Corpus-based Question Answering for *why*-Questions

Ryuichiro Higashinaka and **Hideki Isozaki**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{rh,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a corpus-based approach for answering *why*-questions. Conventional systems use hand-crafted patterns to extract and evaluate answer candidates. However, such hand-crafted patterns are likely to have low coverage of causal expressions, and it is also difficult to assign suitable weights to the patterns by hand. In our approach, causal expressions are automatically collected from corpora tagged with semantic relations. From the collected expressions, features are created to train an answer candidate ranker that maximizes the QA performance with regards to the corpus of why-questions and answers. NAZEQA, a Japanese why-QA system based on our approach, clearly outperforms a baseline that uses hand-crafted patterns with a Mean Reciprocal Rank (top-5) of 0.305, making it presumably the best-performing fully implemented why-QA system.

1 Introduction

Following the trend of non-factoid QA, we are seeing the emergence of work on why-QA; e.g., answering generic "why X?" questions (Verberne, 2006). However, since why-QA is an inherently difficult problem, there have only been a small number of fully implemented systems dedicated to solving it. Recent systems at NTCIR-6[1] Question Answering Challenge (QAC-4) can handle why-questions (Fukumoto et al., 2007). However, their performance is much lower (Mori et al., 2007) than that of factoid QA systems (Fukumoto et al., 2004; Voorhees and Dang, 2005).

We consider that this low performance is due to the great amount of hand-crafting involved in the systems. Currently, most of the systems rely on hand-crafted patterns to extract and evaluate answer candidates (Fukumoto et al., 2007). Such patterns include typical cue phrases and POS-tag sequences related to causality, such as "*because of*" and "*by reason of*." However, as noted in (Inui and Okumura, 2005), causes are expressed in various forms, and it is difficult to cover all such expressions by hand. Hand-crafting is also very costly. Some patterns may be more indicative of causes than others. Therefore, it may be useful to assign different weights to the patterns for better answer candidate extraction, but currently this must be done by hand (Mori et al., 2007). It is not clear whether the weights determined by hand are suitable.

In this paper, we propose a corpus-based approach for why-QA in order to reduce this hand-crafting effort. We automatically collect causal expressions from corpora to improve the coverage of causal expressions, and utilize a machine learning technique to train a ranker of answer candidates on the basis of features created from the expressions together with other possible features related to causality. The ranker is trained to maximize the QA performance with regards to a corpus of why-questions and answers, automatically tuning the weights of the features.

This paper is organized as follows: Section 2 describes previous work on why-QA, and Section 3 describes our approach. Section 4 describes the implementation of our approach, and Section 5 presents the evaluation results. Section 6 summarizes and mentions future work.

2 Previous Work

Although systems that can answer why-questions are emerging, they tend to have limitations in that they can answer questions only with causal verbs (Girju, 2003), in specific domains (Khoo et al.,

[1] http://research.nii.ac.jp/ntcir/ntcir-ws6/ws-en.html

2000), or questions covered by a specific knowledge base (Curtis et al., 2005). Recently, Verberne (2006; 2007a) has been intensively working on why-QA based on the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). However, her approach requires manually annotated corpora with RST relations.

When we look for fully implemented systems for generic "why X?" questions, we only find a small number of such systems. Since why-QA would be a challenging task when tackled straightforwardly, requiring common-sense knowledge and semantic interpretation of questions and answer candidates, current systems place higher priority on achievability and therefore use hand-crafted patterns and heuristics to extract causal expressions as answer candidates and use conventional sentence similarity metrics for answer candidate evaluation (Fukumoto, 2007; Mori et al., 2007). We argue, in this paper, that this hand-crafting is the cause of the current low performance levels. Recently, (Shima and Mitamura, 2007) applied a machine learning approach to why-QA, but they also rely on manually selected cue words to create their features.

