FINDING QUESTION-ANSWER PAIRS FROM ONLINE FORUMS

Erkan SOLMAZ
Introduction

Related Work

Algorithms for Question Detection

Algorithms for Answer Detection

Experiments

Conclusions
INTRODUCTION
What’s an online forum?
- A web application
- It contains valuable user generated content
- Desire to extract and reuse this content

In this paper, we only consider question-answer (QA) pairs from forums.

Why mining QA pairs from forums?
- Essential to many QA services (Yahoo! Answers, Baidu)
- Questions are the focus of forum discussions: Forum Management
- Useful for Knowledge Base of Chatbot
A forum thread

- Contains an initiating post and a couple of reply posts.
  - Initiating post -> several questions
  - Reply posts -> answers to those questions

- Has an asynchronous nature: multiple participants to pursue multiple questions in parallel

We propose a new method to detect QA pairs

- Question detection
- Answer detection
**Question Detection**

- **Objective**: Detect all questions in a thread -> Easy?
- They are often stated in an informal way -> checking (?) and 5W1H words is not enough

**Answer Detection**

- Not easy because possible multiple QA pairs in a thread
- One question may have multiple replies. Which one?
- Query-document model doesn’t work due to forum-specific features such as distance between QA.

**Solution**: Graph-based approach
RELATED WORK
No successful works on extracting QA pairs from forum data

Closest works:
- input-reply pairs for chatbot knowledge
- QA pairs for email summarization
- Researches done in TREC on short answers for factoid questions
- Research FAQ and CQA retrieval
Algorithms For Question Detection
5W1H words and (?) are not adequate

- 30% questions w/o question marks,
- 9% sentences ending with question marks are not question

Solution:

- Extract labeled sequential patterns (LSPs) from both questions and non-questions to characterize them
- Use the discovered patterns as features to build classifiers for question detection
A LSP, \( p \), is an implication in the form of \( \text{LHS} \rightarrow c \), where LHS is a sequence and \( c \) is a class label.

A LSP \( p_1 \) is contained by \( p_2 \) if

- the sequence \( p_1.LHS \) is contained by \( p_2.LHS \)
- \( p_1.c = p_2.c \)

The support of \( p \), \( \text{sup}(p) \), is the percentage of tuples in database D that contain the LSP \( p \).

The probability of the LSP \( p \) being true: “the confidence of \( p \)”

\[
\text{conf}(p) = \frac{\text{sup}(p)}{\text{sup}(p.LHS)}
\]

, where \( \text{sup}(p) \) is the support of \( p \).
Preprocess each sentence by applying Part-Of-Speech (POS) tagger MXPOST Toolkit to tag each sentence

- e.g. “where can you find a job” becomes “where can PRP VB DT NN"

Each processed sentence becomes a database tuple

The combination of POS tags and keywords allows us to capture representative features for question sentences by mining LSPs

- <anyone, VB, how> → Q
- <what, do, PRP, VB> → Q
We will mine LPSs by imposing thresholds

- **Minimum support** to ensure that discovered patterns are general
- **Minimum confidence** to ensure that all discovered LSPs are discriminating

In our experiments, imposed values are 0.5% and 85% respectively

Each discovered LSP forms a binary feature as the input for a classification model

If a sentence includes a LSP, the corresponding feature is set at 1

We build a classifier to detect questions using Ripper classification algorithm
Algorithms For Answer Detection
Three IR methods to rank candidate answers for a given forum question

- Cosine Similarity
- Query likelihood language model
- KL-divergence language model

Classification based re-ranking

- First method: Classifiers are built to extract input-response pairs using content features (e.g. the number overlapping words between input and reply post) and structural features (e.g. is the reply posted by the thread starter)

- Second method: We can treat each question and candidate answer pair as an instance, compute features for the pair, and train a classifier. The resulting classification scores can be used to rank the candidate answers of a question
Graph Based Propagation Method

- It considers the inter-relationships of candidate answers of which previous methods do not make use.
- It’s been successful in Web search where links are usually obvious.
- Idea behind it:
  - If a candidate answer is related to an authoritative candidate answer with high score, the candidate answer, which may not have a high score, is also likely to be an answer.
We build a weighted directed graph given a question $q$ and its candidate answer set $A_q$.

- The problem: how to generate the edge set $E$.

Use KL-divergence language model to determine the edge between two candidates answers $a_o$ and $a_g$.

- KL-divergence captures the asymmetry (opposite in cosine similarity).

An edged is formed from $a_o$ (offspring) to $a_g$ (generator) if the value $\frac{1}{1 + KL(a_o|a_g)}$ is larger than a threshold $\theta$ (determined empirically).

- Our methods are not sensitive to $\theta$ because we allow self-loop i.e. each candidate answer can be its own generator.
Straightforward way: use KL-divergence score. We consider two more factors in computing weight to achieve better performance:

- Distance between question and answer denoted by $d(q, a)$
- Authority level of authors denoted by $\text{author}(i)$

The weight for edge $a_o \rightarrow a_g$

$$w(a_o \rightarrow a_g) = \frac{1}{1 + \text{KL}(P(a_o)\|P(a_g))} + \lambda_1 \frac{1}{d(a_g, q)} + \lambda_2 \text{author}(a_g)$$

- Normalized

$$nw(a_o \rightarrow a_g) = \lambda \frac{1}{|A_q|} + (1 - \lambda) \frac{w(a_o \rightarrow a_g)}{\sum_{g \in G_{a_o}} w(a_o \rightarrow g)}$$
Propagation w/o initial scores

The product of the authority value and the initial ranking score (from previous slide) between candidate answer a and question q will be returned as the final ranking score for a.

\[ Pr(q|a) := authority(a) \times score(q, a) \]

