Target Detection and Knowledge Learning for Domain Restricted Question Answering

Mengdi Zhang(✉), Tao Huang, Yixin Cao, and Lei Hou

Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
mdzhangmd@gmail.com, {huangtao,caoyinxin,houl10}@mails.tsinghua.edu.cn

Abstract. Frequent Asked Questions(FAQ) answering in restricted domain has attracted increasing attentions in various areas. FAQ is a task to automated response user's typical questions within specific domain. Most researches use NLP parser to analyze user's intention and employ ontology to enrich the domain knowledge. However, syntax analysis performs poorly on the short and informal FAQ questions, and external ontology knowledge bases in specific domains are usually unavailable and expensive to manually construct. In our research, we propose a semi-automatic domain-restricted FAQ answering framework SDFA, without relying on any external resources. SDFA detects the targets of questions to assist both the fast domain knowledge learning and the answer retrieval. The proposed framework has been successfully applied in real project on bank domain. Extensive experiments on two large datasets demonstrate the effectiveness and efficiency of the approaches.

Keywords: Frequent Asked Questions · Domain restricted · Domain knowledge learning · Target-word · Question answering

1 Introduction

With the blossom of web-based services, some companies and organizations post online Frequent Asked Questions(FAQ) pages(a list of typical questions and resolved answers pairs), to provide consumers with timely information to common concerns. However, the proliferation of questions tends to overwhelm the user in skimming. Therefore, building a domain-restricted automated FAQ answering framework is in urgent need.

FAQ, a list of typical questions and experts' answers, offers users an online resource to respond to common concerns. The inclusion of FAQ pages enhances consumers' experience and increases potential purchases [14,16,15]. Take the the scenario in a bank FAQ page for example: a novice customer inquires *"Is it free to use mobile bank?"*("我想开通手机银行, 花钱吗?"), then the FAQ page match a common question called *"Does the 95959 mobile-bank charge?"*("95959 手机银行是否收费?") and returns the responding answer to the user. Obviously, the more timely and precise the FAQ answering service responds, the more possible the user choose this company's the products.

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 325–336, 2015.
DOI: 10.1007/978-3-319-25207-0_27

However, it is not an easy task. Questions asked by users are usually short, informal and expressed in different words to refer to the same concept. Most researches use syntax and ontology information to analyze questions. However, those questions are usually too short and informal to be well parsed, and ontology resources are usually unavailable and costly to construct. As a subclass problem of Question Answering, FAQ answering in restricted domain often exhibits different challenges compared to traditional question answering in open domain:

Question Understanding. To understand the questions, most researches like [2,25] use syntactic parsing via some language-specific parsers. However, the performance of the whole system is limited by the precision of the parser. First, the parser relies on a specific language, which implies a difficulty to extend to other languages. Second, expressed in spoken language in most cases, the questions from users couldn't be parsed well via existing parser tools. Third, those questions are quite short to contain complete syntactic elements. Previous studies show that users' questions are usually limited in length [5,7]. In our corpora, the average question lengths are about 20 and 10 Chinese characters respectively (as showing in Fig. 2). Thus, question analysis methods that heavily rely on syntax may not be appropriate in here.

Vocabulary Gap. Due to diverse background knowledge, users don't share the same vocabulary with each other as well as the experts. For example, the user's question *"Since my friend is abroad, can I call the bank to send money to him?"*("朋友在国外, 我打电话到银行可以给他汇钱吗?") actually refers to the formal question in FAQ *"Does the telephone banking's transfer support foreign currency?"*("电话银行对外转账是否支持外币?"). Considerable methods focus on employing an ontology to represent the questions and answers [27,23,2]. Through domain ontology, semantic distance is calculated between words of user's question and FAQ. However, the domain-specific ontology may be unavailable, and is difficult to construct and reuse. Therefore, how to bridge the gap without external resources remains a challenge.

Despite of the above challenges, once a practical FAQ answering framework is successfully formed, its impact for commerce is tremendous. Based on these observations, we propose the semi-automated domain-restricted FAQ answering framework(SDFA) that addresses all above challenges: SDFA collects users' questions from the system log to expand question in FAQ; proposes the target-word concept to understand question; fast learn a lightweight knowledge structure from the FAQ source itself to further bridge the vocabulary gap; all the gradually obtained information and knowledge are carefully designed as evidence to the final answer retrieval. The contributions of our work can be summarized as follows:

- We propose a novel semi-automatic framework SDFA to tackle the restricted domain FAQ answering issue.
- We design a semi-automatic pipeline to learn domain knowledge by clustering the FAQ, instead of from extra resources.