Semantic Role Labeling (SRL) techniques can be used to automatically detect causal expressions. In the CoNLL-2005 shared task (SRL for English), the best system found causal adjuncts with a reasonable accuracy of 65% (Màrquez et al., 2005). However, when we analyzed the data, we found that more than half of the causal adjuncts contain explicit cues such as "*because*." Since causes are reported to be expressed by a wide variety of linguistic phenomena, not just explicit cues (Inui and Okumura, 2005), further verification is needed before SRL can be safely used for why-QA.

Why-questions are a subset of non-factoid questions. Since non-factoid questions are observed in many FAQ sites, such sites have been regarded as valuable resources for the development of non-factoid QA systems. Examples include Burke et al. (1997), who used FAQ corpora to analyze questions to achieve accurate question-type matching; Soricut and Brill (2006), who used them to train statistical models for answer evaluation and formulation; and Mizuno et al. (2007), who used them to train classifiers of question and answer-types. However, they do not focus on why-questions and do not use any causal knowledge, which is considered to be useful for explicit why-questions (Soricut and Brill, 2006).

3 Approach

In this paper, we propose a corpus-based approach for why-QA in order to reduce the hand-crafting effort that is currently necessary. We first automatically collect causal expressions from corpora and use them to create features to represent an answer candidate. The features are then used to train an answer candidate ranker that maximizes the QA performance with regards to a corpus of why-questions and answers. We also enumerate possible features that may be useful for why-QA to be incorporated in the training to improve the QA performance.

Following the systems at QAC-4 (Fukumoto, 2007) and the answer analysis in (Verberne, 2007b; Verberne et al., 2007), we consider the task of why-QA to be a sentence/paragraph extraction task. We also assume that a document retrieval module of a system returns top-N documents for a question on the basis of conventional IR-related metrics and all sentences/paragraphs extracted from them are regarded as answer candidates. Hence, the task becomes the ranking of given sentences/paragraphs.

For an answer candidate (a sentence or a paragraph) to be the correct answer, the candidate should (1) have an expression indicating a cause and (2) be similar to the question in content, and (3) some causal relation should be observed between the candidate and the question. For example, an answer candidate "X was arrested for fraud." is likely to be a correct answer to the question "Why was X arrested?" because "for fraud" expresses a cause, the question and the answer are both about the same event (X being arrested), and "fraud" and "arrest" indicate a causal relation between the question and the candidate. Condition (3) would be especially useful when the candidates do not have obvious cues or topically similar words/phrases to the question; it may be worthwhile to rely on some prior causal knowledge to select one over others. Although current working systems (Fukumoto, 2007; Mori et al., 2007) do not explicitly state these conditions, they can be regarded as using hand-crafted patterns for (1) and (3).[2] Lexical similarity metrics, such as cosine similarity and n-gram overlaps, are generally used for (2).

We represent each answer candidate with causal expression, content similarity, and causal relation

[2](3) is dealt with in a manner similar to the treatment of '`cause_of_death`' in (Smith et al., 2005).

features that encode how it complies with the three conditions. Here, the causal expression features are those based on the causal expressions we aim to collect automatically. For the other two types of features, we turn to the existing similarity metrics and dictionaries to derive features that would be useful for why-QA. To train a ranker, we create a corpus of why-questions and answers and adopt one of the machine learning algorithms for ranking. The following sections describe the three types of features, the corpus creation, and the ranker training. The actual instances of the features, the corpus, and the ranker will be presented in Section 4.

3.1 Causal Expression Features

With the increasing attention paid to SRL, we currently have a number of corpora, such as PropBank (Palmer, 2005) and FrameNet (Baker et al., 1998), that are tagged with semantic relations including a causal relation. Since text spans for such relations are annotated in the corpora, we can simply collect the spans marked by a causal relation as causal expressions. Since an answer candidate that has a matching expression for one of the collected causal expressions is likely to be expressing a cause as well, we can make the existence of each expression a feature. Although the collected causal expressions without any modification might be used to create features, for generality, it would be better to abstract them into syntactic patterns. From *m* causal expressions/patterns automatically extracted from corpora, we can create *m* binary features.

