to use images describing IT services for the purpose of advertisements. To solve these problems, we build and leverage a knowledge graph to enhance searching of crowdsourcing IT services. The key idea is to (1) build an IT service knowledge graph from StackOverflow tag synonym system, Wikipedia, StuQ and data in IT service crowdsourcing platforms, (2) plug two activities into the basic search process – term expansion and service re-ranking, (3) use superordinates, hypernyms, synonyms, descriptions and relations of entities in the knowledge graph to expand user query and service information, and (4) apply a learning-to-rank model with four features to re-rank the search results, enforcing those more relevant services have the higher-ranking position. We have conducted several experiments to evaluate our approach. The results show that our approach achieves an MRR 34.9% higher and a Recall@15 11% higher than those of a basic search approach.**

Keywords – IT Service Crowdsourcing; Knowledge Graph; Semantic Search; Learning-to-rank

I. INTRODUCTION

Recently, crowdsourcing has been widely used in many fields such as image recognition, taxonomy construction and entity resolution [1-2], etc. Those tasks are simple and straightforward, and people can deal with them with common knowledge. However, due to the strong professionalism and specialization, IT crowdsourcing is more complicated [3]. In a typical IT crowdsourcing platform, developers provide various types of IT services, and users search for target services according to their own requirements. The appropriate matching between user query and service information is one of the key values offered by IT crowdsourcing platforms.

Currently, almost all crowdsourcing platforms provide search function following the basic search process as shown in Figure 1. The services data are used to create the reverse index, and the user queries are segmented. And then Elasticsearch performs text matching between the queries and the reverse index. This approach adopts pure text matching technology, which means users have to describe their requirements precisely, or, it can't identify the latent intents of users. We also find that developers tend to use images to describe their services for the purpose of advertisements in most IT crowdsourcing platforms, and thus there are not enough available textual description for services. All these lead to low performance of semantic search in IT crowdsourcing – it is difficult for users to find their target services.

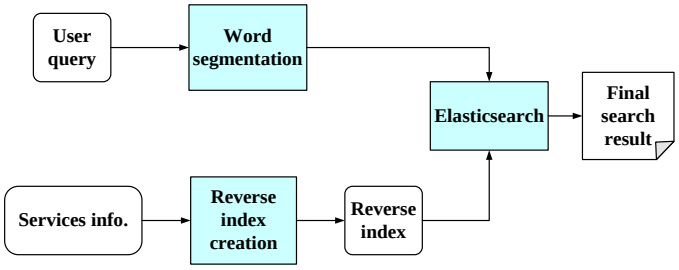

Figure 1. Basic Search Approach for Crowdsourced IT Services

As searching for crowdsourced IT services, how to understand user queries accurately? And how to complete the information of services? To address these challenges, a domain-specific knowledge graph, ITServiceKG, is constructed to enhance user query understanding. ITServiceKG mainly consists of three parts: IT service categories, IT skills and IT service instances. We insert two pluggable activities in the basic search process: term expansion and service re-ranking. After word segmentation, we use superordinates, hypernyms, synonyms, descriptions and relations of entities in ITServiceKG to expand user queries and service information. And then we get the preliminary search results from Elasticsearch, use learning-to-rank model to re-rank these results, and make the more relevant service have the higher-ranking position. We conducted several experiments to evaluate our approach. The results show that compared with the basic approach, the MRR (mean reciprocal rank) is increased by 34.871% and the Recall@15 is increased by 10.976% in our approach.

Our main contributions are summarized as follows:

1) We construct a knowledge graph of IT crowdsourcing services, which represents a complex network among IT service categories, IT skills and IT service instances.

2) We utilize ITServiceKG to expand both user queries and service information, which helps alleviate the problem that the

DOI reference number: 10.18293/SEKE2019-088
*Corresponding author.

