Extracting Question-Context-Answer Triples from Online Forums

Shilin Ding Gao Cong Chin-Yew Lin Xiaoyan Zhu

Abstract

In this paper, we propose a new approach to extracting question-context-answer triples from online discussion forums. More specifically, we propose a general framework based on Conditional Random Fields (CRFs) for context and answer detection, and also extend the basic framework to utilize contexts for answer detection and to better accommodate the features of forums. Experimental results show that our techniques are very promising.

1 Introduction

Forums are web virtual spaces where people can ask questions, answer questions and participate in discussions. The availability of affluent thread discussions in forums has promoted increasing interests in knowledge acquisition and summarization for forum threads. Forum thread usually consists of an initiating post and a number of reply posts. The initiating post usually contains several questions and the reply posts usually contain answers to the questions and perhaps new questions. Forum participants are not physically co-present, and thus reply may not happen immediately after questions are posted. The asynchronous nature and multi-participants make multiple questions and answers interweaved together, which makes it more difficult to summarize.

In this paper, we address the problem of detecting question-context-answer triples from forums. Figure 1 gives an example of a forum thread with questions, contexts and answers annotated. It contains three *question* sentences, S3, S5 and S6. Sentences S1 and S2 are *context*s of *question* 1 (S3). Sentence S4 is the *context* of *question*s 2 and 3, but not 1. Sentence S8 is the answer to question 3. One example of question-context-answer triple is (S4-S5-S10). As shown in the example, a forum question usually requires contextual information to provide background or constraints. Moreover, it sometimes

<context id=1>**S1**: *Hi I am looking for a pet friendly hotel in Hong Kong because all of my family is going there for vacation.* **S2**: *my family has 2 sons and a dog.*</context> <question id=1>**S3**: *Is there any recommended hotel near Sheung Wan or Tsing Sha Tsui?*</question> <context id=2,3>**S4**: *We also plan to go shopping in Causeway Bay.*</context> <question id=2>**S5**: *What's the traffic situation around those commercial areas?*</question> <question id=3>**S6**: *Is it necessary to take a taxi?*</question>. **S7**: *Any information would be appreciated.*
<answer qid=1>**S8**: *The Comfort Lodge near Kowloon Park allows pet as I know, and usually fits well within normal budget.* **S9**: *It is also conveniently located, nearby the Kowloon railway station and subway.*</answer>
<answer qid=2,3> **S10**: *The traffic there is not so good, so I recommend MTR in Causeway Bay because it is cheap to take you around* </answer>

Figure 1: An example thread with question-context-answer annotated

needs contextual information to provide explicit link to its answers. For example, S8 is an answer of *question* 1, but they cannot be linked with any common word. Instead, S8 shares word *pet* with S1, which is a context of *question* 1, and thus S8 could be linked with *question* 1 through S1. We call contextual information the *context* of a question in this paper.

A summary of forum threads in the form of question-context-answer can not only highlight the main content, but also provide a user-friendly organization of threads, which will make the access to forum information easier.

Another motivation of detecting question-context-answer triples in forum threads is that it could be used to enrich the knowledge base of community-based question and answering (CQA) services such as Live QnA and Yahoo! Answers, where *context* is comparable with the question description while *question* corresponds to the question title. For example, there were about 700,000

questions in the Yahoo! Answers travel category as of January 2008. We extracted about 3,000,000 travel related questions from six online travel forums. One would expect that a CQA service with large QA data will attract more users to the service.

However, it is challenging to summarize forum threads into question-context-answer triples. First, detecting contexts of a question is important and non-trivial. We found that 74% of questions in our corpus containing 2,041 questions from 591 forum threads about travel need context. However, relative position information is far from adequate to solve the problem. For example, in our corpus 37% of sentences preceding questions are contexts and they only represent 20% of all correct contexts. To effectively detect contexts, the dependency between sentences is important. For example in Figure 1, both S1 and S2 are contexts of *question* 1. S1 could be labeled as context based on word similarity, but it is not easy to link S2 with the question directly. S1 and S2 are linked by the common word *family*, and thus S2 can be linked with *question* 1 through S1. The challenge here is how to model and utilize the dependency for context detection.