In addition, some why-QA systems may already possess some good hand-crafted patterns to detect causal expressions. Since there is no reason not to use them if we know they are useful for why-QA, we can create a feature indicating whether an answer candidate matches existing hand-crafted patterns.

3.2 Content Similarity Features

In general, if a question and an answer candidate share many words, it is likely that they are about the same content. From this assumption, we create a feature that encodes the lexical similarity of an answer candidate to the question. To calculate its value, existing sentence similarity metrics, such as cosine similarity or n-gram overlaps, can be used.

Even if a question and an answer candidate do not share the same words, they may still be about the same content. One such case is when they are about the same topic. To express this case as a feature, we can use the similarity of the question and the document in which the answer candidate is found. Since the documents from which we extract answer candidates typically have scores output by an IR engine that encode their relevance to the question, we can use this score or simply the rank of the retrieved document as a feature.

A question and an answer candidate may be semantically expressing the same content with different expressions. The simplest case is when synonyms are used to describe the same content; e.g., when "arrest" is used instead of "apprehend." For such cases, we can exploit existing thesauri. We can create a feature encoding whether synonyms of words in the question are found in the answer candidate. We could also use the value of semantic similarity and relatedness measures (Pedersen et al., 2004) or the existence of hypernym or hyponym relations as features.

3.3 Causal Relation Features

There are semantic lexicons where a semantic relation between concepts is indicated. For example, the EDR dictionary[3] shows whether a causal relation holds between two concepts; e.g., between "murder" and "arrest." Using such dictionaries, we can create pairs of expressions, one indicating a cause and the other its effect. If we find an expression for a cause in the answer candidate and that for an effect in the question, it is likely that they hold a causal relation. Therefore, we can create a feature encoding whether this is the case. In cases where such semantic lexicons are not available, they may be automatically constructed, although with noise, using causal mining techniques such as (Marcu and Echihabi, 2002; Girju, 2003; Chang and Choi, 2004).

3.4 Creating a QA Corpus

For ranker training, we need a corpus of why-questions and answers. Because we regard the task of why-QA as a ranking of given sentences/paragraphs, it is best to prepare the corpus in the same setting. Therefore, we use the following procedure to create the corpus: (a) create a question, (b) use an IR engine to retrieve documents for the question, (c) select among all sentences/paragraphs in the retrieved documents those that contain the answer to the question, and (d) store the question and a

[3] http://www2.nict.go.jp/r/r312/EDR/index.html

set of selected sentences/paragraphs with their document IDs as answers.

3.5 Training a Ranker

Having created the QA corpus, we can apply existing machine learning algorithms for ranking, such as RankBoost (Freund et al., 2003) or Ranking SVM (Joachims, 2002), so that the selected sentences/paragraphs are preferred to non-selected ones on the basis of their features. Good ranking would result in good Mean Reciprocal Rank (MRR), which is one of the most commonly used measures in QA.

4 Implementation

Using our approach, we implemented a Japanese why-QA system, **NAZEQA** ("Naze" means "why" in Japanese). The system was built as an extension to our factoid QA system, SAIQA (Isozaki, 2004; Isozaki, 2005), and works as follows:

1. The question is analyzed by a rule-based question analysis component to derive a question type; 'REASON' for a why-question.
2. The document retrieval engine extracts n-best documents from Mainichi newspaper articles (1998–2001) using DIDF (Isozaki, 2005), a variant of the IDF metric. We chose 20 as n. All sentences/paragraphs in the n documents are extracted as answer candidates. Whether to use sentences or paragraphs as answer candidates is configurable.
3. The feature extraction component produces, for each answer candidate, causal expression, content similarity, and causal relation features encoding how it satisfies conditions (1)–(3) described in Section 3.
4. The SVM ranker trained by a QA corpus ranks the answer candidates based on the features.
5. The top-N answer candidates are presented to the user as answers.

