Collecting a Why-question corpus for development and evaluation of an automatic QA-system

Joanna Mrozinski **Edward Whittaker** **Sadaoki Furui**
Department of Computer Science
Tokyo Institute of Technology
2-12-1-W8-77 Ookayama, Meguro-ku
Tokyo 152-8552 Japan
{mrozinsk,edw,furui}@furui.cs.titech.ac.jp

Abstract

Question answering research has only recently started to spread from short factoid questions to more complex ones. One significant challenge is the evaluation: manual evaluation is a difficult, time-consuming process and not applicable within efficient development of systems. Automatic evaluation requires a corpus of questions and answers, a definition of what is a correct answer, and a way to compare the correct answers to automatic answers produced by a system. For this purpose we present a Wikipedia-based corpus of Why-questions and corresponding answers and articles. The corpus was built by a novel method: paid participants were contacted through a Web-interface, a procedure which allowed dynamic, fast and inexpensive development of data collection methods. Each question in the corpus has several corresponding, partly overlapping answers, which is an asset when estimating the correctness of answers. In addition, the corpus contains information related to the corpus collection process. We believe this additional information can be used to post-process the data, and to develop an automatic approval system for further data collection projects conducted in a similar manner.

1 Introduction

Automatic question answering (QA) is an alternative to traditional word-based search engines. Instead of returning a long list of documents more or less related to the query parameters, the aim of a QA system is to isolate the exact answer as accurately as possible, and to provide the user only a short text clip containing the required information.

One of the major development challenges is evaluation. The conferences such as TREC[1], CLEF[2] and NTCIR[3] have provided valuable QA evaluation methods, and in addition produced and distributed corpora of questions, answers and corresponding documents. However, these conferences have focused mainly on fact-based questions with short answers, so called factoid questions. Recently more complex tasks such as list, definition and discourse-based questions have also been included in TREC in a limited fashion (Dang et al., 2007). More complex how- and why-questions (for Asian languages) were also included in the NTCIR07, but the provided data comprised only 100 questions, of which some were also factoids (Fukumoto et al., 2007). Not only is the available non-factoid data quite limited in size, it is also questionable whether the data sets are usable in development outside the conferences. Lin and Katz (2006) suggest that training data has to be more precise, and, that it should be collected, or at least cleaned, manually.

Some corpora of why-questions have been collected manually: corpora described in (Verberne et al., 2006) and (Verberne et al., 2007) both comprise fewer than 400 questions and corresponding answers (one or two per question) formulated by native speakers. However, we believe one answer per question is not enough. Even with factoid questions it is sometimes difficult to define what is a correct

[1] http://trec.nist.gov/
[2] http://www.clef-campaign.org/
[3] http://research.nii.ac.jp/ntcir/

Proceedings of ACL-08: HLT, pages 443–451,
Columbus, Ohio, USA, June 2008.

answer, and complex questions result in a whole new level of ambiguity. Correctness depends greatly on the background knowledge and expectations of the person asking the question. For example, a correct answer to the question "Why did Mr. X take Ms. Y to a coffee shop?" could be very different depending on whether we knew that Mr. X does not drink coffee or that he normally drinks it alone, or that Mr. X and Ms. Y are known enemies.

The problem of several possible answers and, in consequence, automatic evaluation has been tackled for years within another field of study: automatic summarisation (Hori et al., 2003; Lin and Hovy, 2003). We believe that the best method of providing "correct" answers is to do what has been done in that field: combine a multitude of answers to ensure both diversity and consensus among the answers.