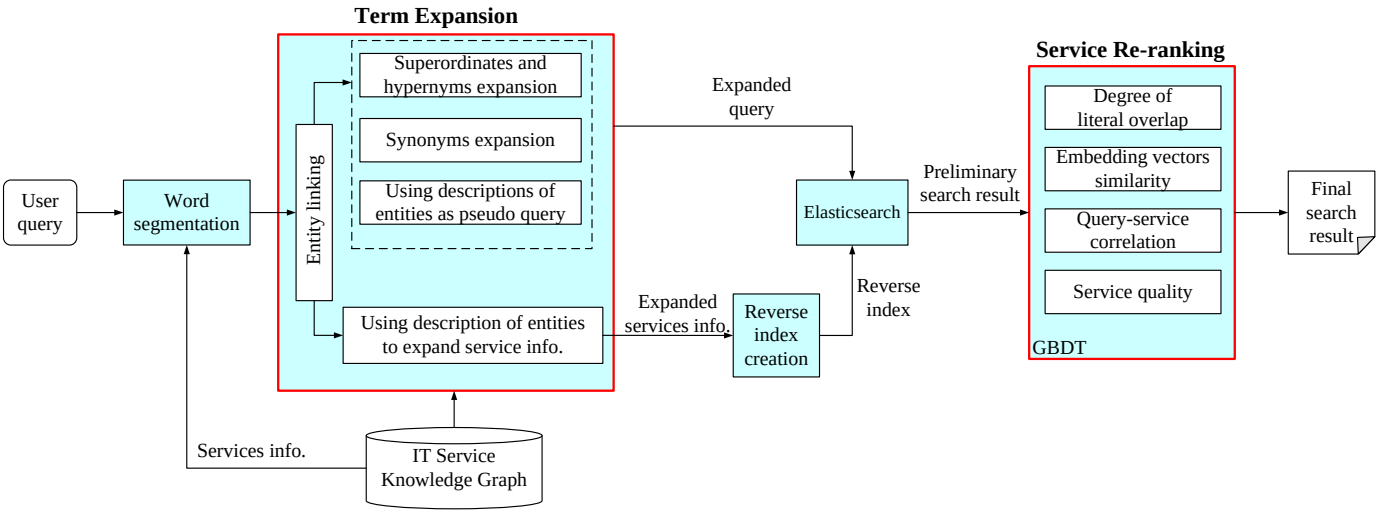

Figure 2. Overview of Our Approach

user queries are not precise and the services lack enough textual descriptions.

3) We propose a learning-to-rank model to obtain the more appropriate results. Four ranking features are designed to boost the ranking position of the more relevant services.

The rest of the paper is organized as follows. In the next section, we review some related works. The details of our approach are presented in Section III. We conduct a series of experiments to evaluate the effectiveness of our approach in Section IV. Finally, we conclude our work in Section V.

II. Related Work

A. Semantic Search

Semantic search is a broad filed, with many different aspects, ranging from query understanding, to answer retrieval, and result representation. In this paper, we focus on query understanding and answer retrieval. In previous research, people use query word expansion and retrieval model to obtain relevant results. They use synonyms as word expansion, extract word stems and obtain their different tenses [4], or use acronyms [5] and relevant words [6] to expand queries. The state-of-art retrieval model is learning-to-rank model applying machine learning methods, like LambdaRank, DSSM, CDSSM, etc. [7].

B. Knowledge Graph and its Application

There exists a wide range of general-purpose encyclopedic knowledge graphs, like WikiData, Freebase, DBPedia, and some domain-specific knowledge bases, such as WordNet, ConceptNet, etc. general-purpose knowledge graphs are not suitable for domain search, since these knowledge graphs are too general and could introduce redundant or unnecessary information, which leads to irrelevant search results. Some researchers construct an in-domain knowledge graph by extract related entities from a general domain knowledge graph [8]. However, this only extract a subset of a general knowledge graph, not optimized for in-domain specific purposes.

Knowledge graphs have been utilized in text information retrieval and made certain achievements [9]. Alexander Kotov, ChengXiang Zhai [10] used path finding and random walk to find related entities in ConceptNet as an expansion for queries. It mainly utilized the graph construction but neglected the text resources of entities. Jeffrey Dalton, Laura Dietz [11] proposed an Entity Query Feature Expansion (EQFE) model to make some improvement to the pseudo-relevance feedback model. Chenyan Xiong [12] proposed an EsdRank model, which treats extracted external words, terms and entities as objects in a latent space of queries and documents. Xiangling Zhang, et al. [13] proposed a concept called common semantic feature, to address the problem of entity set expansion by using KGs. Chenyan Xiong [14] proposed a word-entity duet representation model via combing traditional retrieval model and knowledge graph embedding, to describe the common features shared by the seed entities. Wen Zhang, et al. [15] proposed a new knowledge graph embedding method to learn distributed representations for entities and relations, which explicitly simulates crossover interactions. However, seldom researchers focus on the scenarios where documents lack enough text resources and the domain knowledge is not fully exploited.