– Our framework has been successfully applied in real project on bank domain. Extensive experiments on two large datasets demonstrate the effectiveness and efficiency of the approaches.

2 Problem Definition

In this section, we first formalize the FAQ answering problem, and then demonstrate our proposed techniques.

2.1 Problem Definition

We present required definitions and formulate the problem of FAQ answering in restricted domain. Without loss of generality, we assume the domain FAQ corpus S has been well collected as a input, along with another input, user's question q. Our goal is to return the best matched answer in FAQ for a user's query.

Definition 1 ***QA Pair.*** *QA Pair* $p_i =< Q_i, A_i >$ *is a pair of a typical question* Q_i *and its responding answer* A_i*. FAQ corpus* S *is a set of QA Pair, i.e.,* $S = \{p_i | i = 1, 2.., n\}$.

Both the query and question in FAQ are one interrogative sentence in a list of words. From observation, some words are informative enough to stand for the whole question, while some words are less important, even noising. To detect question's intention, we distinguish those words by define a concept called target-word:

Definition 2 ***Target-word.*** *We define the target-word* w^t *as a word which can stand for the main meanings of the question, i.e., user's intention. There are usually more than one target-word in a question. We represent a question* $Q_i = \{w_1, w_2, ..w_m\}$ *in a ranked list of target-words* $Q_i^t = \{w_1^t, w_2^t, .., w_k^t\}$*, where* $k \leq m$.

Cases of target-words are showing in early example: *"mobile bank"* ("手机银行") and *"free"*("收费") are actually the intentions of the user and should be treated as two target-words of the query. With a further analysis on the relations between words, two kinds of target-word can be summarized: one is actually service's name, and the other is the service's property. We define them as the domain knowledge.

Definition 3 ***Domain Knowledge.*** *We define a domain knowledge structure embedded in the questions: service category and its properties, as* $< C, P_1, P_2, ..P_c >$*. A question can be categorized by this two-layer labels. Domain knowledge* $K = \{< C_i, P_{i1}, P_{i2}, ..P_{ic} > | i = 1, 2, ...d\}$ *is defined for FAQ corpus* S*, where* d *means the number of services included in* S.

The domain knowledge in early example of user's question in bank domain is the service name *"mobile-bank"* ("手机银行"), and its properties *"charge"* ("收费"). Ultimately, the goal of our task is to find a QA Pair whose question is similar to the query, and return the responding answer in that QA Pair to the user. In terms of similarity, it means they are both questions about the same category's same properties, and share similar question's ranked target-word list. Based on these definitions, we define the task of utilizing domain knowledge in FAQ to answer user's query as follow:

Problem 1 ***Data-driven FAQ Answering.*** *Given a FAQ corpus* $S = \{< Q_i, A_i > | i = 1, 2.., n\}$ *and a user's query* q*, the goal is to firstly detect the target-word and learn the domain knowledge* K *from* S*, and finally find a list of QA Pair* $p \in S$ *for* q*, ranked by a function* $Score(q, p)$ *which measures the similarity between* p *and* q *based on the target-words and domain knowledge obtained previously.*

3 Framework

To tackle the problem described above, we propose a semi-automatic FAQ answering framework SDFA. Fig.1 shows the proposed framework which consists of a series of offline preparation on FAQ and online operations on user's query. Besides a common preprocess module, the offline mainly consists of target-word detection and domain knowledge learning, and the online mainly consists of the question analyzing and answer retrieval, which utilize the offline outputs like target-word model and domain knowledge. Additionally, after the final answer returning to user, if the user is satisfied with the final answer, then his/her query will be added to the FAQ corpus, as extension of the corresponding question.

The preprocessing are completed by traditional NLP techniques. Thus target-word detection, domain knowledge learning and answer retrieval are three major modules, and we'll present the details in the following section.

3.1 Target-Word Detection

In this section we model the target-word to detect user's intention. What kind of word should be recognized as target-word? Naturally, we should firstly exclude all obvious non-informative words by text preprocessing, including the removal of polite words, 1 length Chinese words and some personal pronouns. Ruling out these must-not-be word, we get the candidates of target-words.

