Question Pre-Processing in a QA System on Internet Discussion Groups

Chuan-Jie Lin and Chun-Hung Cho
Department of Computer Science and Engineering
National Taiwan Ocean University
No 2, Rd Pei-Ning, Keelung 202, Taiwan, R.O.C
cjlin@mail.ntou.edu.tw; futurehero@seed.net.tw

Abstract

This paper proposes methods to pre-process questions in the postings before a QA system can find answers in a discussion group in the Internet. Pre-processing includes garbage text removal and question segmentation. Garbage keywords are collected and different length thresholds are assigned to them for garbage text identification. Interrogative forms and question types are used to segment questions. The best performance on the test set achieves 92.57% accuracy in garbage text removal and 85.87% accuracy in question segmentation, respectively.

1 Introduction

Question answering has been a hot research topic in recent years. Large scale QA evaluation projects (e.g. TREC QA-Track[1], QA@CLEF[2], and NTCIR[3] QAC and CLQA Tracks) are helpful to the developments of question answering.

However, real automatic QA services are not ready in the Internet. One popular way for Internet users to ask questions and get answers is to visit discussion groups, such as Usenet newsgroups[4] or Yahoo! Answers[5]. Each discussion group focuses on one topic so that users can easily find one to post their questions.

There are two ways a user can try to find answers. You can post your question in a related discussion group and wait for other users to provide answers. Some discussion groups provide search toolbars so that you can search your question first to see if there are similar postings asking the same question. In Yahoo! Answers, you can also judge answers offered by other users and mark the best one.

Postings in discussion groups are good materials to develop a FAQ-style QA system in the Internet. By finding questions in the discussion groups similar to a new posting, responses to these questions can provide answers or relevant information.

But without pre-processing, measuring similarity with original texts will arise some problems:

1. Some phrases such as “many thanks” or “help me please” are not part of a question. These kinds of phrases will introduce noise and harm matching performance.

2. Quite often there is more than one question in one posting. If the question which is most similar to the user's question appears in an existed posting together with other different questions, it will get a lower similarity score than the one it is supposed to have because of other questions.

Therefore, inappropriate phrases should be removed and different questions in one posting should be separated before question comparison.

There is no research focusing on this topic. FAQ finders (Lai et al., 2002; Lytinen and Tomuro, 2002; Burke, 1997) are closely related to this topic. However, there are differences between them. First of all, questions in a FAQ set are often written in perfect grammar without

[1] http://trec.nist.gov/data/qa.html
[2] http://clef-qa.itc.it/
[3] http://research.nii.ac.jp/ntcir/index-en.html
[4] Now they can be accessed via Google Groups: http://groups.google.com/
[5] http://answers.yahoo.com/

Proceedings of the Workshop on Task-Focused Summarization and Question Answering, pages 16–23,
Sydney, July 2006. ©2006 Association for Computational Linguistics

garbage text. Second, questions are often paired with answers separately. I.e. there is often one question in one QA pair.

There were some research groups who divided questions into segments. Soricut and Brill (2004) chunked questions and used them as queries to search engines. Saquete et al. (2004) focused on decomposition of a complex question into several sub-questions. In this paper, question segmentation is to identify different questions posed in one posting.

2 Garbage Text Removal

2.1 Garbage Texts

Articles in discussion groups are colloquial. Users often write articles as if they are talking to other users. For this reason, phrases expressing appreciation, begging, or emotions of writers are often seen in the postings. For example:

> 有關 powerpoint 問題 我想請問一下$_1$ 該如何把 access 的整個視窗放到簡報上撥放謝謝$_2$
> (About Powerpoint, I'd like to ask$_1$, how to put the whole window seen in Access onto a slide? Thank you$_2$!)

The phrases "我想請問一下" ("I'd like to ask") and "謝謝" ("Thank you") are unimportant to the question itself.

These phrases often contain content words, not stop words, and thus are hard to be distinguished with the real questions. If these phrases are not removed, it can happen that two questions are judged "similar" because one of these phrases appears in both questions.

A phrase which contributes no information about a question is called ***garbage text*** in this paper and should be removed beforehand in order to reduce noise. The term ***theme text*** is used to refer to the remaining text.

After examining real querying postings, some characteristics of garbage texts are observed:

1. Some words strongly suggest themselves being in a garbage text, such as "thank" in "thank you so much", or "help" in "who can help me".