Correctness of an answer is also closely related to the required level of detail. The Internet FAQ pages were successfully used to develop QA-systems (Jijkoun and de Rijke, 2005; Soricut and Brill, 2006), as have the human-powered question sites such as Answers.com, Yahoo Answers and Google Answers, where individuals can post questions and receive answers from peers (Mizuno et al., 2007). Both resources can be assumed to contain adequately error-free information. FAQ pages are created so as to answer typical questions well enough that the questions do not need to be repeated. Question sites typically rank the answers and offer bonuses for people providing good ones. However, both sites suffer from excess of information. FAQ-pages tend to also answer questions which are not asked, and also contain practical examples. Human-powered answers often contain unrelated information and discourse-like elements. Additionally, the answers do not always have a connection to the source material from which they could be extracted.

One purpose of our project was to take part in the development of QA systems by providing the community with a new type of corpus. The corpus includes not only the questions with multiple answers and corresponding articles, but also certain additional information that we believe is essential to enhance the usability of the data.

In addition to providing a new QA corpus, we hope our description of the data collection process will provide insight, resources and motivation for further research and projects using similar collection methods. We collected our corpus through Amazon Mechanical Turk service [4] (*MTurk*). The MTurk infrastructure allowed us to distribute our tasks to a multitude of workers around the world, without the burden of advertising. The system also allowed us to test the workers suitability, and to reward the work without the bureaucracy of employment. To our knowledge, this is the first time that the MTurk service has been used in equivalent purpose.

We conducted the data collection in three steps: generation, answering and rephrasing of questions. The workers were provided with a set of Wikipedia articles, based on which the questions were created and the answers determined by sentence selection. The WhyQA-corpus consists of three parts: original questions along with their rephrased versions, 8-10 partly overlapping answers for each question, and the Wikipedia articles including the ones corresponding to the questions. The WhyQA-corpus is in XML-format and can be downloaded and used under the GNU Free Documentation License from `www.furui.cs.titech.ac.jp/`.

2 Setup

Question-answer pairs have previously been generated for example by asking workers to both ask a question and then answer it based on a given text (Verberne et al., 2006; Verberne et al., 2007). We decided on a different approach for two reasons. Firstly, based on our experience such an approach is not optimal in the MTurk framework. The tasks that were welcomed by workers required a short attention span, and reading long texts was negatively received with many complaints, sloppy work and slow response times. Secondly, we believe that the aforementioned approach can produce unnatural questions that are not actually based on the information need of the workers.

We divided the QA-generation task into two phases: question-generation (*QGenHIT*) and answering (*QAHIT*). We also trimmed the amount of the text that the workers were required to read to create the questions. These measures were taken both in order to lessen the cognitive burden of the task

[4] http://www.mturk.com

and to produce more natural questions.

In the first phase the workers generated the questions based on a part of Wikipedia article. The resulting questions were then uploaded to the system as new HITs with the corresponding articles, and answered by available (different) workers. Our hypothesis is that the questions are more natural if their answer is not known at the time of the creation.

Finally, in an additional third phase, 5 rephrased versions of each question were created in order to gain variation (*QRepHIT*). The data quality was ensured by requiring the workers to achieve a certain result from a test (or a *Qualification*) before they could work on the aforementioned tasks.

Below we explain the MTurk system, and then our collection process in detail.

2.1 Mechanical Turk

Mechanical Turk is a Web-based service, offered by Amazon.com, Inc. It provides an API through which employers can obtain a connection to people to perform a variety of simple tasks. With tools provided by Amazon.com, the employer creates tasks, and uploads them to the MTurk Web-site. Workers can then browse the tasks and, if they find them profitable and/or interesting enough, work on them. When the tasks are completed, the employer can download the results, and accept or reject them. Some key concepts of the system are listed below, with short descriptions of the functionality.