Second, it is difficult to link answers with questions. In forums, multiple questions and answers can be discussed in parallel and are interweaved together while the reply relationship between posts is usually unavailable. To detect answers, we need to handle two kinds of dependencies. One is the dependency relationship between contexts and answers, which should be leveraged especially when questions alone do not provide sufficient information to find answers; the other is the dependency between answer candidates (similar to sentence dependency described above). The challenge is how to model and utilize these two kinds of dependencies.

In this paper we propose a novel approach for summarizing forum threads into question-context-answer triples. We make the following contributions:

First, to our knowledge this is the first work on extracting question-context-answer triples from forum threads. We also found that context is very important for answer detection.

Second, we use classification method to identify questions from forum data as focuses of a thread, and then employ Linear Conditional Random Fields (CRFs) to identify contexts and answers, which can capture the relationships between contiguous sentences.

Third, to capture the dependency between contexts and answers, we introduce Skip-chain CRF model for answer detection. We also extend the basic model to 2D CRFs to model dependency between contiguous questions in a forum thread for context and answer identification.

Finally, we conducted experiments on forum data. Experimental results show that 1) our method for finding questions is effective; 2) Linear CRF model outperforms SVM and decision tree in both context and answer detection; 3) Skip-chain CRFs outperform Linear CRFs for answer finding, which demonstrates that **context improves answer finding**; 4) The combination of 2D CRFs and Skip-chain CRFs achieves better performance.

The rest of this paper is organized as follows: The next section discusses related work. Section 3 presents the proposed techniques. We evaluate our techniques in Section 4. Section 5 concludes this paper and discusses future work.

2 Related Work

There is some research on summarizing discussion threads and emails. Newman and Blitzer(2003) focused on clustering newsgroup messages into subtopic groups and extracting top-ranked sentences per group as summary. Motivated by the fact that one message may contain multiple issues, Zhou and Hovy (2005) segmented internet relay chat, clustered segments into subtopics, and identified responding segments of the first segment in each subtopic by assuming the first segment to be focus. Lam et al. (2002) employed a single document summarizer for email summarization. In (Nenkova and Bagga, 2003; Wan and McKeown, 2004; Rambow et al., 2004), email summaries were organized by extracting overview sentences as discussion issues. Carenini et al (2007) leveraged both quotation relation and clue words for email summarization. In contrast, given a forum thread, we extract questions, their contexts, and their answers as summaries.

We also note the existing work on extracting knowledge from discussion threads. Huang et al.(2007) used SVM to extract *input-reply* pairs from forums for chatbot knowledge. Feng et al. (2006a)

used cosine similarity to match students' query with reply posts for discussion-bot. Feng et al. (2006b) identified the most important message in online classroom discussion board. Our problem is quite different from the above work.

Detecting context for question in forums is related to the context detection problem raised in the QA roadmap paper commissioned by ARDA (Burger et al., 2006). To our knowledge, none of the previous work addresses the problem of context detection. The method of finding follow-up questions (Yang et al., 2006) from TREC context track could be adapted for context detection. However, the follow-up relationship is limited between questions while context is not.

Shrestha and McKeown (2004)'s work on email summarization is closer to our work. They used RIPPER as a classifier to detect interrogative questions and their answers and used the resulting question and answer pairs as summaries. However, it did not consider contexts of questions and dependency between answer sentences.

Extensive research has been done in question-answering, e.g. (Berger et al., 2000; Jeon et al., 2005; Cui et al., 2005; Harabagiu and Hickl, 2006; Dang et al., 2007). They mainly focus on constructing answer for certain types of question from a large document collection, and usually apply sophisticated linguistic analysis to both questions and the documents in the collection. In our scenario, we not only need to find answers for various types of questions in forum threads but also the questions themselves.

3 Question-context-answer Triple Detection

In this section we will first introduce the problem of finding question-context-answer triples from forums, and then present our approach.

