Domain Adaptive Question Answering over Knowledge Base

Yulai Yang[1,2,3], Lei Hou[*1,2,3], Hailong Jin[1,2,3], Peng Zhang[1,2,3], Juanzi Li[1,2,3]
Yi Huang[4], and Min Hu[4]

[1] DCST, Tsinghua University, Beijing 100084, China
[2] KIRC, Institute for Artificial Intelligence, Tsinghua University
[3] Beijing National Research Center for Information Science and Technology
[4] China Mobile Research Institute
yyl277078178@163.com; greener2009@gmail.com (*corresponding author);
tsinghua_phd@163.com; zpjumper@gmail.com; lijuanzi@tsinghua.edu.cn;
huangyi@chinamobile.com;humin@chinamobile.com

Abstract. Domain-specific question answering over knowledge base generates an answer for a natural language question based on a domain-specific knowledge base. But it often faces a lack of domain training resources such as question answer pairs or even questions. To address this issue, we propose a domain adaptive method to construct a domain-specific question answering system using easily accessible open domain questions. Specifically, generalization features are proposed to represent questions, which can categorize questions according to their syntactic forms. The features are adaptive from open domain into domain by terminology transfer. And a fuzzy matching method based on character vector are used to do knowledge base retrieving. Extensive experiments on real datasets demonstrate the effectiveness of the proposed method.

Keywords: Natural language question answering· Knowledge base · Domain adaptation.

1 Introduction

Domain-specific question answering over knowledge base (KBQA) is an important way of using domain knowledge, which takes natural language questions as input and returns professional and accurate answers. Many previous approaches rely on hand-crafted patterns or rules, which are time-consuming and inevitably incomplete. To avoid the complex hand-crafted patterns construction, some methods try to generate patterns automatically. Abujabal et al. [1] propose a distant supervision method to learn question patterns and align them with the knowledge base by question and answer (QA) pairs. Besides, end-to-end methods based on neural networks are widely used in KBQA task. They represent both questions and answers as semantic vectors by neural networks, and calculate the similarities between vectors to select the final answers. These methods need the QA pairs for supervised training, which are not easy to obtain in a

specific domain compared with open domain. Therefore, how to utilize existing open domain resources for domain adaptation poses a critical challenge.

To address the above issues, we propose a method which can construct a domain adaptive KBQA system with open domain questions. We define a new pattern to represent questions with the Parsing and POS results which can be easily obtained through existing NLP tools, and then train a key entity and property phrases detection model with open domain questions based on the learned representations. In order to adapt the key phrase detection model to specific domains, we utilize a terminology transfer method to make the distributions of domain questions and general questions consistent in feature space as far as possible. Finally, we retrieve the knowledge base via a character vector-based fuzzy matching to get the final answer. The key phrase detection model is evaluated on a open domain and two domain-specific datasets, one is insurance and the other is China Mobile products. For each specific domain, we construct a KBQA dataset to evaluate the performance of the entire model.

In summary, the main contributions of this paper can be described as follows:

- We propose a general method to construct a domain adaptive KBQA system using open domain questions without any hand-crafted template and QA pair.
- We define a new and simple question representation pattern which is effective for key phrase detection and domain adaptation.
- Experiments show that our domain adaptive method can achieve good performance in both real-world insurance and China Mobile products domain datasets.

The rest of this paper is organized as follows. Section 2 formulates the problem. Section 3 presents the framework and method details. Section 4 describes the method evaluations, Section 5 discusses the related work, and finally Section 6 concludes this work with some future research directions.