In the following sections, we describe the features (399 in all), the QA corpus, and the ranker.

4.1 Causal Expression Features

(F1–F394: AUTO-Causal Expression) We automatically extracted causal expressions from the EDR dictionary. The EDR dictionary is a suite of corpora and dictionaries and includes the EDR corpus, the EDR concept dictionary (hierarchy of word senses), and the EDR Japanese word dictionary (sense to word mappings). The EDR corpus is a collection of independent Japanese sentences taken from various sources, such as newspaper articles, magazines, and dictionary glosses. The corpus is annotated with semantic relations including a causal relation in a manner similar to PropBank and FrameNet corpora. We extracted regions marked by 'cause' tags and abstracted them by leaving only the functional words (auxiliary verbs and case, aspect, tense markers) and replacing others with wildcards '*.' For example, a causal expression "arrested for fraud" would be abstracted to "*-PASS for *." We used CaboCha[4] as a morphological analyzer. From 8,747 regions annotated with 'cause,' we obtained 394 causal expression patterns after filtering out those that occurred only once. Finally, we have 394 binary features representing the existence of each abstracted causal expression pattern.

(F395: MAN-Causal Expression) We emulate the manually created patterns described in (Fukumoto, 2007) and create a binary feature indicating whether an answer candidate is matched by the patterns.

4.2 Content Similarity Features

(F396: Question-Candidate Cosine Similarity) We use the cosine similarity between a question and an answer candidate using the word frequency vectors of the content words. We chose nouns, verbs, and adjectives as content words.

(F397: Question-Document Relevance) We use, as a feature, the inverse of the rank of the document where the answer candidate is found.

(F398: Synonym Pair) This is a binary feature that indicates whether a word and its synonym appear in an answer candidate and a question, respectively. We use the combination of the EDR concept dictionary and the EDR Japanese word dictionary as a thesaurus to collect synonym pairs. We have 133,486 synonym pairs.

4.3 Causal Relation Feature

(F399: Cause-Effect Pair) This is a binary feature that indicates whether a word representing a cause and a word corresponding to its effect appear in an answer candidate and a question, respectively. We used the EDR concept dictionary to find pairs of word senses holding a causal relation and

[4] http://chasen.org/~taku/software/cabocha/

Q13: Why are pandas on the verge of extinction? (000217262)

A:000217262,L2 Since pandas are not good at raising their offspring, the Panda Preservation Center in Sichuan Province is promoting artificial insemination as well as the training of mother pandas.

A:000217262,L3 A mother panda often gives birth to two cubs, but when there are two cubs, one is discarded, and young mothers sometimes crush their babies to death.

A:000406060,L6 However, because of the recent development in the midland, they are becoming extinct.

A:010219075,L122 The most common cause of the extinction for mammals, birds, and plants is degradation and destruction of habitat, followed by hunting and poaching for mammals and the impact of alien species for birds.

Figure 1: An excerpt from the WHYQA collection. The number in parentheses is the ID of the document used to come up with the question. The answers were headed by the document ID and the line number where the sentence is found in the document. (N.B. The above sentences were translated by the authors.)

expanded the senses to corresponding words using the EDR Japanese word dictionary to create cause-effect word pairs. We have 355,641 cause-effect word pairs.

4.4 WHYQA Collection

Since QAC-4 does not provide official answer sets and their questions include only a small number of why-questions, we created a corpus of why-questions and answers on our own.

An expert, who specializes in text analysis and is not one of authors, created questions from articles randomly extracted from Mainichi newspaper articles (1998–2001). Then, for each question, she created sentence-level answers by selecting the sentences that she considered to fully include the answer from a list of sentences from top-20 documents returned from the text retrieval engine with the question as input. Paragraph-level answers were automatically created from the sentence-level answers by selecting the paragraphs containing the answer sentences.