2. Some words appear in both theme texts and garbage texts, hence ambiguity arises. For example:

   "請教高手" (Any expert please help)
   "快閃高手" (Flash Expert)

   The first phrase is a garbage text, while the second phrase is a product name. The word "expert" suggests an existence of a garbage text but not in all cases.

Because punctuation marks are not reliable in Chinese, we use sentence fragment as the unit to be processed. A ***sentence fragment*** is defined to be a fragment of text segmented by commas, periods, question marks, exclamation marks, or space marks. A space mark can be a boundary of a sentence fragment only when both characters preceding and following the space mark are not the English letters, digits, or punctuation marks.

2.2 Strategies to Remove Garbage Texts

Frequent terms seen in garbage texts are collected as ***garbage keywords*** and grouped into classes according to their meanings and usages. Table 1 gives some examples of classes of garbage keywords collected from the training set.

Class	Garbage Keywords
Please	請問一下, 煩請,不好意思…
Thanks	感謝,謝謝,感恩,感溫…
Help	賜教,請教,幫我解答,救我…
Urgent	緊急,緊迫,急迫,急…

Table 1. Some Classes of Garbage Keywords

To handle ambiguity, this paper proposes a length information strategy to determine garbage texts as follows:

If a sentence fragment contains a garbage keyword and the length of the fragment after removing the garbage keyword is less than a threshold, the whole fragment will be judged as a garbage text. Otherwise, only the garbage keyword itself is judged as garbage text if it is never in an ambiguous case.

Different length thresholds are assigned to different classes of garbage keywords. If more than one garbage keyword occurring in a fragment, discard all the keywords first, and then compare the length of the remaining fragment with the maximal threshold among the ones corresponding to these garbage keywords.

In order to increase the coverage of garbage keywords, other linguistic resources are used to expand the list of garbage keywords. Synonyms in Tongyici Cilin (同義詞詞林), a thesaurus of Chinese words, are added into the list. More garbage keywords are added by common knowledge.

3 Question Segmentation

When a user posts an article in a discussion group, he may pose more than one question at one time. For example, in the following posting:

> Office 2003 和 XP←有何不同之處呢? 哪一個比較新呢？最新的版本是???????????
>
> (Office 2003 and XP ← What are the differences between them? Which version is newer? What is the latest version???????????)

there are 3 questions submitted at a time. If a new user wants to know the latest version of Office, responses to the previous posting will give answers.

Table 2 lists the statistics of number of questions in the training set. The first column is the number of questions in one posting. The second and the third columns are the number and the percentage of postings which contain such number of questions, respectively.

Q#	Post#	Perc (%)
1	494	56.98
2	259	29.87
3	82	9.46
4	22	2.54
5	4	0.46
≥ 6	6	0.69
≥ 2	**373**	**43.02**
Total	867	100.00

Table 2. Statistics of Number of Questions in Postings

As we can see in Table 2, nearly half (43.02%) of the postings contain two or more questions. That is why question segmentation is necessary.

3.1 Characteristics of Questions in a Posting

Several characteristics of question texts in postings were found in real discussion groups:

1. Some people use ‘?’ (question mark) at the end of a question while some people do not. In Chinese, some people even separate sentences only by spaces instead of punctuation marks. (Note that there is no space mark between words in Chinese text.)

2. Questions are usually in interrogative form. Either interrogatives or question marks appear in the questions.

3. One question may occur repeatedly in the same posting. It is often the case that a question appears both in the title and in the content. Sometimes a user repeats a sentence several times to show his anxiety.

4. One question may be expressed in different ways in the same posting. The sentences may be similar. For example:

   A: Office2000 的剪貼簿只能維持12個項目?

   B: Office2000 的剪貼簿只能保持12個項目?

   (Can the clipboard of Office2000 only keep 12 items?)

   “維持” and “保持” are synonyms in the meaning of “keep”.

   Dissimilar sentences may also refer to the same question. For example,

   (1) How to use automatic text wrapping in Excel?
   (2) If I want to put two or more lines in one cell, what can I do?
   (3) How to use it?

   These three sentences ask the same question: “How to use automatic text wrapping in Excel?” The second sentence makes a detailed description of what he wants to do. Topic of the third sentence is the same as the first sentence hence is omitted. Topic ellipsis is quite often seen in Chinese.

5. Some users will give examples to explain the questions. These sentences often start with phrases like “for example” or “such as”.