2 Problem Formulation

Definition 1 (Knowledge Base). *A knowledge base/graph $\mathcal{K}$ is a collection of subject-predicate-object triples $(\mathbf{s}, \mathbf{p}, \mathbf{o})$, which also form a graph (hence the name). $\mathbf{s}$ is an entity $e \in \mathcal{E}$ or a concept/class $c \in \mathcal{C}$, $\mathbf{p} \in \mathcal{P}$ is a property and $\mathbf{o}$ could also be literals $l \in \mathcal{L}$ besides entities and concepts.*

Domain knowledge base is often a collection of the domain facts, and Fig. 1 presents a fragment of the insurance knowledge base. Nodes are the subjects such as "com_abrs" or the objects such as "安邦人寿(AnBang)". Directed edges are the predicates which describe the relations or properties between subjects and objects such as "company".

Definition 2 (Question). *A question*[5] *is a linguistic expression used to make a request for information. We only consider the factual question in this paper. A*

[5] https://en.wikipedia.org/wiki/Question

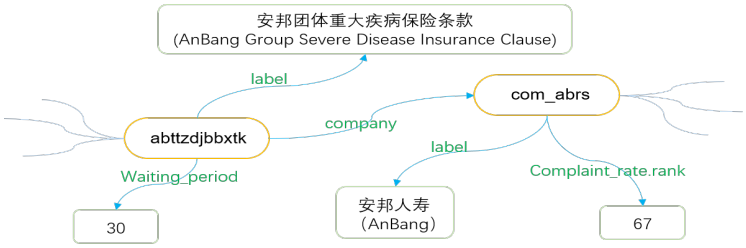

Fig. 1. Example knowledge base fragment.

factual question contains an entity mention and its potential answer is directly connected to the entity in knowledge base $\mathcal{K}$.

We further divide questions into open domain questions $\mathbf{Q}_{open}$ and domain-specific questions $\mathbf{Q}_{domain}$. $\mathbf{Q}_{open}$ does not focus on any domain and can be easily acquired on the Internet, while $\mathbf{Q}_{domain}$ is closely related to a particular field (e.g., *insurance*), which is difficult to be acquired in large scale.

Definition 3 (Domain Adaptive Question Answering over Knowledge Base, DKBQA). *Given a set of open domain questions $\mathbf{Q}_{open}$, a domain specific question $\mathbf{Q}_{domain}$ and a domain knowledge base $\mathcal{K}$, DKBQA aims to detect the key entity and property phrases of the question by analyzing the open domain questions $\mathbf{Q}_{open}$, and get answer directly by knowledge base retrieving with a structured query which has mapped the key phrases into $\mathcal{K}$.*

Since the limited number of domain questions affects the performance of DKBQA, we apparently expect to utilize the large amount of open domain questions. Then how to make the domain adaption becomes the critical issue.

3 Method

The architecture of DKBQA system is a pipeline paradigm, involving the following five steps as shown in Fig. 2: (1) representing the question as a pattern sequence by the existing POS and Parsing tools; (2) recognizing the key entity phrase $\mathbf{p}_e$ and property phrase $\mathbf{p}_p$ in $\mathbf{Q}_{domain}$ by a key phrase detection model trained on the open domain questions $\mathbf{Q}_{open}$; (3) generating candidate properties through knowledge base retrieving; (4) matching one property by calculating the similarity between $\mathbf{p}_p$ and the candidates; (5) retrieving $\mathcal{K}$ to get answer.

3.1 Question Pattern

We try to find a representation method to catch the key semantic information and merge questions into some patterns. Dependency parsing is a good feature because it is a strong abstract expression of sentence and dependency grammar can describe the relationship between a head and its dependents which

