A QA search algorithm based on the fusion integration of text similarity and graph computation*

Zhaoyu Sun, Lei Song, and Jiaming Yu

Beijing Rui Li Technology Co., Ltd., Beijing, China Yujm@powerkeen.com
http://www.powerkeen.com

Abstract. The open domain Q&A QA system based on knowledge graph is essentially an entity alignment and search problem. It needs to find and extract the information contained and lacked in the question in the comparison with graph data unit. Our team proposed two search models: text similarity and graph computation. Combined them with some basic strategies, the result sets of the two are integrated, and a good effect is obtained.

Keywords: text similarity · graph computation · 2-degree problem · entity discovery · model integration.

1 Overview

The open domain Q&A QA system based on knowledge graph is essentially an entity alignment and search problem. It needs to find and extract the information contained and lacked in the question in the comparison with graph data unit. The meta-knowledge form of the "subject-predicate-object" triple of knowledge graph has a corresponding relationship with the intrinsic expression logic of natural language. The problem of simple sentence can basically be one-to-one correspondence; but for the semantically similar but literally irrelevant form or contains nodes with large information density in the graph, the alignment of the question and the graph data becomes very complicated, and the noise information in the question and the redundant and irregular data in the graph make the process become more difficult and error-prone.

Our team proposed two search models: text similarity and graph computation. The former is good at solving the 1-degree problem, and the latter is good at solving the multi-degree problem which has larger number of information nodes. Combined with some basic strategies, the result sets of the two are fused, and a certain effect is obtained.

2 The system framework

This system integrates the processing results of both text similarity and graph calculation models, and improves the performance of single model, including four major processing steps: question analysis, entity discovery, solution search, and fusion integration, as shown in Fig. 1 below. The work in this period also includes certain data pre-processing work, mention data index, triple data reduction, entity type data reduction, and the construction of some spoken templates, similar phrases, and stop words.

The question analysis mainly realizes the spoken language recognition of the question, converts it into a written expression, and calls the dependency syntax analysis to analyze the problem type and the problem object.

Entity discovery searches and finds the core main entity in the question through Mention index, subject entity index, and object entity index, so as to perform subsequent similar search or graph traversal. In this process, some attribute entities or type entities that modify the main entity are usually located on the left side of the main entity, causing the system to misjudge the main entity and encounter a long object entity problem. We identify these problems by certain attributes, type entity identification rules, and rules for object entity priority matching.

In the solution search process, we propose two strategies of 1-2degree entity search based on text similarity and graph calculation search. The entity search based on text similarity mainly solves the semantic part matching and cursor mechanism. The 2-degree associated entity expansion is used to perform similar matching to reduce the noise impact. Based on the graph calculation search strategy, methods such as pruning, node merging and feature matching, etc. are used. For the determination of some of these parameters, perceptron learning techniques are employed.

The fusion integration integrates and combines the two different results in the solution search process, adopts the fusion strategy based on problem type feature, the fusion strategy based on graph traversal path and voting mechanism.

* KeenPower. Beijing Rui Li Technology Co., Ltd.

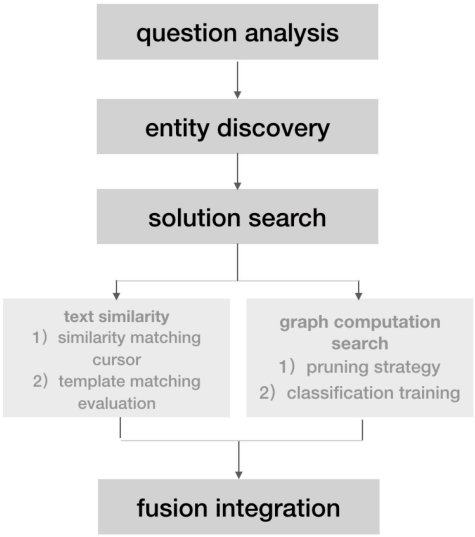

Fig. 1. Processing framework based on the fusion integration of text similarity and graph computation

3 Text similarity search model

From the perspective of the knowledge structure of existing graph, sentence is analyzed by information segmentation. The sentence is a path composed of different knowledge nodes, and different nodes correspond to different phrases in the sentence. Text-based search is to find possible paths in the graph, and then take the search path with the highest similarity.

The evaluation of similarity needs to consider the following aspects:

1) The similar values between graph nodes and phrases in sentence;

2) The similarity matching value of logical order between graph node and node;

3) The question type and the position of the question keyword in sentence correspond to the position of adjacent node in graph.