3.2 Strategies to Separate Questions

According to the observations in Section 3.1, several strategies are proposed to separate questions:

(1) Separating by Question Mark (‘?’)

It is the simplest method. We use it as a baseline strategy.

(2) Identifying Questions by Interrogative Forms

Questions are usually in ***interrogative forms*** including subject inversion (“is he…”, “does it…”), using interrogatives (“who is…”), or a declarative sentence attached with a question mark (“Office2000 is better?”). Only the third form requires a question mark. The first two forms can specify themselves as questions by text only. Moreover, there are particles in Chinese indicating a question as well, such as “嗎” or “呢”.

If a sentence fragment is in interrogative form, it will be judged as a question and separated from the others. A fragment not in interrogative form is merged with the nearest question fragment preceding it (or following it if no preceding one). Note that garbage texts have been removed before question separation.

(3) Merging or Removing Similar Sentences

If two sentence fragments are exactly the same, one of them will be removed. If two sentence fragments are similar, they are merged into one question fragment.

Similarity is measured by the Dice coefficient (Dice, 1945) using weights of common words in the two sentence fragments. The similarity of two sentence fragments *X* and *Y* is defined as follows:

$$Sim(X,Y) = \frac{2 \times \sum_{k \in X \cap Y} Wt(k)}{\sum_{w \in X} Wt(w) + \sum_{t \in Y} Wt(t)} \quad (1)$$

where *Wt*(*w*) is the weight of a word *w*. In Equation 1, *k* is one of the words appearing in both *X* and *Y*. Fragments with similarity higher than a threshold are merged together.

The weight of a word is designed as the weight of its part-of-speech as listed in Table 3. Nouns and verbs have higher weights, while adverbs and particles have lower weights. Note that foreign words are assigned a rather high weight, because names of software products such as “Office” or “Oracle” are often written in English, which are foreign words with respect to Chinese.

POS	Weight
Vt (Transitive Verb), FW (Foreign Word)	100
N (Noun)	90
Vi (Intransitive Verb)	80
A (Adjective)	40
ADV (Adverb), ASP (Tense), C (Connective), DET (Determiner), P (Preposition), T (Particle)	0

Table 3. Weights of Part-of-Speeches

Before computing similarity, word segmentation is performed to identify words in Chinese text. After that, a part-of-speech tagger is used to obtain POS information of each word.

(4) Merging Questions with the Same Type

The information of question type has been widely adopted in QA systems (Zhang and Lee, 2003; Hovy et al., 2002; Harabagiu et al., 2001). ***Question type*** often refers to the possible type of its answer, such as a person name, a location name, or a temporal expression. The question types used in this paper are PERSON, LOCATION, REASON, QUANTITY, TEMPORAL, COMPARISON, DEFINITION, METHOD, SELECTION, YESNO, and OTHER. Rules to determine question types are created manually.

This strategy tries to merge two question fragments of the same question type. This paper proposes two features to determine the threshold to merge two question fragments: length and sum of term weights of a fragment. Length is measured in characters and term weights are designed as in Table 3.

Merging algorithm is as follows: if the feature value of a question fragment is smaller than a threshold, it will be merged into the preceding question fragment (or the following fragment if no preceding one). This strategy applies recursively until no question fragment has a feature value lower than the threshold.

(5) Merging Example Fragments

If a fragment starts with a phrase such as “for example” or “such as”, it will be merged into its preceding question fragment.

4 Experiments

4.1 Experimental Data

All the experimental data were collected from Yahoo! Knowledge$^+$ (Yahoo! 奇摩知識$^+$)[6], discussion groups similar to Yahoo! Answers but using Chinese instead of English.

Three discussion groups, “Business Application” (商務應用), “Website Building” (網站架設), and “Image Processing” (影像處理), were selected to collect querying postings. The reason that we chose these three discussion groups was their moderate growing rates. We could collect enough amount of querying postings published in the same period of time.

The following kinds of postings were not selected as our experimental data:

1. No questions inside
2. Full of algorithms or program codes
3. Full of emoticons or Martian texts (火星文, a funny term used in Chinese to refer to a writing style that uses words with similar pronunciation to replace the original text)
4. Redundant postings

Totally 598 querying postings were collected as the training set and 269 postings as the test set. The real numbers of postings collected from each group are listed in Table 4, where “BA”, “WB”, and “IP” stand for “Business Application”, “Website Building”, and “Image Processing”, respectively.