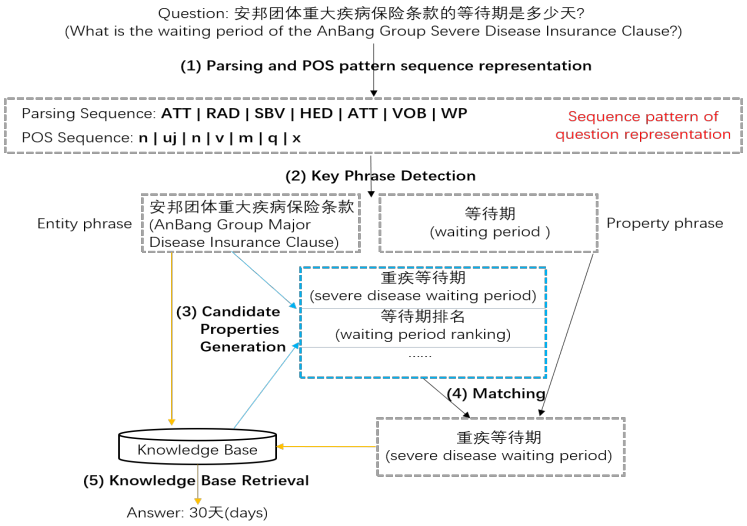

Fig. 2. Architecture of DKBQA system

can be naturally mapped into a semantic expression. For convenience, we only use the sequence of syntactic tags as the pattern ignoring the dependency information. To enhance the ability of sentence representation, we add the POS tag feature into sequence pattern. Through observation and preliminary statistics, the Parsing and POS tags patterns can effectively classify questions with the same keyword position so that we can transform different questions with the same Parsing and POS tags into one sequence pattern. As question pattern aims to classify questions with token sequence ignoring the token meaning, the representation is insensitive to the accuracy of the Parsing and POS.

3.2 Key Phrase Detection

By observing questions in various domains, we find that no matter which domain the questions belong to, they almost share the same syntactic form with different domain terminologies. Based on this observation, the domain phrase detection can be divided into two sub-problems: detect the key phrases in the open domain questions by the Parsing and POS sequence pattern, and adapt the common pattern forms from open domain to the specific domain via terminology transfer.

Key Phrase Detection in Open Domain. Phrase detection can be treated as a sequence labeling problem. As shown in Fig. 3, our goal is to build a tagging system which accepts a question pattern sequence as input and outputs the key phrases positions. Specifically, we employ a dependency parser tool (LTP) [4] to represent a question into the Parsing and POS pattern, and a Bi-RNN model to

predict the entity and property phrases. The model is learned from pairs of question pattern and its corresponding golden phrase locations from the manually annotated training data. As using the question pattern instead of the natural language form, the annotation work becomes easy and simple. Given an input question pattern $\mathbf{X}$ with annotated phrase locations, we first transform $\mathbf{X}$ into a one-hot vector. In this step, we divide the question pattern $\mathbf{X}$ into two forms which are the Parsing Sequence $\mathbf{X}_{par}$, and the POS Sequence $\mathbf{X}_{pos}$. After question representation, we use two Bi-RNN models to encode the pattern sequences into hidden vectors $\mathbf{V}_{par}$ and $\mathbf{V}_{pos}$ separately. Then we concatenate the two vectors as $\mathbf{V}_e$, and decode $\mathbf{V}_e$ via a linear layer. At last, the network outputs a probability of each position being entity and property with a $softmax$ layer.