The analyst was instructed not to create questions by simply converting existing declarative sentences into interrogatives. It took approximately five months to create 1,000 question and answer sets (called the WHYQA collection). All questions are guaranteed to have answers. Figure 1 lists an example question and answer sentences in the collection.

4.5 Training a Ranker by Ranking SVM

Using the WHYQA collection, we trained ranking models using the ranking SVM (Joachims, 2002) (with a linear kernel) that minimizes the pairwise ranking error among the answer candidates. In the training data, the answers were labeled '+1' and non-answers '-1.' When using sentences as answers, there are 4,849 positive examples and 521,177 negative examples. In the case of paragraphs, there are 4,371 positive examples and 261,215 negative examples.

5 Evaluation

For evaluation, we compared the proposed system (**NAZEQA**) with two baselines. Baseline-1 (**COS**) simply uses, for answer candidate evaluation, the cosine similarity between an answer candidate and a question based on frequency vectors of their content words. The aim of having this baseline is to see how the system performs without any use of causal knowledge. Baseline-2 (**FK**) uses hand-crafted patterns described in (Fukumoto, 2007) to narrow down the answer candidates to those having explicit causal expressions, which are then ranked by the cosine similarity to the question. NAZEQA and the two baselines used the same document retrieval engine to obtain the top-20 documents and ranked the sentences or paragraphs in these documents.

5.1 Results

We made each system output the top-1, 5, 10, and 20 answer sentences and paragraphs for all 1,000 questions in the WHYQA collection. We used the MRR and coverage as the evaluation metrics. **Coverage** means the rate of questions that can be answered by the top-N answer candidates. Table 1 shows the MRRs and coverage for the baselines and NAZEQA. A 10-fold cross validation was used for the evaluation of NAZEQA.

We can see from the table that NAZEQA is better in all comparisons. A statistical test (a sign test that compares the number of times one system places the correct answer before the other) showed that NAZEQA is significantly better than FK for the top-5, 10, and 20 answers in the sentence and paragraph-levels ($p<0.01$). Although the sentence-level MRR for NAZEQA is rather low, the paragraph-level MRR for the top-5 answers is 0.305, which is reasonably high for a non-factoid QA system (Mizuno et al., 2007). The coverage is also

	MRR			Coverage		
top-N	COS	FK	**NZQ**	COS	FK	**NZQ**
Sentences as answer candidates:						
top-1	0.036	0.091+	0.113	3.6%	9.1%	11.3%
top-5	0.086	0.139+	**0.196***	19.1%	23.1%	**35.4%**
top-10	0.102	0.149+	**0.216***	31.3%	30.7%	**50.4%**
top-20	0.115	0.152	**0.227***	51.4%	35.5%	**66.6%**
Paragraphs as answer candidates:						
top-1	0.065	0.152+	0.186	6.5%	15.2%	18.6%
top-5	0.140	0.245+	**0.305***	29.2%	41.6%	**53.1%**
top-10	0.166	0.257+	**0.328***	48.8%	50.5%	**70.3%**
top-20	0.181	0.262+	**0.339***	70.7%	56.4%	**85.6%**

Table 1: Mean Reciprocal Rank (MRR) and coverage for the baselines (COS and FK) and the proposed NAZEQA (NZQ in the table) system for the entire WHYQA collection. The top-1, 5, 10, and 20 mean the numbers of topmost candidates used to calculate MRR and coverage. Asterisks indicate NAZEQA's statistical significance ($p<0.01$) over FK, and '+' FK's over COS.