Group	BA	WB	IP
Training Set	198	207	193
Test Set	101	69	99

Table 4. Numbers of Postings in the Data Set

Two persons were asked to mark garbage texts and separate questions in the whole data set. If a conflicting case occurred, a third person (who was one of the authors of this paper) would solve the inconsistency.

4.2 Garbage Texts Removal

The first factor examined in garbage text removal is the length threshold. Table 5 lists the experimental results on the training set and Table 6 on the test set. All garbage keywords are collected from the training set.

Eight experiments were conducted to use different values as length thresholds. The strategy *Len*k sets the length threshold to be *k* characters (no matter in Chinese or English). Hence, *Len*0 is one baseline strategy which removes only the garbage keyword itself. *LenS* is the other baseline strategy which removes the whole sentence fragment where a garbage keyword appears.

The strategy *Heu* uses different length thresholds for different classes of garbage keywords. The thresholds are heuristic values after observing many examples in the training set.

Accuracy is defined as the percentage of successful removal. In one posting, if all real garbage texts are correctly removed and no other text is wrongly deleted, it counts one successful removal.

Strategy	Accuracy (%)
Len0	64.21
LenS	27.59
Len1	73.91
Len2	78.43
Len3	80.60
Len4	78.26
Len5	71.91
Heu	**99.67**
HeuExp	**99.67**

Table 5. Accuracy of Garbage Text Removal with Different Length Thresholds (Training)

Strategy	Accuracy (%)
Len0	62.08
LenS	24.54
Len1	69.52
Len2	75.09
Len3	75.46
Len4	71.75
Len5	65.80
Heu	87.73
HeuExp	**92.57**

Table 6. Accuracy of Garbage Text Removal with Different Length Thresholds (Test Set)

As we can see in both tables, the two baseline strategies are poorer than any other strategy. It means that length threshold is useful to decide garbage existence.

Heu is the best strategy (99.67% on the training set and 87.73% on the test set). *Len*3 is

[6] http://tw.knowledge.yahoo.com/

the best strategy (80.60% on the training set and 75.49% on the test set) among *Len*k, but it is far worse than *Heu*. We can conclude that the length threshold should be assigned individually for each class of garbage words. If it is assigned carefully, the performance of garbage removal will be good.

The second factor is the expansion of garbage keywords. The strategy *HeuExp* is the same as *Heu* except that the list of garbage keywords was expanded as described in Section 2.2.

Comparing the last two rows in Table 6, *HeuExp* strategy improves the performance from 87.73% to 92.57%. It shows that a small amount of postings can provide good coverage of garbage keywords after keyword expansion by using available linguistic resources.

The results of *HeuExp* and *Heu* on the training set are the same. It makes sense because the expanded list suggests garbage existence in the training set no more than the original list does.

4.3 Question Segmentation

Overall Strategies

Six experiments were conducted to see the performance of different strategies for question segmentation. The strategies used in each experiment are:

Baseline: using only ‘?’ (question mark) to separate questions

SameS: removing repeated sentence fragments then separating by ‘?’

Interrg: after removing repeated sentence fragments, separating questions which are in interrogative forms

SimlrS: following the strategy *Interrg*, removing or merging similar sentence fragments of the same question type

ForInst: following the strategy *SimlrS*, merging a sentence fragment beginning with “for instance” and alike with its preceding question fragment

SameQT: following the strategy *ForInst*, merging question fragments of the same question type without considering similarity

Table 7 and Table 8 depict the results of the six experiments on the training set and the test set, respectively. The second column in each table lists the accuracy which is defined as the *percentage* of postings which are separated into the same number of questions as manually tagged. The third column gives the *number* of postings which are correctly separated. The fourth and the fifth columns contain the numbers of postings which are separated into more and fewer questions, respectively.

Strategy	Acc (%)	Same	More	Fewer
Baseline	50.67	303	213	82
SameS	59.03	353	156	89
Interrg	64.88	388	204	6
SimlrS	75.08	449	141	8
ForInst	75.75	453	137	8
SameQT	**88.29**	528	13	57

Table 7. Accuracy of Question Segmentation by Different Strategies (Training Set)

Strategy	Acc (%)	Same	More	Fewer
Baseline	54.28	146	84	39
SameS	65.43	176	54	39
Interrg	65.43	176	93	0
SimlrS	74.35	200	68	1
ForInst	74.35	200	68	1
SameQT	**85.87**	231	16	22

Table 8. Accuracy of Question Segmentation by Different Strategies (Test Set)

As we can see in Table 7, performance is improved gradually after adding new strategies. *SameQT* achieves the best performance with 88.29% accuracy. Same conclusion could also be made by the results on the test set. *SameQT* is the best one with 85.87% accuracy.