Problem statement. A question is a linguistic expression used by a questioner to request information in the form of an answer. A question usually contains question focus, i.e. question concept that embodies information expectation of question (Lehnert, 1977), and constraints. The sentence containing question focus is called *question anchor* or simply *question* and the sentences containing only constraints are called *context*. Context provides constraint or background information to question.

As discussed in Introduction, identifying question-context-answer triples from forums is nontrivial. We approach the problem by first identifying questions in a thread, and then identifying the context and answer of every question within a uniform framework. In this section, we will first briefly present an approach to question detection, and then focus on context and answer detection.

3.1 Question Detection.

For question detection in forums, rules, such as question mark and 5W1H words, are not adequate. With question mark as an example, we find that 30% questions do not end with question marks while 9% sentences ending with question marks are not questions in a corpus. To complement the inadequacy of simple rules, in this paper we build a SVM classifier to detect questions using the features exploited in (Shrestha and McKeown, 2004).

3.2 Context and answer detection

Given a thread and a set of m detected questions $\{Q_i\}_{i=1}^m$, our task is to find the contexts and answers for each question. We first discuss using linear CRFs model for context and answer detection, and then extend the basic framework to Skip-chain CRFs and 2D CRFs to better model our problem. Finally, we will briefly introduce CRF models and the features that we used for CRF model.

3.2.1 Using linear CRFs

For ease of presentation, we discuss detecting contexts of the questions using linear CRF model. The model could be easily extended to answer detection (by using different features).

Context detection. As discussed in Introduction that context detection cannot be trivially solved by position information (See Section 4.2 for details), and dependency between sentences is important for context detection. Recall that in Figure 1 S2 could be labeled as context of Q1 if we consider the dependency between S2 and S1, and that between S1 and Q1, while it is difficult to establish connection between S2 and Q1 without S1. Table 1 shows that the correlation between the labels of contiguous sentences is significant. In other words, when a sentence Y_t's previous Y_{t-1} is not a context ($Y_{t-1} \neq$ C)

Contiguous sentences	$y_t = C$	$y_t \neq C$
$y_{t-1} = C$	1,191	1,366
$y_{t-1} \neq C$	1,377	62,446

Table 1: Contingency table(χ^2 = 13,044, p-value $<$ 0.001)

then it is very likely that Y_t (i.e. $Y_t \neq$ C) is also not a context. It is clear that the candidate contexts are not independent and there are strong dependency relationships between contiguous sentences in a forum. Therefore, a desirable model should be able to capture the dependency.

The context detection can be modeled as a classification problem. Traditional classification tools, e.g. SVM, can be employed, where each pair of question and candidate context will be treated as an instance. However, they cannot capture the dependency relationship between sentences.

To this end, we proposed a general framework to detect contexts and answers based on Conditional Random Fields (Lafferty et al., 2001) (CRFs) which are able to model the sequential dependencies between contiguous nodes. A CRF is an undirected graphical model G of the conditional distribution $P(\mathbf{Y}|\mathbf{X})$. $\mathbf{X}$ are the random variables over the labels of the nodes that are globally conditioned on $\mathbf{X}$, which are the random variables of the observations. (See Section 3.2.4 for more about CRFs)

Linear CRF model has been successfully applied in NLP and text mining tasks (McCallum and Li, 2003; Sha and Pereira, 2003). However, our problem cannot be modeled with Linear CRFs in the same way as other NLP tasks, where one node has a unique label. In our problem, each node (sentence) might have multiple labels since (1) one sentence could be the context of multiple questions in a thread or (2) it could be the context of one question but not the other. Thus, it is difficult to find a solution such that we can tag context sentences for all questions in a thread in single pass.

Here we assume that questions in a given thread are independent and are found, and then we can label a thread with m questions one-by-one in m-passes. In each pass, one question Q_i is selected as focus and each other sentence in the thread will be labeled as *context* C of Q_i or not using Linear CRF model. The graphical representations of Linear CRFs is shown in Figure2(a). The linear-chain edges can capture the dependency between two contiguous nodes. The observation sequence $\mathbf{x} = <\mathbf{x}_1, \mathbf{x}_2,...,\mathbf{x}_t>$, where t is the number of sentences in a thread, represents predictors (to be described in Section 3.2.5), and the tag sequence $\mathbf{y}=<y_1,...,y_t>$, where $y_i \in \{C, P\}$, determines whether a sentence is plain text P or context C of question Q_i.