III. Our Approach

A. Approach Overview

To address the challenges of semantic search in IT service crowdsourcing, we propose a knowledge graph-based approach as shown in Figure 2. Our approach leverages an IT service knowledge graph (ITServiceKG) to enhance query and service understanding, and then seamlessly plugs two additional activities in the basic search process to provide precise IT service search: term expansion and service re-ranking.

1) *Term expansion:* After word segmentation, we apply entity linking to locate the entities of ITServiceKG in both queries and service information. Then we expand those using superordinates, hypernyms, synonyms, descriptions and relations of these entities in ITServiceKG. Thus our

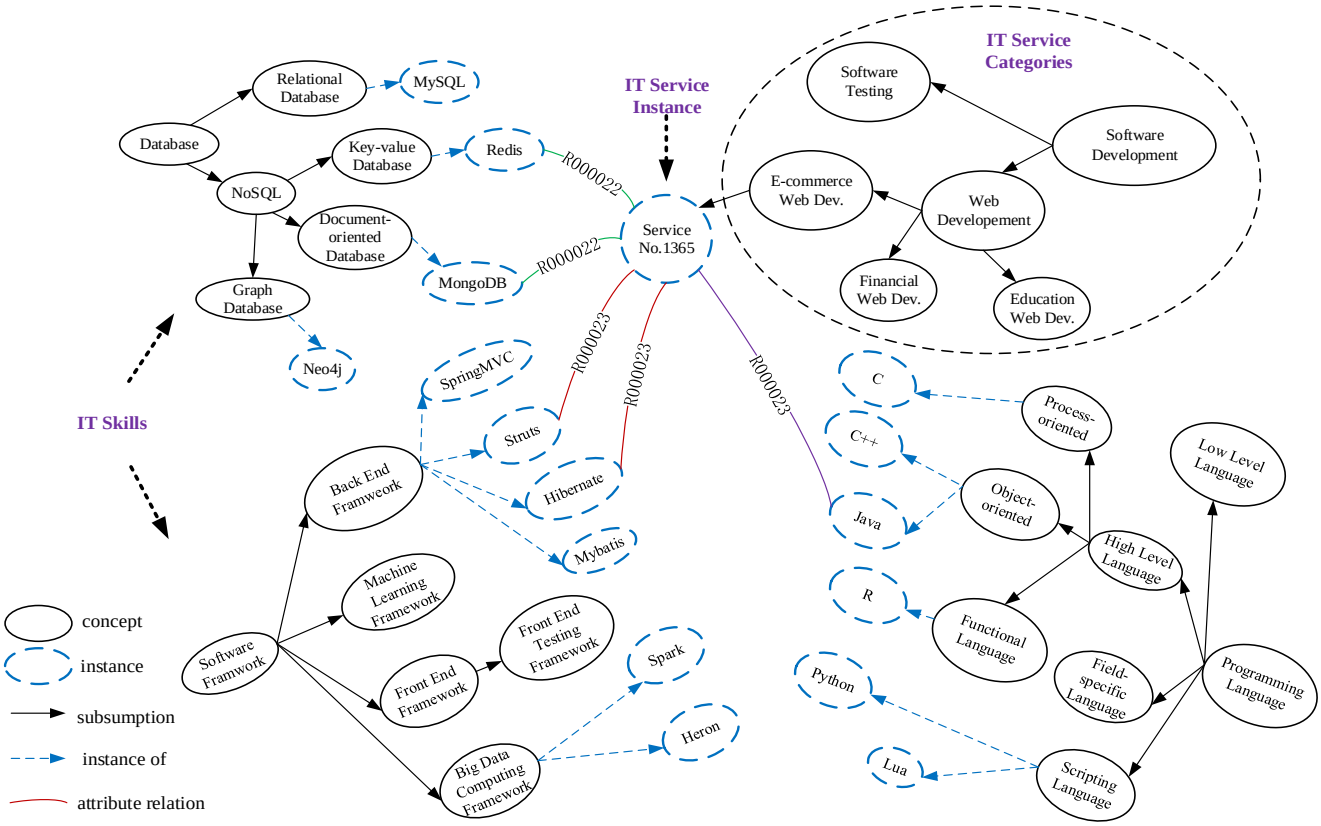

Figure 3. One fragment of IT Service Knowledge Graph

approach boosts query understanding, and alleviates the problem that service information lacks enough text resources.