In Table 7, *Baseline* achieves only 50.67% accuracy. That matches our observations: (1) one question is often stated many times by sentences ended with question marks in one posting (as 213 postings were separated into more questions); (2) some users do not use ‘?’ in writing (as 82 postings were separated into fewer questions).

SameS greatly reduces the cases (57 postings) of separation into more questions by removing repeated sentences.

On the other hand, *Interrg* greatly reduces the cases (76 postings) of separation into fewer questions. Many question sentences without question marks were successfully captured by detecting the interrogative forms.

SimlrS also improves a lot (successfully reducing number of questions separated in 63 postings). But *ForInst* only improves a little. It is more common to express one question several times in different way than giving examples.

SameQT achieves the best performance, which means that question type is a good strategy. Different ways to express a question are usually in the same question type. Comparing with *SimlrS* which also considers sentence fragments in the same question type, more improvement comes from the successful merging of fragments with topic ellipses, co-references, or paraphrases. However, there may be other questions in the same question type which are wrongly merged together (as 49 failures in the training set).

Considering the results on the test set, *Interrg* does not improve the overall performance comparing to *SameS* because the improvement equals the drop. *ForInst* does not improve either. It seems that giving examples is not common in the discussion groups.

Thresholds in *SameQT*

In the strategy *SameQT*, two features, length and sum of term weights, are used to determine thresholds to merge question fragments as mentioned in Section 3.2. In order to decide which feature is better and which threshold value should be set, two experiments were conducted.

LenThr	Acc (%)	LenThr	Acc (%)
0	75.75	9	85.62
3	76.25	10	86.62
4	78.60	15	88.29
5	81.94	20	88.13
6	84.95	**30**	**88.63**
7	85.79	40	88.29
8	86.29	∞	**88.29**

Table 9. Accuracy of Question Segmentation with Different Length Thresholds

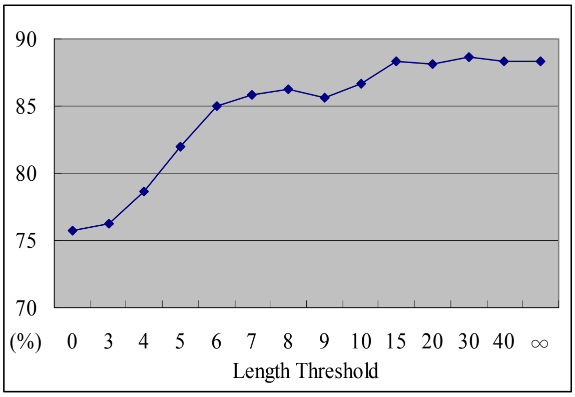

Figure 1. Accuracy of Question Segmentation with Different Length Thresholds

Table 9 depicts the experimental results of using length of sentence fragments as merging threshold. The column "LenThr" lists different settings of length threshold and the column "Acc" gives the accuracy.

The performance is gradually improved as the value of length threshold increases. The best one is LenThr=30 with 88.63% accuracy. However, "Always Merging" (LenThr=∞) achieves 88.29% accuracy, which is also acceptable comparing to the best performance. Fig 1 shows the curve of accuracy against length threshold.

Table 10 presents the experimental results of using sum of term weights as merging thresold. The column "WgtThr" lists different settings of length threshold and the column "Acc" gives the accuracy.

The performance is also gradually improved as the value of weight threshold increases. When WgtThr is set to be 500, 700, or 900, the performance is the best, with 88.46% accuracy. But the same as the threshold settings of length feature, the best one does not outperform "Always Merging" strategy (WgtThr=∞**,** 88.29% accuracy) too much. Fig 2 shows the curve of accuracy against similarity threshold.