Answer detection. Answers usually appear in the posts after the post containing the question. We observed that paragraph is usually a good segment for answer while the proposed approach is applicable to other kinds of segments. There are also strong dependencies between contiguous answer segments. Thus, position information and similarity method are not adequate for answer detection. To cope with the dependency between contiguous answer segments, we employ linear CRF models for answer detection.

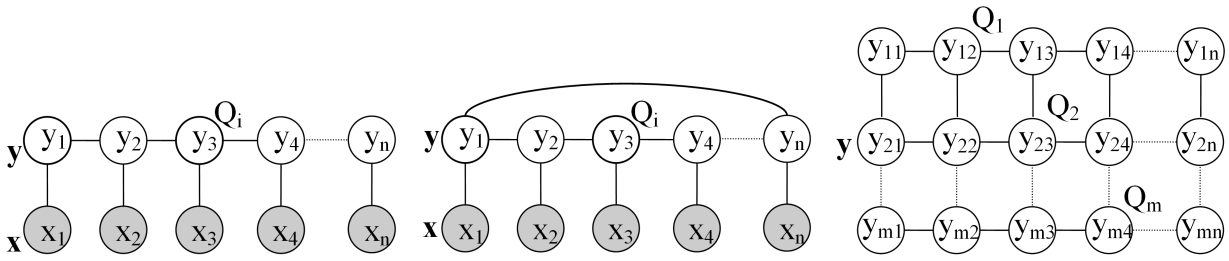
3.2.2 Leveraging context for answer detection using Skip-chain CRFs

We observed 74% questions lack contextual information in our corpus. As discussed in introduction, the constraints or background information provided by context are very useful to link question and answers. Therefore, contexts should be leveraged to detect answers. The linear CRF model can capture the dependency between contiguous sentences. However, it cannot capture the long distance dependency between contexts and answers.

One straightforward method of leveraging context is to detect contexts and answers in two phases, i.e. to first identify contexts, and then label answers using both the context and question information (e.g. the similarity between context and answer can be used as features in CRFs). The two-phase procedure, however, still cannot capture the non-local dependency between contexts and answers in a thread.

To model the long distance dependency between contexts and answers, we will use Skip-chain CRF model to detect context and answer together. Skip-chain CRF model is applied for entity extraction and meeting summarization (Sutton and McCallum, 2006; Galley, 2006). The graphical representation of a Skip-chain CRF given in Figure2(b) consists of two types of edges: linear-chain (y_t to y_{t-1}) and skip-chain edges (y_1 to y_n).

Ideally, the skip-chain edges will establish the



(a) Linear CRFs (b) Skip-chain CRFs (c) 2D CRFs

Figure 2: CRF Models

Skip-Chain	$y_v = A$	$y_v \neq A$
$y_u = C$	3,504	6,822
$y_u \neq C$	1,255	7,464

Table 2: Contingence table(χ^2=963,p-value < 0.001)

connection between candidate pairs with high probability of being context and answer of a question. To introduce skip-chain edges between any pairs of non-contiguous sentences will be computationally expensive for Skip-chain CRFs, and also introduce noise. To make the cardinality and number of cliques in the graph manageable and also eliminate noisy edges, we would like to generate edges only for sentence pairs with high possibility of being context and answer. We explain how do we achieve this next. Given a question Q_i in post P_j of a thread with n posts, its contexts usually occur within post P_j or before P_j while answers appear in the posts after P_j. In this paper, we will establish an edge between each candidate answer v and one candidate context in $\{P_k\}_{k=1}^{j}$ such that they have the highest possibility of being a context-answer pair of question Q_i. We use the product of $sim(x_u, Q_i)$ and $sim(x_v, \{x_u, Q_i\})$ to estimate the possibility of being a context-answer pair for (u, v).

$$\underset{u \in \{P_k\}_{k=1}^{j}}{\text{argmax}} \; sim(x_u, Q_i).sim(x_v, \{x_u, Q_i\}) \quad (1)$$

Table 2 shows that y_u and y_v in the skip chain generated by our heuristics influence each other.