- **HIT** Human Intelligence Task, the unit of a payable chore in MTurk.
- **Requester** An "employer", creates and uploads new *HIT*s and rewards the *workers*. Requesters can upload simple HITs through the MTurk Requester web site, and more complicated ones through the MTurk Web Service APIs.
- **Worker** An "employee", works on the hits through the MTurk Workers' web site.
- **Assignment**. One HIT consists of one or more assignments. One worker can complete a single HIT only once, so if the requester needs multiple results per HIT, he needs to set the assignment-count to the desired figure. A HIT is considered completed when all the assignments have been completed.
- **Rewards** At upload time, each HIT has to be assigned a fixed reward, that cannot be changed later. Minimum reward is $0.01. Amazon.com collects a 10% (or a minimum of $0.05) service fee per each paid reward.
- **Qualifications** To improve the data quality, a HIT can also be attached to certain tests, "qualifications" that are either system-provided or created by the requester. An example of a system-provided qualification is the average approval ratio of the worker.

Even if it is possible to create tests that workers have to pass before being allowed to work on a HIT so as to ensure the worker's ability, it is impossible to test the motivation (for instance, they cannot be interviewed). Also, as they are working through the Web, their working conditions cannot be controlled.

2.2 Collection process

The document collection used in our research was derived from the Wikipedia XML Corpus by Denoyer and Gallinari (2006). We selected a total of 84 articles, based on their length and contents. A certain length was required so that we could expect the article to contain enough interesting material to produce a wide selection of natural questions. The articles varied in topic, degree of formality and the amount of details; from "Horror film" and "Christmas worldwide" to "G-Man (Half-Life)" and "History of London". Articles consisting of bulleted lists were removed, but filtering based on the topic of the article was not performed. Essentially, the articles were selected randomly.

2.2.1 QGenHIT

The first phase of the question-answer generation was to generate the questions. In QGenHIT we presented the worker with only part of a Wikipedia article, and instructed them to think of a why-question that they felt could be answered based on the original, whole article which they were not shown. This approach was expected to lead to natural curiosity and questions. Offering too little information would have lead to many questions that would finally be left unanswered, and it also did not give the workers enough to work on. Giving too much information

Qualification	The workers were required to pass a test before working on the HITs.
QGenHIT	Questions were generated based on partial Wikipedia articles. These questions were then used to create the QAHITs.
QAHIT	Workers were presented with a question and a corresponding article. The task was to answer the questions (if possible) through sentence selection.
QRepHIT	To ensure variation in the questions, each question was rephrased by 5 different workers.

Table 1: Main components of the corpus collection process.

Article topic: Fermi paradox
Original question Why is the moon crucial to the rare earth hypothesis? **Rephrased Q 1** How does the rare earth theory depend upon the moon? **Rephrased Q 2** What makes the moon so important to rare earth theory? **Rephrased Q 3** What is the crucial regard for the moon in the rare earth hypothesis? **Rephrased Q 4** Why is the moon so important in the rare earth hypothesis? **Rephrased Q 5** What makes the moon necessary, in regards to the rare earth hypothesis?
Answer 1. Sentence ids: 20,21. Duplicates: 4. The moon is important because its gravitational pull creates tides that stabilize Earth's axis. Without this stability, its variation, known as precession of the equinoxes, could cause weather to vary so dramatically that it could potentially suppress the more complex forms of life. **Answer 2. Sentence ids: 18,19,20. Duplicates: 2.** The popular Giant impact theory asserts that it was formed by a rare collision between the young Earth and a Mars-sized body, usually referred to as Orpheus or Theia, approximately 4.45 billion years ago. The collision had to occur at a precise angle, as a direct hit would have destroyed the Earth, and a shallow hit would have deflected the Mars-sized body. The moon is important because its gravitational pull creates tides that stabilize Earth's axis. **Answer 3. Sentence ids: 20,21,22. Duplicates: 2.** The moon is important because its gravitational pull creates tides that stabilize Earth's axis. Without this stability, its variation, known as precession of the equinoxes, could cause weather to vary so dramatically that it could potentially suppress the more complex forms of life. The heat generated by the Earth/Theia impact, as well as subsequent Lunar tides, may have also significantly contributed to the total heat budget of the Earth's interior, thereby both strengthening and prolonging the life of the dynamos that generate Earth's magnetic field Dynamo 1. **Answer 4. Sentence ids: 18,20,21. No duplicates.** The popular Giant impact theory asserts that it was formed by a rare collision between the young Earth and a Mars-sized body, usually referred to as Orpheus or Theia, approximately 4.45 billion years ago. The moon is important because its gravitational pull creates tides that stabilize Earth's axis. Without this stability, its variation, known as precession of the equinoxes, could cause weather to vary so dramatically that it could potentially suppress the more complex forms of life. **Answer 5. Sentence ids: 18,21. No duplicates.** The popular Giant impact theory asserts that it was formed by a rare collision between the young Earth and a Mars-sized body, usually referred to as Orpheus or Theia, approximately 4.45 billion years ago. Without this stability, its variation, known as precession of the equinoxes, could cause weather to vary so dramatically that it could potentially suppress the more complex forms of life.