Given a word, according to the structural and linguistic characteristics of questions, both lexical and semantic features are used to pick up the the target-word, including (1)the word itself, (2)word location in the question, (3)word length, (4)term frequency in the retrieval corpus, (5)POS tag of the word itself, (6)POS tag of the word ahead and (7)POS tag of the word after.

We use **Logistic Regression** [6] to assess the probability of each word to be a target-word in the question. In logistic function, the probability of a word w to be target-word is defined as follow:

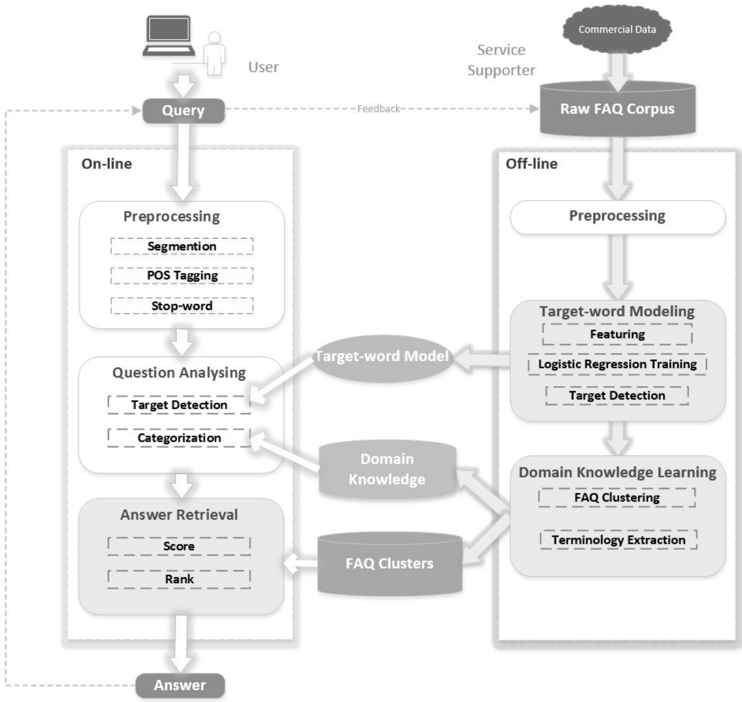

Fig. 1. An overall architecture for SDFA.

$$P(Y=1|X=x_w)=\frac{1}{1+e^{-(\alpha x_w+\beta)}} \tag{1}$$

where x_w is the feature vector describing attributes of word w; α are weights of the attributes and β is a bias, both of which are learned by maximizing the objective function O:

$$O(\alpha,\beta)=\prod_{w\in W} P(Y=1|X=x_w)^{y_w}P(Y=0|X=x_w)^{1-y_w} \tag{2}$$

where y_w is the label of w in the training set.

Through the target identification module, we rank the words in question according to their target-word probability, highlighting the informative points of the question.

3.2 Domain Knowledge Learning

Domain Knowledge is crucial to Restricted Domain Question Answering(RDQA). Recent researches use ontology in domain to calculate semantic distance between words when matching query to FAQ. However, external ontology knowledge bases in specific domains are usually unavailable. Additionally, manually building a domain knowledge base is laboriously expensive and highly subjective.

Meanwhile, through observation on FAQ's target-words obtained from the target-detection module, the FAQ corpus itself actually carries abundant domain knowledge about the services: service's name and service's properties. As we define before, they are domain knowledge $K = \{< C_i, P_{i1}, P_{i2}, ..P_{ic} > | i = 1, 2, ...d\}$ in FAQ corpus S. Therefore, we propose a semi-automatic and data-driven approach to learn the domain knowledge K by two steps: (1) FAQ clustering, and (2) Terminology Extraction.

FAQ Clustering. We partition the FAQ corpus by clustering questions in FAQ, to obtain the whole C_i categories. We represent questions as TF-IDF vectors and employ an incremental DBSCAN algorithm for clustering. Specially, we select Cosine Distance as distance metrics for DBSCAN; we run DBSCAN in a few cycles to re-clustering questions in bad clusters and un-clustered questions left by DBSCAN, which we define as the remaining question set of each loop; a maximum number L of the loop and a minimum size R of each loop's remaining questions are set to control the clustering circulation.