Skip-chain CRFs improve the performance of answer detection due to the introduced skip-chain edges that represent the joint probability conditioned on the question, which is exploited by skip-chain feature function: $f(y_u, y_v, Q_i, \mathbf{x})$.

3.2.3 Using 2D CRF Model

Both Linear CRFs and Skip-chain CRFs label the contexts and answers for each question in separate passes by assuming that questions in a thread are independent. Actually the assumption does not hold in many cases. Let us look at an example. As in Figure 1, Sentence S10 is an answer for both question 2 and question 3. S10 could be recognized as the answer of question 2 due to the shared word *traffic*, but there is no direct relation between question 3 and S10. To label S10, we need consider the dependency relation between question 2 and 3. In other words, the question-answer relation between question 3 and S10 can be captured by a joint modeling of the dependency among S10, question 2 and question 3. The labels of the same sentence for two contiguous questions in a thread would be conditioned on the dependency relationship between the questions. Such a dependency cannot be captured by both Linear CRFs and Skip-chain CRFs.

To capture the dependency between the contiguous questions, we employ 2D CRFs to help context and answer detection. 2D CRF model is used in (Zhu et al., 2005) to model the neighborhood dependency in blocks within a web page. As shown in Figure2(c), 2D CRF models the labeling task for all questions in a thread. The ith row in a grid corresponds to one pass of Linear CRF model (or Skip-chain model) which labels contexts and answers for question Q_i. The vertical edges in the figure represent the joint probability conditioned on the contiguous questions, which will be exploited by 2D feature function: $f(y_{i,j}, y_{i+1,j}, Q_i, Q_{i+1}, \mathbf{x})$. Thus, the information generated in single CRF chain could be propagated over the whole grid. In this way, context and answer detection for all questions in the thread could be modeled together.

3.2.4 Conditional Random Fields (CRFs)

The Linear, Skip-Chain and 2D CRFs can be generalized as pairwise CRFs, which have two kinds of cliques in graph G: 1) node y_t and 2) edge (y_u, y_v). The joint probability is defined as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\Big\{\sum_{k,t} \lambda_k f_k(y_t, \mathbf{x}) + \sum_{k,t} \mu_k g_k(y_u, y_v, \mathbf{x})\Big\},$$

where $Z(\mathbf{x})$ is the normalization factor, f_k is the feature on nodes, g_k is on edges between u and v, and λ_k and μ_k are parameters.

Linear CRFs are based on the first order Markov assumption that the contiguous nodes are dependent. The pairwise edges in Skip-chain CRFs represent the long distance dependency between the skipped nodes, while the ones in 2D CRFs represent the dependency between the horizontal nodes.

Inference and Parameter Estimation. For linear CRFs, dynamic programming is used to compute the *maximum a posteriori* (MAP) of $\mathbf{y}$ given $\mathbf{x}$. However, for more complicated graphs with cycles, exact inference needs the junction tree representation of the original graph and the algorithm is exponential to the treewidth. For fast inference, loopy Belief Propagation (Pearl, 1988) is implemented.

Given the training Data $D = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{n}$, the parameter estimation is to determine the parameters based on maximizing the log-likelihood $L_\lambda = \sum_{i=1}^{n} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$. In linear CRF model, dynamic programming and L-BFGS [1] can be used to optimize objective function L_λ, while for complicated CRFs, Loopy BP are used instead to calculate the marginal probability.

3.2.5 Features used in CRF models

The main features used in linear CRF models for context detection are listed in Table 3. We next briefly introduce them.