Feature Set	Sent.	Para.
All features (**NAZEQA**)	0.181	0.287
w/o F1–F394 (AUTO-Causal Exp.)	**0.138***	**0.217***
w/o F395 (MAN-Causal Exp.)	0.179	0.286
w/o F396 (Q-Cand. Cosine Similarity)	**0.131***	**0.188***
w/o F397 (Doc.-Q Relevance)	0.161	0.275
w/o F398 (Synonym Pair)	0.180	0.282
w/o F399 (Cause-Effect Pair)	*0.184*	0.287

Table 2: Performance changes in MRR (top-5) when we exclude one of the feature sets. Asterisks indicate a statistically significant drop in performance from NAZEQA. In this experiment, we used a two-fold cross validation to reduce computational cost.

high for NAZEQA, making it possible to find answers within the top-10 sentences and top-5 paragraphs for more than 50% of the questions. Because there are no why-QA systems known to be better than NAZEQA in MRR and coverage and because NAZEQA clearly outperforms a competitive baseline (FK), we conclude that NAZEQA has one of the best performance levels for why-QA.

It is interesting to know how each of the feature sets (e.g., AUTO-Causal Expression Features) contributes to the QA performance. Table 2 shows how the performance in MRR (top-5) changes when one of the feature sets is excluded in the training. Although the drop in performance by removing the Question-Candidate Cosine Similarity feature is understandable, the performance also drops significantly from NAZEQA when we exclude AUTO-Causal Expression features, showing the effectiveness of our automatically collected causal patterns.

Rank	Feature Name	Weight
1	Question-Candidate Cosine Similarity	4.66
2	Exp.[*de* (by) * *wo* (-ACC) * *teshimai* (-PERF)]	1.86
3	Exp.[*no* (of) * *niyote wa* (according to)]	1.44
4	Exp.[*no* (of) * *na* (AUX) * *no* (of) * *de* (by)]	1.42
5	Exp.[*no* (of) * *ya* (or) * *niyotte* (by)]	1.35
6	Exp.[*no* (of) * *ya* (or) * *no* (of) * *de* (by)]	1.30
7	Exp.[*na* (AUX) * *niyotte* (by)]	1.23
8	Exp.[*koto niyotte* (by the fact that)]	1.22
9	Exp.[*to* (and) * *no* (of) * *niyotte* (by)]	1.20
10	Document-Question Relevance	0.89
	⋮	
27	Synonym Pair	0.40
102	MAN-Causal Expression	0.16
127	Cause-Effect Pair	0.15

Table 3: Weights of features learned by the ranking SVM. 'AUTO-Causal Expression' is denoted as 'Exp.' for lack of space. AUX means an auxiliary verb. The abstracted causal expression patterns are shown in square brackets with their English translations in parentheses.

The MAN-Causal Expression, Synonym Pair, and Cause-Effect Pair features, do not seem to contribute much to the performance. One of the reasons for the small contribution of the MAN-Causal Expression feature may be that the manual patterns used to create this feature overlap greatly with the automatically collected causal expression patterns, lowering the impact of the MAN-Causal Expression feature. The small contribution of the Synonym Pair feature is probably attributed to the way the answers were created in the creation of the WHYQA Collection. Since the answer candidates from which the expert chose the answers were those retrieved by a text retrieval engine that uses lexical similarity to retrieve relevant documents, it is possible that the answers that contain synonyms had already been filtered out in the beginning, making the Synonym Pair feature less effective. Without the Cause-Effect Pair feature, the performance does not change or even improves a little when sentences are used as answers. The reason for this may be that the syntactically well-formed sentences of the newspaper articles might have made causal cues and patterns more effective than prior causal knowledge. We need to investigate the difference between the manually created causal patterns and the automatically collected ones. We also need to investigate whether the Synonym Pair and Cause-Effect Pair features could be useful in other conditions; e.g., when answers are created in different ways. We also need to examine the quality of our synonym and cause-effect word pairs because

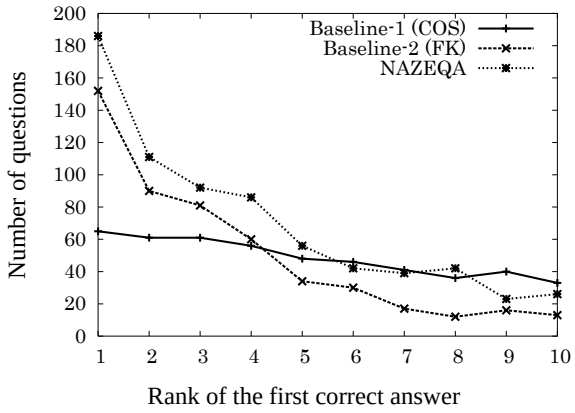

Figure 2: Distribution of the ranks of first correct answers. Paragraphs were used as answers. A 10-fold cross validation was used to evaluate NAZEQA.