WgtThr	Acc (%)	WgtThr	Acc (%)
0	75.75	350	87.29
50	77.93	400	88.13
100	83.11	450	88.29
150	85.28	**500**	**88.46**
200	86.29	**700**	**88.46**
250	86.79	**900**	**88.46**
300	87.46	∞	**88.29**

Table 10. Accuracy of Question Segmentation with Different Weight Thresholds

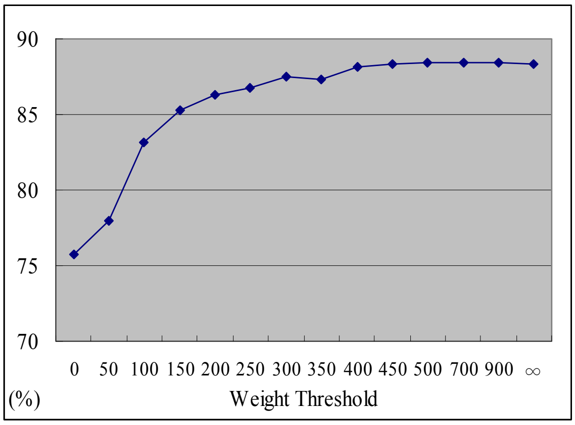

Figure 2. Accuracy of Question Segmentation with Different Weight Thresholds

From the results of above experiments, we can see that although using length feature with a threshold LenThr=30 achieves the best performance, "Always Merging" is more welcome for a online system because no feature extraction or computation is needed with only a little sacrifice of performance. Hence we choose "Always Merging" as merging strategy in *SameQT*.

5 Conclusion and Future Work

This paper proposes question pre-processing methods for a FQA-style QA system on discussion groups in the Internet. For a posting already existing or being submitted to a discussion group, garbage texts in it are removed first, and then different questions in it are identified so that they can be compared with other questions individually.

An expanded list of garbage keywords is used to detect garbage texts. If there is a garbage keyword appearing in a sentence fragment and the fragment has a length shorter than a threshold corresponding to the class of the garbage keyword, the fragment will be judged as a garbage text. This method achieves 92.57% accuracy on the test set. It means that a small set is sufficient to collect all classes of garbage keywords.

In question segmentation, sentence fragments in interrogative forms are considered as question fragments. Besides, repeated fragments are removed and fragments of the same question types are merged into one fragment. The overall accuracy is 85.87% on the test set.

In the future, performance of a QA system with or without question pre-processing will be evaluated to verify its value.

New methods to create the list of garbage keywords more robotically should be studied, as well as the automatic assignments of the length thresholds of classes of garbage keywords.

New feature should be discovered in the future in order to segment questions more accurately.

Although the strategies and the thresholds are developed according to experimental data in Chinese, we can see that many of them are language-independent or can be adapted with not too much effort.

Reference

Burke, Robin, Kristian Hammond, Vladimir Kulyukin, Steven Lytinen, Noriko Tomuro, and Scott Schoenberg (1997) "Natural language processing in the FAQFinder system: Results and prospects," *Proceedings of the 1997 AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, pp. 17-26.

Dice, Lee R. (1945) "Measures of the amount of ecologic association between species," *Journal of Ecology*, Vol. 26, pp. 297-302.

Harabagiu, Sanda, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Răzvan Bunescu, Roxana Gîrju, Vasile Rus, and Paul Morărescu (2001) "The Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering," *Proceedings of ACL-EACL 2001*, pp. 274-281.

Hovy, Eduard, Ulf Hermjakob, and Chin-Yew Lin (2002) "The Use of External Knowledge in Factoid QA," *Proceedings of TREC-10*, pp. 644-652.

Lai, Yu-Sheng, Kuao-Ann Fung, and Chung-Hsien Wu (2002) "FAQ Mining via List Detection," *Proceedings of the COLING Workshop on Multilingual Summarization and Question Answering*.

Lytinen, Steven and Noriko Tomuro (2002) "The use of question types to match questions in FAQFinder," *Proceedings of the 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pp. 46-53.

Saquete, Estela, Patricio Martinez-Barco, Rafael Munoz, and Jose Luis Vicedo Gonzalez (2004) "Splitting Complex Temporal Questions for Question Answering Systems," *Proceedings of ACL 2004*, pp. 566-573.

Soricut, Radu and Eric Brill (2004) "Automatic Question Answering: Beyond the Factoid," *Proceedings of HLT-NAACL 2004*, pp. 57-64.

Zhang, Dell and Wee Sun Lee (2003) "Question Classification using Support Vector Machines," *Proceedings of SIGIR 2003*, pp. 26-32.