Table 2: Data example: Question with rephrased versions and answers.

(long excerpts from the articles) was severely disliked among the workers simply because it took a long time to read.

We finally settled on a solution where the partial content consisted of the title and headers of the article, along with the first sentences of each paragraph. The instructions to the questions demanded rigidly that the question starts with the word "Why", as it was surprisingly difficult to explain what we meant by why-questions if the question word was not fixed.

The reward per HIT was $0.04, and 10 questions were collected for each article. We did not force the questions to be different, and thus in the later phase some of the questions were removed manually as they were deemed to mean exactly the same thing. However, there were less than 30 of these duplicate questions in the whole data set.

2.2.2 QAHIT

After generating the questions based on partial articles, the resulting questions were uploaded to the system as HITs. Each of these QAHITs presented a single question with the corresponding original article. The worker's task was to select either 1-3 sentences from the text, or a No-answer-option (*NoA*). Sentence selection was conducted with Javascript functionality, so the workers had no chance to include freely typed information within the answer (although a comment field was provided). The reward per HIT was $0.06. At the beginning, we collected 10 answers per question, but we cut that down to 8 because the HITs were not completed fast enough.

The workers for QAHITs were drawn from the same pool as the workers for QGenHIT, and it was possible for the workers to answer the questions they had generated themselves.

2.2.3 QRepHIT

As the final step 5 rephrased versions of each question were generated. This was done to compensate the rigid instructions of the QGenHIT and to ensure variation in the questions. We have not yet measured how well the rephrased questions match the answers of their original versions. In the final QRepHIT questions were grouped into groups of 5. Each HIT consisted of 5 assignments, and a $0.05 reward was offered for each HIT.

QRepHIT required the least amount of design and trials, and workers were delighted with the task. The HITs were completed fast and well even in the case when we accidentally uploaded a set of HITs with no reward.

As with QAHIT, the worker pool for creating and rephrasing questions was the same. The questions were rephrased by their creator in 4 cases.

2.3 Qualifications

To improve the data quality, we used the qualifications to test the workers. For the QGenHITs we only used the system-provided "HIT approval rate"-qualification. Only workers whose previous work had been approved in 80% of the cases were able to work on our HITs.

In addition to the system-provided qualification, we created a why-question-specific qualification. The workers were presented with 3 questions, and they were to answer each by either selecting 1-3 most relevant sentences from a list of about 10 sentences, or by deciding that there is no answer present. The possible answer-sentences were divided into groups of essential, OK and wrong, and one of the questions did quite clearly have no answer. The scoring was such that it was impossible to get approved results if not enough essential sentences were included. Selecting sentences from the OK-group only was not sufficient, and selecting sentences from the wrong-group was penalized. A minimum score per question was required, but also the total score was relevant – component scores could compensate each other up to a point. However, if the question with no answer was answered, the score could not be of an approvable level. This qualification was, in addition to the minimum HIT approval rate of 80%, a prerequisite for both the QRepHITs and the QAHITs.