2) *Service re-ranking:* After the preliminary search results are returned by the Elasticsearch engine, a learning-to-rank model is applied to re-rank the results and obtain the top-N IT services. Our approach designs several novel features, including degree of literal overlap, embedding vector similarity, query-service correlation, and service quality. Thus, it makes the more relevant service have the higher-ranking position.

Next explains the details of IT service knowledge base, term expansion and service re-ranking.

B. Building IT Service Knowledge Graph

A domain knowledge graph in IT crowdsourcing should include the information of IT services and the knowledge in IT service implementation. Therefore, we design ITServiceKG from three aspects: IT service categories, IT skills and IT service instances. In ITServiceKG, each entity (i.e. concept or instance) has following attributes: name, synonym, cooccurrence, and description; and relations between entities includes subsumption, instance-of, and attribute relations. One fragment of ITServiceKG is shown in Figure 3.

1) IT service categories: IT service categories record the hierarchical structure of service categories. These data are provided by the real-world IT crowdsourcing platform–*JointForce*[1]. It holds a three-layer structure. The first layer contains various types of software services, including those for software development, software testing, architecture design, DevOps deployment, logo design, etc. The second layer collects sub-types of specific IT services. Let "software development" be an example. Its child services include software services such as "web development", "App development", "embed system development", and so on. And the third layer places domain-oriented IT services.

2) IT skills: IT skills contain the technologies that IT services adopt, such as programming languages, frameworks and database. Every skill holds a multi-layer structure. Take "Database" as an example, it can be divided into "Relational Database" and "NoSQL Database", and MySQL is an instance of "Relational Database". IT skills data are collected mainly from StackOverflow tag synonym system[2], Wikipedia[3] and StuQ skill map[4]. And the cooccurrence data is processed from search logs.

3) IT service instances: IT service instances are to-be-crowdsourced IT services in the platform. They are connected to IT service categories and IT skills. Taking an example in Figure

[1]https://www.jointforce.com
[2]http://stackoverflow.com/tags/synonyms
[3]https://www.wikipedia.org
[4]https://github.com/TeamStuQ/skill-map

3, Service No.1365 is an instance of e-commerce website development, with Redis and MangoDB as databases, Java as programming language, and Struts and Hibernate as frameworks.

C. Term Expansion

Term expansion is a common technique in information retrieval (IR), but we reach two conclusions that implies space for improvement: a) in previous research, the query expansion resources are from synonyms and cooccurrence words. The relations of superordinates and hyponyms are neglected; b) most researchers only pay attention to the query expansion but neglect the document expansion. In IT service crowdsourcing platforms, there are not enough textual description for services, which impedes the performance of semantic search. Therefore, the expansion of both user queries and service information is indispensable, and we leverage ITServiceKG to expend them along the following four channels.

1) Synonym and cooccurrence expansion. The entities in the ITServiceKG have attributes "synonym" and "cooccurrence", and we use synonym and cooccurrence terms of the linked entity to expand the quires.

2) Superordinates and hyponyms expansion. The IT service categories in ITServiceKG have a hierarchical structure. For example, in Fig. 3, the entity "Software Framework" has a hyponym "Back End Framework", the latter has a hyponym "Hibernate", and "Hibernate" as a relation with Service No.1365. We expand both for queries and services information with superordinates and hyponyms of the linked entity in ITServiceKG. For example, if a user wants to search for IT services with back-end frameworks, but does not know any concrete name of it, we can expand this query by adding the hyponyms of back-end framework: "Struts", "Hibernate", etc. For another example, Service No.1365 has a relation with entity "Struts", we can expand the information of Service No.1365 by adding the superordinate of "Struts" - "Back End Framework".