Terminology Extraction. By last step, we get k clusters of FAQ questions partitioned by the domain services. This step is to label the C and P for each cluster from the candidate pool of ranked target-words list of the cluster: firstly, the C of a cluster is selected from the top-1 target-words from all questions in the cluster ranked by frequency then, the properties P are selected from remain target-words; execute above operations for each cluster, and finally the domain knowledge K is constructed. The domain knowledge learned can be treated as a two-layer terminologies structure of services scattered in FAQ, as showing in Table 1. Each entry in the domain knowledge responds to a specific service, consisting of a category name C and its properties P_i.

Table 1. A snippet domain knowledge of banking

Category	**Properties**
phone-bank	query open-account register close-account
电话银行	查询 开户 注册 销户
text-bank	password binding query
短信银行	密码 绑定 查询
e-pay	query close-account register remittance
e支付	查询 注销 注册 汇款
noble-metal	sale price buying specification
贵金属	销售 零售价 购买 规格
fund	investment custody subscribing purchase
基金	定投 托管 认购 申购

Indexed by the learned domain knowledge, each question in FAQ is off-line labeled with a set of C and P IDs if it contains the corresponding terms in K, an equally categorization process. The input query is on-line categorized in the same way.

3.3 Retrieval Model

Given a question, the main goal of the retrieval model is to find the most relevant QA Pair in FAQ and output the answer as the final answer. Based on the target and category information obtained previously, the probability of relevance between the query and each candidate QA Pair is calculated through a target-word based BM25 algorithm. Thus, we can rank the QA Pairs in the probability of relevance, and then strategically return matched answer according to the query's categories. For example, if a query contains only one category label, we will return the top1 answer to the user; if a query contains multiple category labels, we will return the top3 or top5 answers depending on the circumstances.

The retrieval model includes two steps: (1) categorize user's query based on domain knowledge, and (2) find the related QA Pair by the advanced BM25 algorithm.

Candidate Documents. Clearly, the relevant documents should have the same categories and properties. For example, the question *"How to log off personal mobile bank?"*(" 如何注销个人手机银行?") will be classified to *"Mobile Bank"* concept with *"Log off"* action, which should be retrieved under FAQs whose concept and action label are the same. Therefore, only a subset of the whole collection that shares the common category and property with the question is worth retrieving, which we called candidate documents.

Target-word Based BM25. BM25 [17] is a probabilistic relevance framework and has been widely used in text-retrieval area. It considers each term of the question as an independent unit, and the final probability of relevance between the document and the question is proved to be proportional to the weighted sum of all the terms. The weight of term is computed in traditional BM25 as follows.

$$w_i(tf) = w_i(idf) * \frac{tf}{tf + K} \tag{3}$$

In Eq.3, $w_i(idf) = log\frac{N - n_i + 0.5}{n_i + 0.5}$ is a close approximation to the classical idf, where N is the size of the whole collection of the documents and n_i is the number of documents containing term t_i. For K in Eq.3,

$$K = k_1 * (1 - b + b * \frac{dl}{avdl}) \tag{4}$$

dl is the document length and $avdl$ is the average length of all the documents. Note that b is used to control the document length normalization, and k_1 is to smooth the special term frequency. Particularly, for high K, increments in tf contribute significantly to the relevance, whereas for low K, the additional contribution of another occurrence of the word decreases rapidly.

As the term weight, TF-IDF provides a statistical perspective; however, target words indicate the semantic importance. To model this feature, we define the final probability of relevance as follows:

$$P(rel|q,p) \propto \sum_{q,tf} \lambda_i * w_i(tf) \tag{5}$$

where

$$\lambda_i = \begin{cases} \eta & tf_i > 0, i \in q \\ -\gamma & tf_i > 0, i \notin q \end{cases} \tag{6}$$

indicates the degree of being the target of the question. To be specific, if the term is a target word of the question, it will be rewarded; otherwise, when the term that only occurs in the QA pair is also a target word, it will be punished. To differentiate the importance of standard question and extended question, we use $\widehat{tf} = \tau * tf$ instead of the normal term frequency, where $\tau = 2, 1, 2$ denotes standard question, extended question and standard answer respectively.