The similarity feature is to capture the words similarity and semantic similarity between candidate contexts and answers. The semantic similarity is computed based on Wu and Palmer's measure (Wu and Palmer, 1994) using WordNet (Fellbaum, 1998). The similarity between contiguous sentences will be used to capture the dependency for CRFs. In addition, to bridge the lexical gaps between question and context, we learned top-3 context terms for each question term from 300,000 question-description pairs obtained from Yahoo! Answers using mutual information (Berger et al., 2000) (question description in Yahoo! Answers is comparable to contexts in forums), and then use them to expand question and compute cosine similarity.

Similarity features:
· Cosine similarity with the question
· Similarity with the question using WordNet
· Cosine similarity between contiguous sentences
· Similarity between contiguous sentences using WordNet
· Cosine similarity with the expanded question using the lexical matching words
Structural features:
· The relative position to current question
· Is the author is the same with that of the question.
Discourse and lexical features:
· The number of Pronouns in the question
· The presence of fillers, fluency devices (e.g. "uh", "ok")
· The presence of acknowledgment tokens
· The number of non-stopwords
· Whether the question has a noun or not
· Whether the question has a verb or not

Table 3: Features for linear CRFs. Unless otherwise mentioned, we refer to features of the sentence whose label to be predicted

The structural features of forums provide strong clues for contexts. For example, contexts of a question usually occur in the post containing the question or preceding posts.

We extracted the discourse features from a question, such as the number of pronouns in the question. A more useful feature would be to find the entity in surrounding sentences referred by a pronoun. We tried GATE (Cunningham et al., 2002) for anaphora resolution of the pronouns in questions, but the performance became worse with the feature, which is probably due to the difficulty of anaphora resolution in forum discourse. We also observed that questions often need context if the question do not contain a noun or a verb.

For answer detection, we use slightly different features and will not discuss them here due to space limitation. In addition, we need similarity features between skip-chain sentences for Skip-chain CRFs and similarity features between questions for 2D CRFs.

4 Experiments

4.1 Experimental setup

Corpus. We obtained about 1 million threads from TripAdvisor [2] forum and randomly selected 591 fo-

[1] L-BFGS stands for limited memory Broyden-Fletcher-Goldfarb-Shanno.

[2] http://www.tripadvisor.com/ForumHome, one of the most popular travel forums

Feature	Prec(%)	Rec(%)	F_1(%)
5W-1H words	69.98	14.95	24.63
Question Mark	**91.25**	69.85	79.12
RIPPER	88.84	75.81	81.76
Our	88.75	**87.03**	**87.85**

Table 4: Performance of Question Detection

rum threads as our corpus. Each thread in our corpus contains at least two posts and on average each thread consists of 4.46 posts. Two annotators were asked to tag questions, their contexts, and answers in each thread. The kappa statistic for identifying question is 0.96, for linking context and question given a question is 0.75, and for linking answer and question given a question is 0.69. We conducted experiments on both the union and intersection of the two annotated data. The experimental results on both data are qualitatively comparable. We only report results on union data due to space limitation. The union data contains 2,041 questions, 2,479 contexts and 3,441 answers.

Metrics. We calculated `precision`, `recall`, and F_1`-score` for all tasks. All the experimental results are obtained through the average of 5 trials of 5-fold cross validation.

4.2 Experimental results

Question Detection. The experiment is to evaluate the performance of our question detection method against simple rules and the method used in (Shrestha and McKeown, 2004), denoted as `RIPPER`. The results are given in Table 5. The first two rows show the results of simple rules. The rule `5W-1H words` is that a sentence is a question if it begins with 5W-1H words; The rule `Question Mark` is that a sentence is a question if it ends with question mark. Although `Question Mark` achieves the best precision, its recall is low. Our method outperforms the simple rules in terms of F_1`-score`. Our method differs from `RIPPER` in that we adopt SVM model while RIPPER was used in (Shrestha and McKeown, 2004).