3) Using the descriptions of entities as pseudo query. ITServiceKG contains rich textual descriptions about entities. Given an entity e, we use its name and description as pseudo query. This makes sense because the query is usually short and concise, only including two or three keywords usually. So we use the descriptions of entities as the pseudo query, without worrying that the expanded query is too long or introduces too many irrelevant information.

4) Using the description of entities to expand service information. It makes sense that the extra information of entities in ITServiceKG can help us understand the meaning of the service information. It is like a process of looking up words in the dictionary. When we read an article and find a new word, we may look up the word in the dictionary. The meaning of the new word will help us understand the article. The domain knowledge graph behaves in the similar way. The description of entities can be helpful for users who do not have the background knowledge about the entities.

D. Service Re-Ranking

After the term expansion, the expanded query and service information are obtained. We build the reverse index in the Elastic search engine, and get the preliminary results. Then, we apply machine learning techniques to re-rank the list of the search results and make the more relevant result have the higher-ranking position. This rank model can be viewed as a supervised learning problem. The input is the query and all to-be-crowdsourced IT services, the output is the top-N IT services, each of which has a relevant score of 0-1.

We adopt Gradient Decision Tree (GBDT) as our learning-to-rank model. It is a point wise ranking model. We design four features for this rank model, shown as below:

1) The degree of literal overlap. If the query and the service's name and description have more common characters, they are more relevant.

$$\mathrm{d} = \frac{S_q \cap S_t}{\left|S_q \cup S_t\right|} ,$$

where S_q denotes the character set of a query and S_t denotes the character set of a service name.

2) Embedding vector similarity. We embed the query and the service's name and description into vectors by TD-IDF. The cosine similarity of the vectors between the query and the service indicates the relevance.

$$sim = \cos(v_q, v_t) ,$$

where v_q denotes the query vector and v_t denotes the vector of the service.

3) Query-service correlation. In IT service crowdsourcing platform, we record user service clicking events on search results as <query, service, timestamp> triples in the log. More clicking times indicates higher relevance.

4) Service quality. User quality evaluation on crowdsourced services are also recorded in IT service crowdsourcing platform. People tend to search for services with good quality, so it makes sense that we give them higher relevance score.

$$quality = w_p \cdot p + w_s \cdot s + w_t \cdot t ,$$

where *t* denotes the score of service on-time delivery, and *s* denotes the product quality score, *t* denotes the technical support score, and *w* is the corresponding weight.

We transfer the query-service pair to feature vectors and use them as the input of the GBDT model. Then a point wise ranking model is learned through model training on historical data, and re-ranks the current search results.

IV. Experiments

We implemented our semantic search approach, and conducted evaluations to explore the following research questions:

(RQ1): What is the effectiveness of our approach compared with the baseline?

(RQ2): How much does each expansion channel contribute to IT service search?

TABLE I. THE OVERALL PERFORMANCE COMPARISON

Methods	P@10(%)	Δ	R@10(%)	Δ	P@15(%)	Δ	R@15(%)	Δ	MRR	Δ
TF-IDF	13.096	—	49.563	—	9.856	—	56.396	—	0.542	—
BM25	13.107	+0.084	51.872	+4.659	9.905	+0.497	57.197	+1.420	0.553	+2.030
term expansion	13.053	-0.328	52.369	+5.661	9.996	+1.420	61.263	+8.630	0.695	+28.229
term expansion + service re-ranking	**13.298**	**+1.542**	**52.965**	**+6.864**	**10.236**	**+3.856**	**62.586**	**+10.976**	**0.731**	**+34.871**

(RQ3): How much does each feature contribute to IT service re-ranking?

A. Experiment Setup

1) Dataset. Experimental data is offered by JointForce, the biggest IT crowdsourcing platform in China. There are 8753 services and 18025 search records. Each record in search logs contains a query (q), top k service list returned by search engine ($Sq@k$), and service list clicked by user (Iq). We split the dataset into training (80%) and testing (20%).