A total of 2355 workers took the test, and 1571 (67%) of them passed it, thus becoming our available worker pool. However, in the end the actual number of different workers was only 173.

Examples of each HIT, their instructions and the Qualification form are included in the final corpus. The collection process is summarised in Table 1.

3 Corpus description

The final corpus consists of questions with their rephrased versions and answers. There are total of 695 questions, of which 159 were considered unanswerable based on the articles, and 536 that have 8-10 answers each. The total cost of producing the corpus was about $350, consisting of $310 paid in workers rewards and $40 in Mechanical Turk fees, including all the trials conducted during the development of the final system.

Also included is a set of Wikipedia documents (*WikiXML*, about 660 000 articles or 670MB in compressed format), including the ones corresponding to the questions (84 documents). The source of WikiXML is the English part of the Wikipedia XML Corpus by Denoyer and Gallinari (2006). In the original data some of the HTML-structures like lists and tables occurred within sentences. Our sentence-selection approach to QA required a more fine-grained segmentation and for our purpose, much of the HTML-information was redundant anyway. Consequently we removed most of the HTML-structures, and the table-cells, list-items and other similar elements were converted into sentences. Apart from sentence-information, only the section-title information was maintained. Example data is shown in Table 2.

3.1 Task-related information

Despite the Qualifications and other measures taken in the collection phase of the corpus, we believe the quality of the data remains open to question. However, the Mechanical Turk framework provided additional information for each assignment, for example the time workers spent on the task. We believe this information can be used to analyse and use our data better, and have included it in the corpus to be used in further experiments.

- **Worker Id** Within the MTurk framework, each worker is assigned a unique id. Worker id can be used to assign a reliability-value to the workers, based on the quality of their previous work. It was also used to examine whether the same workers worked on the same data in different phases: Of the original questions, only 7 were answered and 4 other rephrased by the same worker they were created by. However, it has to be acknowledged that it is also possible for one worker to have had several accounts in the system, and thus be working under several different worker ids.

- **Time On Task** The MTurk framework also provides the requester the time it took for the worker to complete the assignment after accepting it. This information is also included in the corpus, although it is impossible to know precisely how much time the workers actually spent on each task. For instance, it is possible that one worker had several assignments open at the same time, or that they were not concentrating fully on working on the task. A high value of Time On Task thus does not necessarily mean that the worker actually spent a long time on it. However, a low value indicates that he/she did only spend a short time on it.

- **Reward** Over the period spent collecting the data, we changed the reward a couple of times to speed up the process. The reward is reported per HIT.

- **Approval Status** Within the collection process we encountered some clearly unacceptable work, and rejected it. The rejected work is also included in the corpus, but marked as rejected. The screening process was by no means perfect, and it is probable that some of the approved work should have been rejected.

- **HIT id, Assignment id, Upload Time** HIT and assignment ids and original upload times of the HITs are provided to make it possible to retrace the collection steps if needed.

- **Completion Time** Completion time is the timestamp of the moment when the task was completed by a worker and returned to the system. The time between the completion time and the upload time is presumably highly dependent on the reward, and on the appeal of the task in question.

3.2 Quality experiments

As an example of the post-processing of the data, we conducted some preliminary experiments on the answer agreement between workers.

Out of the 695 questions, 159 were filtered out in the first part of QAHIT. We first uploaded only 3 assignments, and the questions that 2 out of 3 workers deemed unanswerable were filtered out. This left 536 questions which were considered answered, each one having 8-10 answers from different workers. Even though in the majority of cases (83% of the questions) one of the workers replied with the NoA, the ones that answered did agree up to a point: of all the answers, 72% were such that all of their sentences were selected by at least two different workers. On top of this, an additional 17% of answers shared at least one sentence that was selected by more than one worker.

To understand the agreement better, we also calculated the average agreement of selected sentences based on sentence ids and N-gram overlaps between the answers. In both of these experiments, only those 536 questions that were considered answerable were included.