3.1 Entity discovery

Searching for the way that all triples are similarly aligned entities will be affected by larger noise information, resulting in the loss of search entry, and the annotation information of object entity will also affect the ordering of search hit entities. For example: the most similar triple of "德国著名的汽车品牌" is as follows,

{
"*id*" : 37730758,
"*subject*" : " <著名汽车品牌图鉴> ",
"*predict*" : " <软件名称> ",
"*object*" : "著名汽车品牌图鉴"
}, causes its entry entity (subject entity - 德国) to be lost.

Therefore, entity discovery needs to rely on mention data, subject entity index data, and object entity index data. All possible subject entities and object entities in the question are the starting point of the question. The rest information of the question belongs to predicate information. Generally, the matching requirements of subject and object entity should be higher than those of predicate, even predicate node information is not included in the question.

1) Use object entity index to find whole sentence entity

Long sentence may appear as object entity as a whole.

For example: 首位登上过时代周刊封面的华人女歌手? 谁首次实现了中国的首次统一?

2) Discover entity using mention index

3) Identify the predicate entity or type entity that may appear before main entity

For example: 书籍《亚利安01》的价格是多少

"书籍"、"亚利安01"were found to be entities, but the former and the latter have category relation.

1-degree problem solving The search implements the stepwise matching of the information contained in the question by a similar matching cursor mechanism. Each time an information node is matched, the matching cursor will narrow the description information range of the question.

The 1-degree problem solving uses main entity as subject or object. The remaining information after excluding main entity in the question is used as predicate information. The search is performed through triple index, and the search result is verified, return to the triple, if the relationship between predicate and the remaining information in the question is inclusion relationship or the similarity of the largest similar match string is above a certain threshold, the triple is considered to be the answer corresponding to the question description. Then the matching cursor will move to the right. If the remaining information in the question still contains potential predicate entity or object entity after excluding stop words, it is regarded as a 2-degree problem and is executed by other modules; otherwise, it is regarded as 1-degree question and the search results are returned directly.

2-degree problem solving The matching of triple to problem in 2-degree problem requires at least two triples to be satisfied. Since the alignment sequence between the structure of graph and the phrases in the question is not exactly matched from left to right, in order to avoid the complicated processing of phrase combination sequence, we will spread the nodes related to main entity at 2 degrees. Then, calculate the similarity between each piece and the question, and select the best matching triple as candidate answer. Combined with the structure of triples, the description of 2-degree problem can be summarized into the following six types:

1) S-P-<>-P-?

The main entity starts as a subject and queries its indirect attribute values.

For example: 《蒙娜丽莎》的作者皈依的是什么宗教? 赵明诚的配偶有哪些主要作品?

2) S-<>-O-P-?

Similar to 1), but the subject entity is not directly related to the attribute, but its object entity.

for example: 澳大利亚的悉尼有什么著名景点?

3) S-P-?-P-<>

Similar to 1), but its problem is another subject of the object entity of main entity.

for example: 周恩来的妻子曾当过什么主席?

4) O-P-S-P-?

The main entity starts as an object entity and queries its indirect attribute values.

For example: 小说《哈利波特》的作者是谁? 郦道元的《水经注》编撰于哪个朝代?

5) O-<>-?-P-O1

The main entity starts as an object entity, and with the aid of other entities in the question, queries its subject entity.

For example: 北京大学出了哪些科学家? 理查德·格里格和菲利普·津巴多合著的书是什么?

6) O-P-?-<>-O1

Similar to 5), but predicate information is the parent attribute of main entity.

for example: 有哪些位于湖北的公司?

In general, 2-degree problem is divided into three steps: 1) spread 2-degree solution space; 2) pre-screening result set of question similar match; 3) based on the pre-screening result set from 2), similar match is graded according to 6 types of template, and the triple with the highest degree of matching is considered a candidate answer.