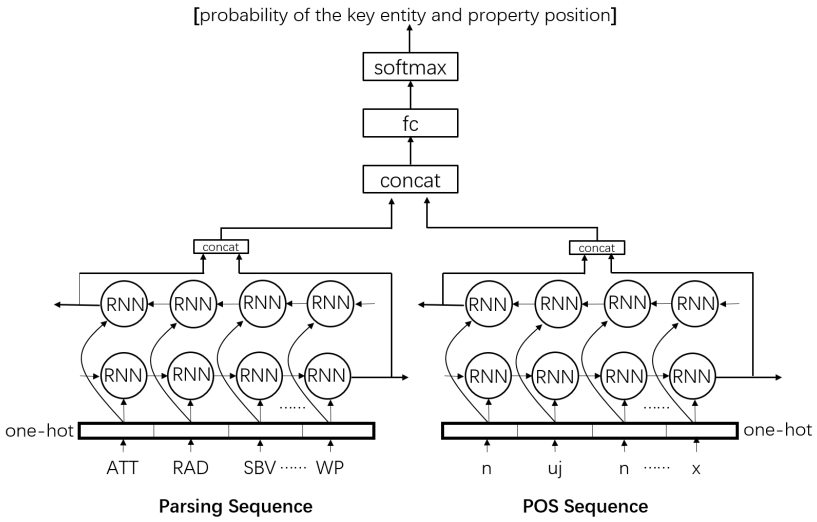

Fig. 3. The architecture of the key phrase detection model

Terminology Transfer. In this subsection, we expect to adapt the above model into a specific domain by constructing a terminology lexicon, making the specific domain questions have similar sequence patterns and characteristic distributions in the Parsing and POS feature space. Intuitively, we can directly employ the labels of the entities in domain knowledge base $\mathcal{K}$ as the terminology lexicon items by adding the POS tags and high frequencies. For the universality consideration, we do not attempt to build an additional synonym list for entities. Instead, we add some domain-specific terms obtained from domain related documents in order to increase the dictionary coverage, based on the assumption that these terms may become the unit of query phrases. We obtain domain-specific terms through a term extraction method proposed by Pan[9]. Through termi-

nology transfer, we can make the POS and Parsing results close to the open domain ones especially for some long or low-frequency phrases, and improve the availability of the prediction model in the specific domain. An example is shown as Fig. 4.

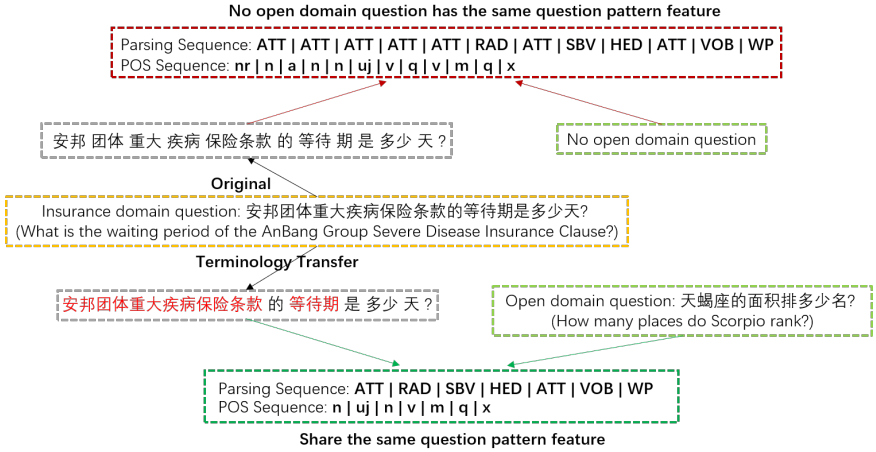

Fig. 4. An example of terminology transfer

3.3 Knowledge Base Search

Key phrase detection recognizes the KB-independent key phrases of the query which reflects the user intention. To connect the query vocabulary to semantic items in knowledge base $\mathcal{K}$, we construct a lexicon $\mathbf{L}$ which consists of an entity lexicon $\mathbf{L}_E$ and a property lexicon $\mathbf{L}_P$. Then we utilize a character-vector based fuzzy query method to retrieve the lexicon.

Character Vector Lexicon. The lexicon $\mathbf{L}$ is consist of two parts: 1) the labels and IDs of the entities or properties in $\mathcal{K}$ just like the terminology lexicon, and 2) the character vector presentation for each label. As our domain knowledge base is in Chinese, we utilize Chinese Wiki corpus to train the character vector via the CBOW [8] model. The difference from conventional word2vec is that the training is conducted on single characters instead of words. We add all the character vectors together to represent a phrase. The properties in a domain-specific knowledge base are always better distinguished than the open domain one as its small size and obvious domain boundaries.

Matching. We retrieve the knowledge base by a fuzzy matching method. Firstly, get an ID of the most related entity with the query subject phrase. Secondly,

generate a set of candidate properties from all the properties directly connected with the entity. And then we can select a property by the same fuzzy matching method as entity retrieval. Once the entity and property are obtained, we can form a structured query to get the result directly by knowledge base retrieval. Fuzzy matching is implemented with the cosine similarity between query vocabulary and semantic items.