4 Experiments

4.1 Data Preparation

In the experiment, we take bank as the restricted domain. Since there is no benchmark dataset for bank domain, we construct two datasets from two different banks. Each dataset contains both the standard FAQ and the extended questions in spoken language, collected from the corresponding bank's consult log. In this way, each standard question in FAQ is corresponding to a couple of extended questions (after the system is running, the user's query can be added to the extended question and automatically accumulate). Both the datasets come from the needs of real projects in life. The total statistics of two datasets are showing in Table 2 and the distributions of question's length in these two FAQ set are showing in Fig.2.

Table 2. Statistics on Datasets.

	#QA Pairs	#Extended Questions	#Test Set	#Target-word Train Set
Bank1	48,495	127,026	4,336	2,272
Bank2	2,399	42,404	5,536	500

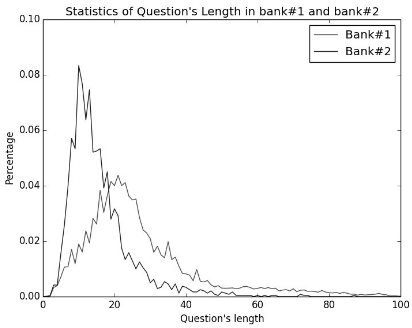

Fig. 2. The Distribution of Question's Length.

4.2 Evaluation Measures

Given a test question, our proposed methods return a list of QA Pair ranked in descending order of probability of relevance. Generally, we extract the answer from the top QA Pair as the expected answer. As such, we are typically interested in the $Precision@1$, which measures the percentage of the results whose top-1 answer is correct. Similarly, the $Precision@5$ measures the percentage of the results where the correct answer showing among top-5. Additionally, for those not hitting the top, we calculate the Mean Reciprocal Rank(MRR), the multiplicative inverse of the rank of the first correct answer: $MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$. Consider rank position $rank_i$, of the first relevant QA pair in the candidate rank list, then the Reciprocal Rank score is $\frac{1}{rank_i}$ and MRR is the mean RR across multiple test questions. The more close the MRR score to 1, the more possible the correct answer in the result is near to the top.

4.3 Experiments

To evaluate the effectiveness of each single module and the whole framework, we gradually designed a series of experiments and judged the performance based on the metrics as described in previous section.

In the target detection module, we use Logistic Regression to train the target-word model from a manually labeled question set and then detect the target-word of FAQ set. The second step is to learn domain knowledge and categorize each question. we run DBSCAN clustering on the FAQ set and then manually label each cluster with domain terminology guarded by the target-word detected from the first step. The DBSCAN clustering requires two parameters ε and $minPt$, which stands for physical distance and the minimum number of points required to form a dense region respectively. In our experiment, ε is set between 0.40 and 0.45, and $minPt$ is set as 100 for bank1 and 30 for bank2. The maximum clustering loop L is set as 5 for bank1 and 3 for bank2, and the minimum remains R is both set as 10%.

For the FAQ retrieval step, we execute experiments on five strategies:

(1)BM25. In the traditional BM25 model, we empirically initialize the model's parameters in Eq.4: the default parameter values we set are k1=1.2 and b=1 as recommended in [17].

(2)BM25^t. In the target-word based BM25 model, we set the K1 and b as same in (1). We use the target-word score as the term's weight of BM25 score in Eq.5 and we set the reword η between 1.2-1.5 in Eq.6.

(3)BM25^t+Class. We first learn domain knowledge from the FAQ corpus by clustering the QA Pair, obtaining the categories and their properties. Then use these learned two-layer domain knowledge to categorize each QA Pair's type in the FAQ corpus and the input question's type and finally execute the same retrieval process as (2), which means we map the question to certain document space then retrieve the best matched document in this small data space.

(4)BM25^t+Class+Punish. After executing the same process of strategy (3), we get a list of candidate QA Pairs, ranking in the positive relevance with the query. Before returning the tops to the user, we rerank this list by punishing the negative difference as described in Eq.6. We set the punish $-\gamma$ as -1.

(5)Cosine. As a baseline, we represent the query and each question of FAQ in TF-IDF vector and calculate cosine similarity of vectors as the ranking basis.