3.2.1 Answer agreement on sentence ids

As the questions were answered by means of sentence selection, the simplest method to check the agreement between the workers was to compare the ids of the selected sentences. The agreement was calculated as follows: each answer was compared to all the other answers for the same question. For each case, the agreement was defined as $Agreement = \frac{CommonIds}{AllIds}$, where $CommonIds$ is the number of sentence ids that existed in both answers, and $AllIds$ is the number of different ids in both answers. We calculated the overall average agreement ratio (*Total Avg*) and the average of the best matches between two assignments within one HIT (*Best Match*). We ran the test for two data sets: The most typical case of the workers cheating was to mark the question unaswerable. Because of this the first data set included only the real answers, and the NoAs were removed (*NoA not included*, 3872 answers). If an answer was compared with a NoA, the agreement was 0, and if two NoAs were compared, the agreement was 1. We did, however, also include the figures for the whole data set (*NoA included*, 4638 answers). The results are shown in Table 3.

The Best Match -results were quite high compared to the Total Avg. From this we can conclude that in the majority of cases, there was at least one quite similar answer among those for that HIT. However, comparing the sentence ids is only an indicative measure, and it does not tell the whole story about agreement. For each document there may exist several separate sentences that contain the same kind of information, and so two answers can be alike even though the sentence ids do not match.

	Total Avg	Best Match
NoA not included	0.39	0.68
NoA included	0.34	0.68

Table 3: Answer agreement based on sentence ids.

3.2.2 Answer agreement based on ROUGE

Defining the agreement over several passages of texts has for a long time been a research problem within the field of automatic summarisation. For each document it is possible to create several summarisations that can each be considered correct. The problem has been approached by using the ROUGE-metric: calculating the N-gram overlap between manual, "correct" summaries, and the automatic summaries. ROUGE has been proven to correlate well with human evaluation (Lin and Hovy, 2003).

Overlaps of higher order N-grams are more usable within speech summarisation as they take the grammatical structure and fluency of the summary into account. When selecting sentences, this is not an issue, so we decided to use only unigram and bigram counts (Table 4: *R-1, R2*), as well as the skip-bigram values (*R-SU*) and the longest common N-gram metric R-L. We calculated the figures for two data sets in the same way as in the case of sentence id agreement. Finally, we set a lower bound for the results by comparing the answers to each other randomly (the NoAs were also included).

The final F-measures of the ROUGE results are presented in Table 4. The figures vary from 0.37 to 0.56 for the first data set, and from 0.28 to 0.42 to the second. It is debatable how the results should be interpreted, as we have not defined a theoretical upper bound to the values, but the difference to the randomised results is substantial. In the field of automatic summarisation, the overlap of the automatic

results and corresponding manual summarisations is generally much lower than the overlap between our answers (Chali and Kolla, 2004). However, it is difficult to draw detailed conclusions based on comparison between these two very different tasks.

	R-1	R-2	R-SU	R-L
NoA not included	0.56	0.46	0.37	0.52
NoA included	0.42	0.35	0.28	0.39
Random Answers	0.13	0.01	0.02	0.09

Table 4: Answer agreement: ROUGE-1, -2, -SU and -L.

The sentence agreement and ROUGE-figures do not tell us much by themselves. However, they are an example of a procedure that can be used to post-process the data and in further projects of similar nature. For example, the ROUGE similarity could be used in the data collection phase as a tool of automatic approval and rejection of workers' assignments.

4 Discussion and future work

During the initial trials of data collection we encountered some unexpected phenomena. For example, increasing the reward did have a positive effect in reducing the time it took for HITs to be completed, however it did not correlate in desirable way with data quality. Indeed the quality actually decreased with increasing reward. We believe that this unexpected result is due to the distributed nature of the worker pool in Mechanical Turk. Clearly the motivation of some workers is other than monetary reward. Especially if the HIT is interesting and can be completed in a short period of time, it seems that there are people willing to work on them even for free.