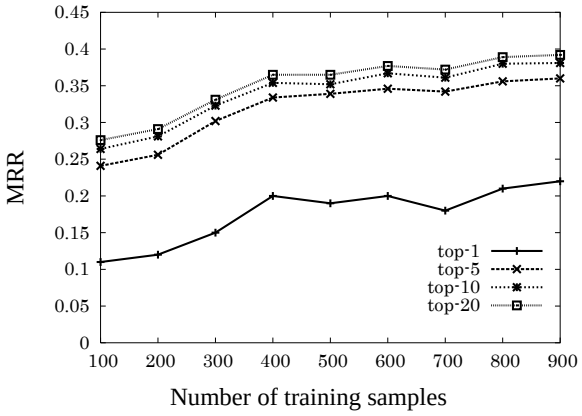

Figure 3: Learning curve: Performance changes when answering Q1–Q100 with different sizes of training samples. Paragraphs are used as answer candidates.

their quality itself may be to blame.

Furthermore, analyzing the trained ranking models allows us to calculate the weights given to the features (Hirao et al., 2002). Table 3 shows the weights of the top-10 features. We also include in the table the weights of the Synonym Pair, MAN-Causal Expression and Cause Effect Pair features so that the role of all three types of features in our approach can be shown. The analyzed model was the one trained with all 1,000 questions in the WHYQA collection with paragraphs as answers. Just as suggested by Table 2, the Question-Candidate Cosine Similarity feature plays a key role, followed by automatically collected causal expression features.

Figure 2 shows the distribution of the ranks of the first correct answers for all questions in the WHYQA collection for COS, FK, and NAZEQA. The distribution of COS is almost uniform, indicating that lexical similarity cannot be directly translated into causality. The figure also shows that NAZEQA consistently outperforms FK.

It may be useful to know how much training data is needed to train a ranker. We therefore fixed the test set to Q1–Q100 in the WHYQA collection and trained rankers with nine different sizes of training data (100–900) created from Q101–$\{$Q200 $\cdots$ Q1000$\}$. Figure 3 shows the learning curve. Naturally, the performance improves as we increase the data. However, the performance gains begin to decrease relatively early, possibly indicating the limitation of our approach. Since our approach heavily relies on surface patterns, the use of syntactic and semantic features may be necessary.

6 Summary and Future Work

This paper proposed corpus-based QA for why-questions. We automatically collected causal expressions from semantically tagged corpora and used them to create features to train an answer candidate ranker that maximizes the QA performance with regards to the corpus of why-questions and answers. The implemented system NAZEQA outperformed baselines with an MRR (top-5) of 0.305 and the coverage was also high, making NAZEQA presumably the best-performing system as a fully implemented why-QA system.

As future work, we are planning to investigate other features that may be useful for why-QA. We also need to examine how QA performance and the weights of the features differ when we use other sources for answer retrieval. In this work, we focused only on the '`cause`' relation in the EDR corpus to obtain causal expressions. However, there are other relations, such as '`purpose`,' that may also be related to causality (Verberne, 2006).

Although we believe our approach is language-independent, it would be worth verifying it by creating an English version of NAZEQA based on causal expressions that can be derived from PropBank and FrameNet. Finally, we are planning to make public some of the WHYQA collection at the authors' webpage so that various why-QA systems can be compared.

Acknowledgments

We thank Jun Suzuki, Kohji Dohsaka, Masaaki Nagata, and all members of the Knowledge Processing

Research Group for helpful discussions and comments. We also thank the anonymous reviewers for their valuable suggestions.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proc. COLING-ACL*, pages 86–90.