4.4 Results and Analysis

The overall results are showing in Table 3. As we can see, our method, the target-word based BM25 utilizing the domain knowledge and considering the term punishment, performs best on two datasets. Each method shows a consistent performance on all three evaluation measures. For simplicity, we only take the *Precision*@1 here for detailed analysis. The Cosine method performs steadily but limited: its *Precision*@1 is about 40% on two datasets. Traditional BM25 performs better than Cosine. After considering the target-word, the BM25^t has averagely improved by 2.5 percent. Interestingly, BM25^t with categorization utilizing the domain knowledge is a little bit lower than the BM25^t itself. However, the search time, though not showing here for the limited room, has reduced 50%. It indicates that domain knowledge can map the query to a much small search space to improve the efficiency, with a slight risk of filtering some correct answer simultaneously. Finally, by punishing the negative difference between the query and candidate QA Pairs, the rank of the correct answer can be further improved.

Our proposed strategy, the BM25^t+Class+Punish, performs the best on the two different bank datasets, which proves the practical and robustness of our framework.

Table 3. Overall results

Method	**Bank1**			**Bank2**		
	Precision@1	*Precision@5*	*MRR*	*Precision@1*	*Precision@5*	*MRR*
Cosine	41.3%	64.5%	55.7%	45.4%	68.1%	57.1%
BM25	61.1%	79.4%	68.2%	62.8%	84.3%	70.3%
BM25^t	63.6%	81.7%	70.0%	64.2%	87.0%	73.9%
BM25^t +Class	63.5%	81.3%	69.8%	64.1%	86.7%	73.6%
BM25^t+Class+Punish	66.6%	84.1%	73.9%	65.3%	88.2%	74.6%

5 Related Work

There are several lines of researches that are related to our work, and we present some of the related literatures as follows:

Frameworks and architecture of domain restricted QA system [13,12,22,3] have been proposed, mostly based on text, yet rarely concerned with question answer pairs format data in FAQ [4]. Instead of extracting answers from free text, the QA systems on FAQ focus on retrieving the most relevant QA Pair in respect to the user's question. In FAQ systems, the main categories include NLP-based [23,26,24], statistical-based [9,10] and template-based methods [18,19,20]. Researches on question similarity calculation include cosine similarity on TF-IDF vectors, BM25 and etc. [21,1,8]. Our work focuses upon bank service FAQ answering system and combines NLP and statistic methods.

There is one closely related work [11], which proposes a cluster-based retrieval system Fract. Fract clusters the query logs into predefined FAQ categories and extract weight scores of potentially occurring words from the clusters by using LSA techniques. During retrieval time, Fract extracts important terms from query by parsing and expand these terms with the potentially occurring words to help in ranking relevant FAQs. Differently, we propose a new concept called Target-word to detect user's intention and cluster the FAQ corpus, instead of the query logs, to learn the domain knowledge. Both these information and knowledge are carefully designed as evidence combined into an adjusted target-word based BM25 score function to retrieve final answer.

6 Conclusion

In our research, we propose a semi-automatic domain-restricted FAQ answering framework SDFA, without relying on any external resources. SDFA detects the targets of questions to assist both the fast domain knowledge learning and the answer retrieval. The proposed framework has been successfully applied in real project on bank domain. Extensive experiments on two large datasets demonstrate the effectiveness and efficiency of the approaches.

Acknowledgments. The work is supported by 973 Program (No. 2014CB340504), NSFC-ANR (No. 61261130588), Tsinghua University Initiative Scientific Research Program (No. 20131089256), Science and Technology Support Program (No. 2014BAK04B00), and THU-NUS NExT Co-Lab.