2) Evaluation Metrics. We use precision (P@k), recall (R@k) and MRR (Mean Reciprocal Rank) to measure the performance of our approach. For query sets Q, these metrics are defined as follows:

Precision (P@k): It is the percentage of correctly discovered services in all discovered k services.

$$\text{P@k} = \frac{1}{|Q|}\sum_{q\in Q}\frac{|Sq@k \cap Iq|}{|Sq@k|}$$

Recall (R@k): It is the percentage of correctly discovered services in all correct services.

$$\text{R@k} = \frac{1}{|Q|}\sum_{q\in Q}\frac{|Sq@k \cap Iq|}{|Iq|}$$

MRR: The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer: 1 for first place, 1⁄2 for second place, 1⁄3 for third place and so on. The MRR is the average of the reciprocal ranks of results for a sample of the query sets Q:

$$MRR = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{rank_q},$$

where $rank_q$ refers to the rank position of the first relevant service for the query q.

B. RQ1. Overall Performance

Basic search approaches with BM25 and TD-IDF are selected for overall performance comparison. The experimental result is shown in the Table I. We can observe that the performance of BM25 is better than TD-IDF, therefore, we choose BM25 as the reference in the subsequent experiments. Moreover, we can observe that when applying term expansion, the recall is enhanced more prominently than the precision. It makes sense because the term expansion introduces more useful relevant terms and can find more correct services which cannot be retrieved before. After further applying ranking model, the precision is enhanced, because we make the more relevant services have higher ranking position. The performance of @15 is better than @10, which means @15 is a balanced metric for both precision and recall.

The experiment demonstrates that our approach (i.e. basic search with BM25 + term expansion + service re-ranking) outperforms the baselines. The precision is improved by 3.856%, he recall is improved by 9.912%, and the MRR is improved by 34.871%. This is because we introduce extra domain knowledge and retrieve some results that are neglected in the baseline approaches.

C. RQ2. Expansion Channel Contribution Analysis

During term expansion, our approach adopts four expansion channels: synonym and cooccurrence expansion (SCQ), superordinates and hyponyms expansion (SHQ), descriptions of entities as pseudo query (DQ), and descriptions of entities to expand service information (DS). In this experiment, we perform an analysis to evaluate each channel's contribution to the performance of IT service search.

We choose basic search approach with BM25 as the baseline, and add each expansion channel one by one. Recall@15 is chosen as the evaluation metric, because term expansion mainly enhances the recall.

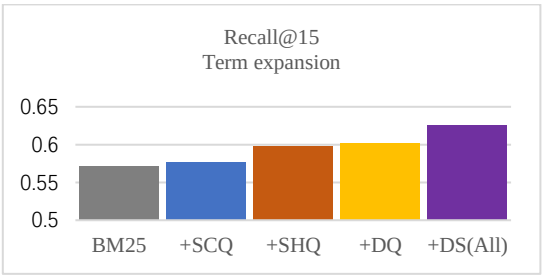

Figure 4. Expansion Channel Contribution Analysis

The impact of each expansion channel on the overall performance in this experiment is shown in Figure 4. We can observe that the performance is improved greatly when introducing "Superordinates and hyponyms expansion" and "Using description of entities to expand service information". We take following two examples to illustrate why superordinates and hyponyms can improve the performance.

The first query example is "ERP web service that uses Structs and Hibernate". Before term expansion, we find some related services, all of them relying on exact match. But Struts is not as popular as Spring now, so the services using Struts are rare. When we introduce the superordinates of entity "Struts"-

"Back End Framework" and add it to the query, we can find some services with back-end framework, such as using SpringMVC and Mybatis.

The second query example is "student management system that uses NoSQL". Before applying term expansion, we cannot find services with keyword "NoSQL", because "NoSQL" does not appear in the name or description of any service, since more detailed words are used to describe their techniques, such as "Redis", "MongoDB" rather than the general word "NoSQL". Therefore, we introduce the hyponyms of the entity "NoSQL", and then we can find services use Neo4j and MongoDB, which are both NoSQL databases.

We can also observe that when we use the description of entities to expand service information, the performance is improved greatly. The reason is that there are not enough available textual descriptions for services, and after we use description of entities to expand them, this problem is alleviated, and thus the performance becomes better.

D. RQ3. Ranking Feature Contribution Analysis

Our approach adopts four features for the learning-to-ranking model: degree of literal overlap (DLV), embedding vectors similarity (EVS), query-service correlation (QSC) and service quality (SQ). In this experiment, we perform an analysis to evaluate each feature's contribution to the performance of ranking model.

The baseline is the basic approach with term expansion. The metric is Recall@15. We add the features one by one and the impact of each feature on the performance is shown in Figure 5.