How to compute authority score

\[ authority(a_g) = \sum_{a_o \in C_a} nw(a_o \rightarrow a_g) \times authority(a_o) \]

Propagation with initial scores

\[ Pr(q|a) = \lambda \frac{Pr(q|a)}{\sum_{t \in C_q} Pr(q|t)} + (1-\lambda) \sum_{v \in C_q} nw(v \rightarrow a) \times Pr(q|v) \]

, where \( \lambda \) determined empirically is a trade-off between a and its offsprings
When training data is available, the graph-based model can be integrated with classification model in two ways:

- For each candidate QA pair, the results of graph-based methods can be added to classification method to determine if the candidate answer is an answer of the question (ranking).
- The classification score returned by a classifier is often (or can be transformed into) the probability for a candidate answer being a true answer and can be used as initial score for propagation of graph-based model.
Experiments
SOURCE DATA

Data is selected from 3 forums of different scales
- TripAdvisor -> 1,212,153 threads
- LonelyPlanet -> 86,772 threads
- BootsnAll Network -> 25,298 threads

Each thread in our corpus contains at least 2 posts (4.46 on avg)

2 annotators tag questions and their answers in each thread with kappa statistic 0.96

Two datasets are generated from the union of the two annotated data for question detection (Q-TUnion and Q-TInter)

Five datasets for answer detection
**Evaluation on Question Detection**

<table>
<thead>
<tr>
<th>Data</th>
<th>Method</th>
<th>Prec(%)</th>
<th>Rec(%)</th>
<th>F₁(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-TUnion</td>
<td>5W-1H words</td>
<td>69.0</td>
<td>14.8</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>Question Mark</td>
<td>96.8</td>
<td>78.4</td>
<td>86.6</td>
</tr>
<tr>
<td></td>
<td>SM [18]</td>
<td>81.9</td>
<td>87.8</td>
<td>84.6</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>96.5</td>
<td>98.5</td>
<td>97.5</td>
</tr>
<tr>
<td>Q-TInter</td>
<td>5W-1H words</td>
<td>69.0</td>
<td>15.3</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td>Question Mark</td>
<td>98.7</td>
<td>77.6</td>
<td>86.9</td>
</tr>
<tr>
<td></td>
<td>SM [18]</td>
<td>92.7</td>
<td>86.8</td>
<td>89.7</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>97.8</td>
<td>97.0</td>
<td>97.4</td>
</tr>
</tbody>
</table>

- The result are obtained through 10-fold cross-validation
- SM is simple rules and the method in a previous related work.
## Evaluation on Answer Identification Cont’d

<table>
<thead>
<tr>
<th>Method</th>
<th>Abbrev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Answer/Random Guess</td>
<td>NA</td>
</tr>
<tr>
<td>LexRank [15]</td>
<td>Lex</td>
</tr>
<tr>
<td>Classification [7, 23] (Section 4.1)</td>
<td>Cla</td>
</tr>
<tr>
<td>Cosine similarity (Sec. 4.1)</td>
<td>CS</td>
</tr>
<tr>
<td>Query Likelihood language model (Sec. 4.1)</td>
<td>QL</td>
</tr>
<tr>
<td>KL divergence language model (Sec. 4.1)</td>
<td>KL</td>
</tr>
<tr>
<td>Graph+Cosine similarity (Sec. 4.2)</td>
<td>G+CS</td>
</tr>
<tr>
<td>Graph+Query Likelihood language model (Sec. 4.2)</td>
<td>G+QL</td>
</tr>
<tr>
<td>Graph+KL divergence language model (Sec. 4.2)</td>
<td>G+KL</td>
</tr>
<tr>
<td>Graph(Classification) (Sec. 4.3)</td>
<td>G(Cla)</td>
</tr>
<tr>
<td>Classification(Graph) (Sec. 4.3)</td>
<td>Cla(G)</td>
</tr>
</tbody>
</table>

The methods and their abbreviations
Evaluation on Question Detection

<table>
<thead>
<tr>
<th>Method</th>
<th>All questions</th>
<th>Question with answer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@1(#)</td>
<td>MRR</td>
</tr>
<tr>
<td>NA</td>
<td>0.525(806)</td>
<td>0.585</td>
</tr>
<tr>
<td>Lex</td>
<td>0.529(812)</td>
<td>0.616</td>
</tr>
<tr>
<td>Cla</td>
<td>0.588(903)</td>
<td>0.667</td>
</tr>
<tr>
<td>CS</td>
<td>0.559(858)</td>
<td>0.643</td>
</tr>
<tr>
<td>QL</td>
<td>0.568(872)</td>
<td>0.644</td>
</tr>
<tr>
<td>KL</td>
<td>0.578(887)</td>
<td>0.659</td>
</tr>
<tr>
<td>G+CS</td>
<td>0.603(925)</td>
<td>0.677</td>
</tr>
<tr>
<td>G+QL</td>
<td>0.620(952)</td>
<td>0.687</td>
</tr>
<tr>
<td>G+KL</td>
<td><strong>0.665(1,021)</strong></td>
<td><strong>0.719</strong></td>
</tr>
</tbody>
</table>

Results on A-TUnion data
- Graph-based methods significantly outperform their respective counterparts in terms of all the three measures as expected
- The overall performance deteriorated due to the questions without answers
The extracted QA pairs could be used to enrich the knowledge base of community based QA service, instant answer service and Chatbot.

Experimental results on real data show that our approach is effective.

Future work:
- Detecting questions without answers
- Revisiting more TREC QA techniques to see if they can help answer detection in forums
- Modelling the relationship of questions in the same thread to improve answer detection
- Investigating the effectiveness of our techniques in passage retrieval on TREC QA data