References

1. Achananuparp, P., Hu, X., Zhou, X., Zhang, X.: Utilizing sentence similarity and question type similarity to response to similar questions in knowledge-sharing community. In: Proceedings of QAWeb 2008 Workshop, vol. 214 (2008)
2. Basili, R., Hansen, D.H., Paggio, P., Pazienza, M.T., Zanzotto, F.M.: Ontological resources and question answering. In: Workshop on Pragmatics of Question Answering, held jointly with NAACL 2004. Citeseer (2004)
3. Benamara, F.: Cooperative question answering in restricted domains: the webcoop experiment. In: Proceedings of the Workshop Question Answering in Restricted Domains, within ACL (2004)
4. Burke, R.D., Hammond, K.J., Kulyukin, V., Lytinen, S.L., Tomuro, N., Schoenberg, S.: Question answering from frequently asked question files: Experiences with the faq finder system. AI Magazine **18**(2), 57 (1997)
5. Chau, M., Fang, X., Liu Sheng, O.R.: Analysis of the query logs of a web site search engine. JASIST **56**(13), 1363–1376 (2005)
6. Cox, D.R., Snell, E.J.: Analysis of binary data, vol. 32. CRC Press (1989)
7. Croft, W.B., Cook, R., Wilder, D.: Providing government information on the internet: Experiences with thomas. In: DL (1995)
8. Guo, X.Y., Zhang, B.F., Fang, A.G., Zhou, C.F.: Research on question similarity algorithm for intelligent question answering system and its implementation. Computer Applications **2**, 449–452 (2005)

9. Jijkoun, V., de Rijke, M.: Retrieving answers from frequently asked questions pages on the web. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 76–83. ACM (2005)
10. Kim, H., Seo, J.: High-performance faq retrieval using an automatic clustering method of query logs. Information Processing & Management **42**(3), 650–661 (2006)
11. Kim, H., Seo, J.: Cluster-based faq retrieval using latent term weights. IEEE Intelligent Systems **2**, 58–65 (2008)
12. Ma, Z., Sun, A., Yuan, Q., Cong, G.: A tri-role topic model for domain-specific question answering (2015)
13. Mollá, D., Vicedo, J.L.: Question answering in restricted domains: An overview. Computational Linguistics **33**(1), 41–61 (2007)
14. Mustaffa, S., Beaumont, N.: The effect of electronic commerce on small australian enterprises. Technovation **24**(2), 85–95 (2004)
15. Page, C., Lepkowska-White, E.: Web equity: a framework for building consumer value in online companies. Journal of Consumer Marketing **19**(3), 231–248 (2002)
16. Pei, Y., Xue, W., Li, D., Chang, J., Su, Y.: Research on the relationship between empowerment and sales performance in e-commerce service industry enterprises. In: LISS 2014, pp. 959–967. Springer (2015)
17. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM SIGIR, pp. 232–241. Springer (1994)
18. Sneiders, E.: Automated faq answering: continued experience with shallow language understanding. In: Papers from the 1999 AAAI Fall Symposium on Question Answering Systems, pp. 97–107 (1999)
19. Sneiders, E.: Automated question answering using question templates that cover the conceptual model of the database. In: Andersson, B., Bergholtz, M., Johannesson, P. (eds.) NLDB 2002. LNCS, vol. 2553, pp. 235–239. Springer, Heidelberg (2002)
20. Sneiders, E.: Automated faq answering with question-specific knowledge representation for web self-service. In: 2nd Conference on Human System Interactions, HSI 2009, pp. 298–305. IEEE (2009)
21. Song, W., Feng, M., Gu, N., Wenyin, L.: Question similarity calculation for faq answering. In: Third International Conference on Semantics, Knowledge and Grid, pp. 298–301. IEEE (2007)
22. Vila, K., Ferrández, A.: Model-driven restricted-domain adaptation of question answering systems for business intelligence. In: Proceedings of the 2nd International Workshop on Business Intelligence and the WEB, pp. 36–43. ACM (2011)
23. Wang, F., Teng, G., Ren, L., Ma, J.: Research on mechanism of agricultural faq retrieval based on ontology. In: Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2008, pp. 955–958. IEEE (2008)
24. Winiwarter, W.: Adaptive natural language interfaces to faq knowledge bases. Data & Knowledge Engineering **35**(2), 181–199 (2000)
25. Xu, K., Zhang, S., Feng, Y., Zhao, D.: Answering natural language questions via phrasal semantic parsing. In: Zong, C., Nie, J.-Y., Zhao, D., Feng, Y. (eds.) NLPCC 2014. CCIS, vol. 496, pp. 333–344. Springer, Heidelberg (2014)
26. Yang, S.Y., Chuang, F.C., Ho, C.S.: Ontology-supported faq processing and ranking techniques. Journal of Intelligent Information Systems **28**(3), 233–251 (2007)
27. Zajac, R.: Towards ontological question answering. In: Proceedings of the Workshop on Open-Domain Question Answering, vol. 12, pp. 1–7. ACL (2001)