Linear CRFs for Context and Answer Detection. This experiment is to evaluate Linear CRF model (Section 3.2.1) for context and answer detection by comparing with SVM and C4.5(Quinlan, 1993). For SVM, we use SVM^{light}(Joachims, 1999) and report the best SVM result when using linear or polynomial kernels. For context detection, SVM and C4.5 use the same set of features. For answer detection, for SVM and C4.5 we add the similarity between real context and answer as extra features; otherwise, they failed. As shown in Table 5, Linear CRF model outperforms SVM and C4.5 for both context and answer detection, even if Linear CRF did not use any context information for answer finding. The main reason for the improvement is that CRF models can capture the sequential dependency between segments in forums as discussed in Section 3.2.1.

Model	Prec(%)	Rec(%)	F_1(%)
Context Detection			
SVM	61.76	58.89	60.27
C4.5	60.09	54.13	56.95
Linear CRF	**63.25**	**69.17**	**66.07**
Answer Detection			
SVM	61.36	46.81	53.31
C4.5	68.36	40.55	50.90
Linear CRF	**78.85**	**49.37**	**59.76**

Table 5: Context and Answer Detection

position	Prec(%)	Rec(%)	F_1(%)
Context Detection			
Previous One	37.33	20.48	26.45
Previous All	33.66	79.40	47.28
Anwer Detection			
Following One	59.34	21.44	31.50
Following All	26.71	100	42.12

Table 6: Using position information for detection

We next report a baseline of context detection using previous sentences in the same post with its question since contexts often occur in the question post or preceding posts. Similarly, we report a baseline of answer detecting using following segments of a question as answers. The results given in Table 6 show that location information is far from adequate to detect contexts and answers.

The usefulness of contexts. This experiment is to evaluate the usefulness of contexts in answer detection, by adding the similarity between the context (obtained with different methods) and candidate answer as an extra feature for CRFs. Table 7 shows the impact of context on answer detection using Linear CRFs. L-CRF+context uses the context found by Linear CRFs, and performs better than Linear CRF without context. We also found that the performance

Model	Prec(%)	Rec(%)	F_1(%)
No context	78.85	49.37	59.76
L-CRF+context	79.56	55.81	64.64
Prev. sentence	79.48	51.73	61.71
Real context	80.09	57.36	65.88

Table 7: Contextual Information for Answer Detection

Model	Prec(%)	Rec(%)	F_1(%)
L-CRF+context	79.56	55.81	64.64
Skip-chain	78.40	67.74	72.38
2D	76.95	65.66	70.61
2D+Skip-chain	81.81	69.76	75.31

Table 8: Skip-chain and 2D CRFs for answer detection

of L-CRF+context is close to that using real context, while it is better than CRFs using the previous sentence as context. The results clearly shows that contextual information greatly improves the performance of answer detection. This was also observed for other classification methods in our experiments: SVM and C4.5 (in Table 5) failed if we did not use context.

Improved Models. This experiment is to evaluate the effectiveness of Skip-Chain CRFs (Section 3.2.2) and 2D CRFs (Section 3.2.3) for our tasks. The results are given in Table 8. As expected, Skip-chain CRFs outperform L-CRF+context since Skip-chain CRFs can model the inter-dependency between contexts and answers while in L-CRF+context the context can only be reflected by the features on the observations. We also observed that 2D CRFs improves the performance of L-CRF+context and we achieved the best performance if we combine the 2D CRFs and Skip-chain CRFs. For context detection, there is slightly improvement, e.g. `Precision` (64.48%) `Recall` (71.51%) and F_1`-score` (67.79%).

Evaluating Features. We also evaluated the contributions of each category of features in Table 3 to context detection. We found that similarity features are the most important and structural feature the next. We also observed the same trend for answer detection. We omit the details here due to space limitation.

As a summary, 1) our question detection method is indeed effective; 2) our CRF model outperforms SVM and C4.5 for both context and answer detections; 3) context is very useful in answer detection; 4) the Skip-chain CRF method is effective in leveraging context for answer detection; and 5) the combination of 2D CRFs and Skip-chain CRF achieves the best performance.

5 Discussions and Conclusions

We presented a new approach to detecting question-context-answer triples in forums with good performance. We next discuss our experience not covered by the experiments, and future work.