Robin Burke, Kristian Hammond, Vladimir Kulyukin, Steve Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the FAQFinder system. *AI Magazine*, 18(2):57–66.

Du-Seong Chang and Key-Sun Choi. 2004. Causal relation extraction using cue phrase and lexical pair probabilities. In *Proc. IJCNLP*, pages 61–70.

Jon Curtis, Gavin Matthews, and David Baxter. 2005. On the effective use of Cyc in a question answering system. In *Proc. IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, pages 61–70.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. 2004. Question answering challenge for five ranked answers and list answers – overview of NTCIR4 QAC2 subtask 1 and 2 –. In *Proc. NTCIR*, pages 283–290.

Jun'ichi Fukumoto, Tsuneaki Kato, Fumito Masui, and Tsunenori Mori. 2007. An overview of the 4th question answering challenge (QAC-4) at NTCIR workshop 6. In *Proc. NTCIR*, pages 483–440.

Jun'ichi Fukumoto. 2007. Question answering system for non-factoid type questions and automatic evaluation based on BE method. In *Proc. NTCIR*, pages 441–447.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proc. ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 76–83.

Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proc. 19th COLING*, pages 342–348.

Takashi Inui and Manabu Okumura. 2005. Investigating the characteristics of causal relations in Japanese text. In *Proc. ACL 2005 Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*.

Hideki Isozaki. 2004. NTT's question answering system for NTCIR QAC2. In *Proc. NTCIR*, pages 326–332.

Hideki Isozaki. 2005. An analysis of a high-performance Japanese question answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(3):263–279.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142.

Christopher S. G. Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proc. 38th ACL*, pages 336–343.

W. Mann and S. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. In *Text*, volume 8, pages 243–281.

Daniel Marcu and Abdessamad Echihabi. 2002. In *Proc. 40th ACL*, pages 368–375.

Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Català. 2005. Semantic role labeling as sequential tagging. In *Proc. CoNLL*, pages 193–196.

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 *Proc. NTCIR*, pages 487–492.

Tatsunori Mori, Mitsuru Sato, Madoka Ishioroshi, Yugo Nishikawa, Shigenori Nakano, and Kei Kimura. 2007. A monolithic approach and a type-by-type approach for non-factoid question-answering – Yokohama National University at NTCIR-6 QAC –. In *Proc. NTCIR*, pages 469–476.

Martha Palmer. 2005. The proposition bank: An annotated corpus of semantic roles. *Comp. Ling.*, 31(1):71–106.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::Similarity - Measuring the Relatedness of Concepts. In *Proc. HLT-NAACL (Demonstration Papers)*, pages 38–41.

Hideki Shima and Teruko Mitamura. 2007. JAVELIN III: Answering non-factoid questions in Japanese. In *Proc. NTCIR*, pages 464–468.

Troy Smith, Thomas M. Repede, and Steven L. Lytinen. 2005. Determining the plausibility of answers to questions. In *Proc. AAAI Workshop on Inference for Textual Question Answering*, pages 52–58.

Radu Soricut and Eric Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Journal of Information Retrieval*, 9:191–206.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proc. SIGIR (Posters and Demonstrations)*, pages 735–736.

Suzan Verberne. 2006. Developing an approach for why-question answering. In *Proc. 11th European Chapter of ACL*, pages 39–46.

Suzan Verberne. 2007a. Evaluating answer extraction for why-QA using RST-annotated Wikipedia texts. In *Proc. 12th ESSLLI Student Session*, pages 255–266.

Suzan Verberne. 2007b. Paragraph retrieval for why-question answering. In *Proc. Doctoral Consortium Workshop at SIGIR-2007*, page 922.

Ellen M. Voorhees and Hoa Trang Dang. 2005. Overview of the TREC 2005 question answering track. In *Proc. TREC*.