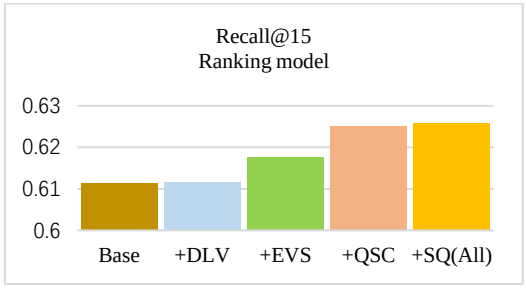

Figure 5. Ranking Feature Contribution Analysis

From Figure 5 we can see that all the features take effect. The embedding vectors similarity and the query-service correlation contribute most in the four features. The embedding vectors similarity mainly represents the similarity between the queries and the name of services. This implies the name of service plays an important role in the model. The query-service correlation implies how correlated a query and a service is in the history of previous crowdsourcing, so it can give advice to the search in the present.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a knowledge graph approach to enhancing semantic search for crowdsourced IT services. Compared with the traditional search approach, the MRR of our approach increases 34.871% and the Recall@15 increases 10.976%. The key to success comes from two aspects: (1) an IT service knowledge graph is built and utilized to expand both user queries and service information; (2) a learning-to-rank model with four features is designed to re-rank the preliminary search results.

As for future work, we will employ neural networks and learn latent representations of words, entities, and their relations in the knowledge base. These latent representations can be learnt in an unsupervised manner to be subsequently leveraged in a ranking model.

ACKNOWLEDGMENT

This research is supported by 973 Program in China (Grant No. 2015CB352203), National Nature Science Foundation of China (Grant No. 61472242 and 61572312), and Shanghai Municipal Commission of Economy and Informatization (No. 201701052). Thanks JointForce for providing the experimental data set.

REFERENCES

[1] Y. Sun, A. Singla, D. Fox, and A. Krause. Building hierarchies of concepts via crowdsourcing. *arXiv preprint arXiv:1504.07302* (2015).

[2] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5, 11 (July 2012), 1483-1494.

[3] Runtao Qiao, Shuhan Yan and Beijun Shen, A Reinforcement Learning Solution to Cold-Start Problem in Software Crowdsourcing Recommendations. In *International Conference on Progress in Informatics and Computing* (*PIC*) 2018.

[4] Bhogal J, Macfarlane A, Smith P. A review of ontology based query expansion. *Information Processing & Management* 43, 4 (July 2007), 866-886.

[5] Wei Xing, Peng F, Dumoulin B. Analyzing web text association to disambiguate abbreviation in queries. In *SIGIR* 2008. ACM, 751-752.

[6] Derczynski L, Wang Jun, Gaizauskas R, et al. A Data Driven Approach to Query Expansion in Question Answering. *arXiv preprint arXiv:1203.5084* (2012).

[7] B Mitra, N Craswell, Neural models for information retrieval. *arxiv preprint arXiv:1705.01509* (2017).

[8] Chenyan Xiong. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In *WWW* 2017. ACM, 1271-1279.

[9] Laura Dietz, Chenyan Xiong, Edgar Meij. The First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR). In *SIGIR* 2017. ACM, 1427-1428.

[10] Alexander Kotov, ChengXiang Zhai. Tapping into knowledge base for concept feedback leveraging conceptnet to improve search results for difficult queries. In *WSDM* 2012. ACM, 403-412.

[11] Jeffrey Dalton, Laura Dietz, James Allan, Entity query feature expansion using knowledge base links. In *SIGIR* 2014. ACM, 365-374.

[12] Chenyan Xiong, Jamie Callan. Esdrank: Connecting query and documents through external semi-structured data. In *CIKM* 2015. ACM, 951-960.

[13] Xiangling Zhang, Yueguo Chen, Jun Chen , et al. Entity Set Expansion via Knowledge Graphs. In *SIGIR* 2017. ACM, 1101-1104.

[14] Chenyan Xiong, Jamie Callan, Tie-Yan Liu. Word-Entity Duet Representations for Document Ranking. In *SIGIR* 2017. ACM, 763-772.

[15] Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, Huajun Chen, Interaction Embeddings for Prediction and Explanation in Knowledge Graphs. In *WSDM* 2019. ACM, 96-104.