MTurk requesters cannot however rely on this voluntary workforce. From MTurk Forums it is clear that some of the workers rely on the money they get from completing the HITs. There seems to be a critical reward-threshold after which the "real workforce", i.e. workers who are mainly interested in performing the HITs as fast as possible, starts to participate. When the motivation changes from voluntary participation to maximising the monetary gain, the quality of the obtained results often understandably suffers.

It would be ideal if a requester could rely on the voluntary workforce alone for results, but in many cases this may result either in too few workers and/or too slow a rate of data acquisition. Therefore it is often necessary to raise the reward and rely on efficient automatic validation of the data.

We have looked into the answer agreement of the workers as an experimental post-processing step. We believe that further work in this area will provide the tools required for automatic data quality control.

5 Conclusions

In this paper we have described a dynamic and inexpensive method of collecting a corpus of questions and answers using the Amazon Mechanical Turk framework. We have provided to the community a corpus of questions, answers and corresponding documents, that we believe can be used in the development of QA-systems for why-questions. We propose that combining several answers from different people is an important factor in defining the "correct" answer to a why-question, and to that goal have included several answers for each question in the corpus.

We have also included data that we believe is valuable in post-processing the data: the work history of a single worker, the time spent on tasks, and the agreement on a single HIT between a set of different workers. We believe that this information, especially the answer agreement of workers, can be successfully used in post-processing and analysing the data, as well as automatically accepting and rejecting workers' submissions in similar future data collection exercises.

Acknowledgments

This study was funded by the Monbusho Scholarship of Japanese Government and the 21st Century COE Program "Framework for Systematization and Application of Large-scale Knowledge Resources (COE-LKR)"

References

Yllias Chali and Maheedhar Kolla. 2004. Summarization Techniques at DUC 2004. In *DUC2004*.

Hoa Trang Dang, Diane Kelly, and Jimmy Lin. 2007. Overview of the TREC 2007 Question Answering

Track. In E. Voorhees and L. P. Buckland, editors, *Sixteenth Text REtrieval Conference (TREC)*, Gaithersburg, Maryland, November.

Ludovic Denoyer and Patrick Gallinari. 2006. The Wikipedia XML Corpus. *SIGIR Forum*.

Junichi Fukumoto, Tsuneaki Kato, Fumito Masui, and Tsunenori Mori. 2007. An Overview of the 4th Question Answering Challenge (QAC-4) at NTCIR workshop 6. In *Proceedings of the Sixth NTCIR Workshop Meeting*, pages 433–440.

Chiori Hori, Takaaki Hori, and Sadaoki Furui. 2003. Evaluation Methods for Automatic Speech Summarization. In *In Proc. EUROSPEECH*, volume 4, pages 2825–2828, Geneva, Switzerland.

Valentin Jijkoun and Maarten de Rijke. 2005. Retrieving Answers from Frequently Asked Questions Pages on the Web. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83, New York, NY, USA. ACM Press.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Human Technology Conference (HLT-NAACL)*, Edmonton, Canada.

Jimmy Lin and Boris Katz. 2006. Building a Reusable Test Collection for Question Answering. *J. Am. Soc. Inf. Sci. Technol.*, 57(7):851–861.

Junta Mizuno, Tomoyosi Akiba, Atsushi Fujii, and Katunobu Itou. 2007. Non-factoid Question Answering Experiments at NTCIR-6: Towards Answer Type Detection for Realworld Questions. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 487–492.

Radu Soricut and Eric Brill. 2006. Automatic Question Answering Using the Web: Beyond the Factoid. *Inf. Retr.*, 9(2):191–206.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2006. Data for Question Answering: the Case of Why. In *LREC*.

Susan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Discourse-based Answering of Why-questions. *Traitement Automatique des Langues*, 47(2: Discours et document: traitements automatiques):21–41.