4 Experiment

4.1 Dataset

Open Domain Dataset. Open domain questions are chosen from the training set of the NLPCC 2016 KBQA task[6], which contains 14,609 Chinese questions. Specifically, we select 2,646 questions as training set and 500 questions as test set for manual annotation. In order to provide a good generalization ability, we make some statistics of the whole questions to preserve the Parsing and POS features distribution. We find that the Parsing pattern has a strong generalization ability, i.e., 36 parsing patterns can present 4,127 questions. The generalization ability of the POS pattern is weak, but it can provide a good recognition ability.

Specific Domain Datasets. Specific domain datasets include two parts. One is the insurance domain dataset, the other is the China Mobile products dataset. Each of them contains 100 manually generated questions, and all answers can be found in the domain knowledge bases. For each query intention, there are some diverse forms of natural language representation.

4.2 Experiment Settings

Baseline Approaches. For key phrase detection, we utilize a traditional method based on TF-IDF and templates as the baseline (TPL+TFIDF) which first divides the questions into entity and property parts by manually constructed templates, extracts the keywords by TF-IDF separately, and merges the high-scoring and adjacent ones to a phrase as the result.

For domain KBQA, we construct a system based on templates and n-gram as the baseline (TPL+NGram). In TPL+NGRam, several templates is constructed to recognize the candidate entities and properties. Then we retrieve the knowledge base with a variety of combining forms of the entity and property generated by 3-gram method. To make the comparison fair, the NLP tools and dictionaries are same among all methods.

Evaluation Metrics. We use the Accuracy to evaluate the key phrase detection, and Precision, Recall and F1-score to evaluate the overall performance.

[6] http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html

Settings. For each question, we use jieba[7] to get the POS sequence because of its flexible user dictionary and LTP[8] to get the Parsing sequence. Then we represent pattern sequences in one-hot forms. The parameters in key phrase detection model are: learning rate=0.05, dropout rate=0.1, RNN layers=2, activation is ReLU, the loss function is cross entropy and the optimizer is SGD. The models are implemented by mxnet[9].

4.3 Results

Key Phrase Detection. We utilize both open and specific domain datasets to evaluate the performance of key phrase detection model respectively. The result is shown in Table 1. "CMCC" represents China Mobile products domain. "Ent_Acc" and "Pro_Acc" represent the accuracy of key entity and property phrases detection respectively. "Dict" represents domain dictionary. "TT" is short for the terminology transfer method. The results verify that our model works well in both open and specific fields.

Table 1. Key phrase detection results

	Open domain		Insurance domain		CMCC domain	
	Ent_Acc	Pro_Acc	Ent_Acc	Pro_Acc	Ent_Acc	Pro_Acc
$TPL + TFIDF$	47.0	58.8	-	-	-	-
$TPL + TFIDF + Dict$	-	-	49.0	31.0	74.0	39.0
$Ours - POS - TT$	77.8	67.4	43.0	29.0	44.0	36.0
$Ours - TT$	**80.4**	**70.4**	43.0	30.0	45.0	39.0
$Ours - POS$	-	-	52.0	37.0	**77.0**	62.0
$Ours$	-	-	**54.0**	**48.0**	**77.0**	**63.0**

Domain KBQA Results. We utilize insurance and CMCC domain questions to evaluate the performance of DKBQA system over the pre-constructed domain knowledge bases. The results are shown in Table 2. Compared with the baseline2, DKBQA achieves better Recall and F1 score. Although the outputs of the key phrase detection model still have some errors, the errors are not completely invalid and can still contain some words of the key phrases. Through the character-based fuzzy matching method, we can correct the previous errors with a high probability, and achieve a better result finally.