We found our methods often cannot identify questions expressed by imperative sentences in question detection task, e.g. "*recommend a restaurant in New York*". This would call for future work. We also observed that factoid questions, one of focuses in the TREC QA community, take less than 10% question in our corpus. It would be interesting to revisit QA techniques to process forum data.

Since contexts of questions are largely unexplored in previous work, we analyze the contexts in our corpus and classify them into three categories: 1) context contains the main content of question while question contains no constraint, e.g. "*i will visit NY at Oct, looking for a cheap hotel but convenient. Any good suggestion?* "; 2) contexts explain or clarify part of the question, such as a definite noun phrase, e.g. '*We are going on the Taste of Paris. Does anyone know if it is advisable to take a suitcase with us on the tour.*, where the first sentence is to describe *the tour*; and 3) contexts provide constraint or background for question that is syntactically complete, e.g. "*We are interested in visiting the Great Wall(and flying from London). Can anyone recommend a tour operator.*" In our corpus, about 26% questions do not need context, 12% questions need Type 1 context, 32% need Type 2 context and 30% Type 3. We found that our techniques often do not perform well on Type 3 questions.

Other future work includes: 1) to summarize multiple threads using the triples extracted from individual threads. This could be done by clustering question-context-answer triples; 2) to use the traditional text summarization techniques to summarize the multiple answer segments; 3) to integrate the Question Answering techniques as features of our framework to further improve answer finding; and 4) to reformulate questions using its context to generate more user-friendly questions for CQA services.

References

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of SIGIR*.

J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski16, E. Voorhees, and R. Weishedel. 2006. Issues, tasks and program structures to roadmap research in question and answering (qna). ARAD: Advanced Research and Development Activity (US).

G. Carenini, R. Ng, and X. Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of WWW*.

H. Cui, R. Sun, K. Li, M. Kan, and T. Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of SIGIR*.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of ACL*.

H. Dang, J. Lin, and D. Kelly. 2007. Overview of the trec 2007 question answering track. In *Proceedings of TREC*.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May.

D. Feng, E. Shaw, J. Kim, and E. Hovy. 2006a. An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of IUI*.

D. Feng, E. Shaw, J. Kim, and E. Hovy. 2006b. Learning to detect conversation focus of threaded discussions. In *Proceedings of HLT-NAACL*.

M. Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of EMNLP*.

S. Harabagiu and A. Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of ACL*.

J. Huang, M. Zhou, and D. Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *Proceedings of IJCAI*.

J. Jeon, W. Croft, and J. Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of CIKM*.

T. Joachims. 1999. Making large-scale support vector machine learning practical. MIT Press, Cambridge, MA, USA.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

D. Lam, S. Rohall, C. Schmandt, and M. Stern. 2002. Exploiting e-mail structure to improve summarization.

W. Lehnert. 1977. *The process of question answering.* Ph.D. thesis, New Haven, CT, USA.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-2003*.

A. Nenkova and A. Bagga. 2003. Facilitating email thread access by extractive summary generation. In *Proceedings of RANLP*.

P. Newman and J. Blitzer. 2003. Summarizing archived discussions: A beginning. In *Proceedings of IUI*.

J. Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

J. Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

O. Rambow, L. Shrestha, J. Chen, and C. Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL*.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *HLT-NAACL*.

L. Shrestha and K. McKeown. 2004. Detection of question-answer pairs in email conversations. In *Proceedings of COLING*.

C. Sutton and A. McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press. To appear.

S. Wan and K. McKeown. 2004. Generating overview summaries of ongoing email thread discussions. In *Proceedings of COLING*.

Z. Wu and M. S. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of ACL*.

F. Yang, J. Feng, and G. Fabbrizio. 2006. A data driven approach to relevancy recognition for contextual question answering. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*.

L. Zhou and E. Hovy. 2005. Digesting virtual "geek" culture: The summarization of technical internet relay chats. In *Proceedings of ACL*.

J. Zhu, Z. Nie, J. Wen, B. Zhang, and W. Ma. 2005. 2d conditional random fields for web information extraction. In *Proceedings of ICML*.