4 Graph computation model

4.1 Structure chart of model

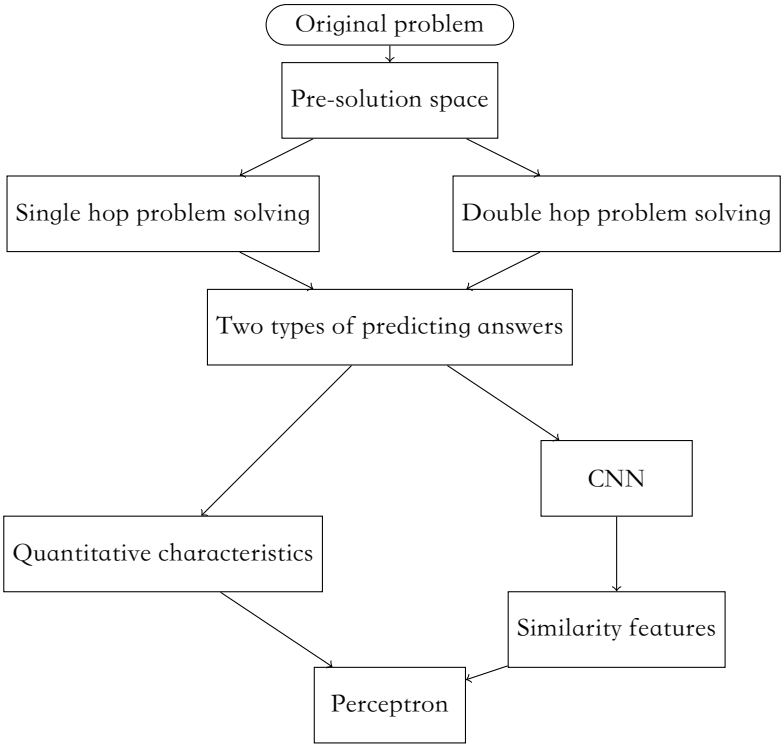

Fig. 2 Solution search model of graph calculation

4.2 Pre-solution space

The entity in the question can basically use pkuorder as a dictionary for entity connection. The generation method of pre-solution space is: for possible entities, go to the database to search its child nodes or parent nodes as the first hop node, and then search for the child node of the first hop node. It acts as the second hop node. The corresponding graphs of different entity entries are stored separately.

For example, the question ″董卿主持的正在播出的节目是?″,when we take ″董卿″ as entity entry, we can get the following pre-solution graph.

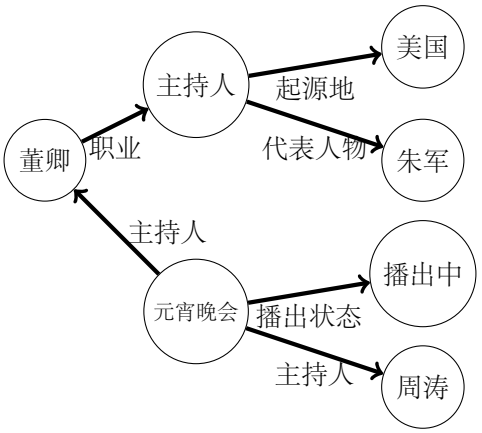

Fig. 3 An example of pre-solution space generation

This example explains why two directions are used in the first hop. Because ″元宵晚会的主持人是董卿″ and ″董卿主持了元宵晚会″, the same fact is expressed, but we can't determine which mode is used for database to store this knowledge. The pre-solution tree is usually more than one. For this problem, the ″节目″ can also generate a pre-solution tree.

4.3 The similarity features of quantitative characteristics

For example, "Double hop problem":
"张柏芝、谢霆锋合作的动作电影里的服装是谁设计的".
Let us remember a knowledge triplet as $S-P-O$

Structural features First, we will get two pre-solution trees with Cecilia Cheung and Nicholas Tse as entries. For the subtree with the form of $S_1-P_1-O-P_2-S_2$on each pre-solution tree, it is a subtree of, where S_1="<张柏芝>" or "<谢霆锋>". We use the following characteristics to calculate seq, the jaccard distance of $seq, S_1+P_1+P_2$, the number of common characters of$seq, S_1+P_1+P_2+O+S_2$), hint, seq, the edit distance of $S_1+P_1+P_2$, Les, seq and the most important thing is that we introduce a definition called placeholder. That is, these cue words such as "是谁,是什么,哪里,何时,哪个,有什么"in the question. It is found that there is often important information of answer reminder near the cue word. For this problem, "design" is a very important message. If the P_2 of the subtree $S_1-P_1-O-P_2-S_2$ contains design, then its score should be very high. Specifically, we find a substring S_{attach} in the sentence for P_2, so that the string can coincide with the head or tail of P_2, and then take the distance between the string S_{attach} and the placeholder as a feature.