[7] https://pypi.org/project/jieba
[8] https://www.ltp-cloud.com
[9] https://mxnet.incubator.apache.org

Table 2. Domain KBQA results

	Insurance domain			CMCC domain		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
$TPL + NGram$	**95.7**	47.0	63.0	**92.8**	52.0	66.7
$DKBQA$	90.5	**74.0**	**81.4**	90.8	**79.0**	**84.5**

5 Related Work

The approaches of KBQA can be divided into two categories according to different question representations: symbol representation based and distributed representation based approaches.

Symbol representation based approach aims to parse natural language questions into structured queries such as Sparql or Sql. One kind of approach uses a domain-independent meaning representation derived from the combinatory categorical grammar (CCG) parse.Another kind of approach tries to get standard query by the matching between questions and templates which are manually constructed based on knowledge base [6,2]. These methods all have these two deficiencies: 1) the construction of CCG dictionary or templates is a very complex and time-consuming job; 2) Dictionaries and templates are difficult to cover the diversity of natural languages. Yin et al. [11] present a semantic parsing method via staged query graph generation which leverages the knowledge base in an early stage to prune the search space. This method is widely used because it simplifies the semantic matching issue.

Distributed representation based approach utilizes the distributed vectors to represent the question and knowledge base, then learns a rank model with existing QA pairs to score each candidate answer. Bordes et al. [3] attempt to embed questions and answers in a shared vector space. With the development of neural network technology, the distributed representation based approach is becoming more and more widely used [12].

Domain adaptation describes the task of learning a predictor in a target domain while labeled training data only exits in a different source domain [10]. A common method first learns an input representation with both domain corpus by autoencoder, then trains the predictor with the representation of the labeled source domain dataset [7,5].

6 Conclusion and Future Work

In this paper, we present a domain adaptive approach to construct a domain KBQA system with the open domain questions. Our system achieves good performance on both insurance and CMCC domain datasets. In the future, we would like to extend our method to deal with complex questions and improve the domain adaptability.

Acknowledgements

The work is supported by NSFC key projects (U1736204, 61533018, 61661146007), Ministry of Education and China Mobile Joint Fund (MCM20170301), a research fund supported by Alibaba Group, and THUNUS NExT Co-Lab.

References

1. Abujabal, A., Yahya, M., Riedewald, M., Weikum, G.: Automated template generation for question answering over knowledge graphs. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1191–1200 (2017)
2. Adolphs, P., Theobald, M., Schäfer, U., Uszkoreit, H., Weikum, G.: YAGO-QA: answering questions by structured knowledge queries. In: Proceedings of the 5th IEEE International Conference on Semantic Computing. pp. 158–161 (2011)
3. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 615–620 (2014)
4. Che, W., Li, Z., Liu, T.: LTP: A chinese language technology platform. In: COLING 2010, 23rd International Conference on Computational Linguistics, Demonstrations Volume. pp. 13–16 (2010)
5. Chen, M., Xu, Z.E., Weinberger, K.Q., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: Proceedings of the 29th International Conference on Machine Learning (2012)
6. Fader, A., Zettlemoyer, L.S., Etzioni, O.: Paraphrase-driven learning for open question answering. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 1608–1618 (2013)
7. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th International Conference on Machine Learning. pp. 513–520 (2011)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
9. Pan, L., Wang, X., Li, C., Li, J., Tang, J.: Course concept extraction in moocs via embedding-based graph propagation. In: IJCNLP(1). pp. 875–884. Asian Federation of Natural Language Processing (2017)
10. Wiese, G., Weissenborn, D., Neves, M.L.: Neural domain adaptation for biomedical question answering. In: Proceedings of the 21st Conference on Computational Natural Language Learning. pp. 281–289 (2017)
11. Yih, W.T., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (2015)
12. Zhang, Y., Liu, K., He, S., Ji, G., Liu, Z., Wu, H., Zhao, J.: Question answering over knowledge base with neural attention combining global knowledge information. CoRR **abs/1606.00979** (2016)