Node consolidation We got an array with elements $(path: S_1-P_1-O-P_2-S_2, jaccard, hint, les, placeholder)$ (using S_2 as a possible answer node). But it is noticed that what the question asked was the "张柏芝,谢霆锋共同主演的电影". As the bridge node O, "无极" will actually connect with "张柏芝" and "谢霆锋" as S_1. In the case of bridge node O is the same as answer S_2, we will merge the two candidate answers and recalculate relevant features.

Pruning Since the previous entity link used an exhaustive matching method, the size of the candidate answers is very large at this step. If the classifier is trained on this, the training will fail due to the category imbalance. Inspired by the decision-making tree model, we use heuristic rules for pruning. Using the $mean$ value of one or several indicators as a benchmark, we search for magnification a, so that the number of pre-selected answers which are greater than $amean$ is less than 80. Thus, we control the positive and negative ratios of training.

Similarity features Using pure quantitative characteristics makes us lose a lot of information, so it is necessary to input new features for the final perceptron. Here CNN is used to measure the similarity between the path of candidate answers $S1P1OP2, S1P1$ and questions seq. Using single hop, double hop to solve each problem will get answer and relevant path, training is performed by using path, seq as X, the $F1$ of answer as Y.

5 Fusion integration

5.1 Fusion strategy based on problem type features

Answer type is determined according to the cue words of the question ("是谁, 什么时间, 哪个国家" etc.), and the one accord with the type is prioritized to be selected. For example, in the case of a character, "董卿" has an attribute of "类型-> 人物", thus, we can judge that this entity is a character. In fact, attribute needs to be reconstructed, and the type in $pkutype$ has error information. We can reconstruct the type in an iterative way and clean the noise. Specifically, starting from the attribute of "人物", we can get a bunch of entities, and then this bunch of entities will generate a bunch of predicates P. We sort the predicates and use some features like tf-idf to screen out common features (such as "中文名"), so we build the mapping of the "人物" type to the predicate. We can in turn reconstruct the $pkutype$ through the predicate set so as to eliminate noise.

5.2 Fusion strategy based on graph traversal path

When the path that generates the answer accords well with the problem (information is neither too much or too little), then the problem is a priority.

5.3 Voting

If there are two models that choose the same answer, then the final answer will be this answer.

Table 1. The evaluation data of three test set Models

Test set	*Precision*	*Recall*	*F*1
Similarity model	0.610731404642	0.640806847647	0.614250648591498
Perceptron model	0.5372	0.5690	0.5346
Fusion integration	0.664356404642	0.694806847647	0.666090475431325

6 Experimental data

The similarity model and the graph computation perceptron model were evaluated on the validation set and the test set, respectively, and the fusion integration test was performed on the test set. The test data is shown in the table below.

The test data shows that the similarity and graph calculation perceptron models have their own strong characteristics, and the two have complementary characteristics. After fusion integration, scores are improved.

7 Summary and outlook

This paper proposes a comparison analysis of implementation problems and graph knowledge between two search models: text similarity model and graph calculation model. It is observed in the experiment that text similarity model has certain advantages in the aspect of full enquiry investigation. Full-text indexing works well for entity discovery and alignment in large-scale triple data. In the solving process, full-text indexing combined with synonyms can get the effective primary selection of answer sets and narrow the screen space of the result sets. The graph calculation perceptron model effectively utilizes the association features between nodes, when combined with the entity and problem type information, it can effectively determine the scope of answer entity. In the process of solving complex problems, both models use the link mode between entities in the graph structure to find the best matching path, and extract answer entities according to problem type and object. The results of the two models are consolidated and integrated according to entity type and path, which can improve the results.

Meanwhile, it is found that some problems can be improved in the optimization process of models: 1) the performance of similar dynamic solution space generation and pre-screening calculation processing is low; 2) a more optimized learning style can be adopted in the weight parameter optimization of similarity values of the entities with different roles in the link mode between entities. In addition, how the relative relationship between entity and predicate in the syntactic parse tree can be further explored as a feature of assisted optimization.

References

1. Yih W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics 2015.
2. Zhu C., Ren K., Liu X., Wang H., Tian Y., Yu Y. A Graph Traversal Based Approach to Answer Non-Aggregation Questions over DBpedia. In: Qi G., Kozaki K., Pan J., Yu S. (eds) Semantic Technology. JIST 2015.
3. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: EMNLP, pp. 1533 – 1544 (2013)
4. Unger, C., Bühmann, L., Lehmann, J., Ngomo, A.-C.N., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 639 – 648. ACM (2012)