Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions

Ryuichiro Higashinaka and Hideki Isozaki[†]
NTT Communication Science Laboratories, NTT Corporation

This paper describes our approach for answering why-questions that we initially introduced at NTCIR-6 QAC-4. The approach automatically acquires causal expression patterns from relation-annotated corpora by abstracting text spans annotated with a causal relation and by mining syntactic patterns that are useful for distinguishing sentences annotated with a causal relation from those annotated with other relations. We use these automatically acquired causal expression patterns to create features to represent answer candidates, and use these features together with other possible features related to causality 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 baselines with a Mean Reciprocal Rank (top-5) of 0.223 when sentences are used as answers and with a MRR (top-5) of 0.326 when paragraphs are used as answers, making it presumably the best-performing fully implemented why-QA system. Experimental results also verified the usefulness of the automatically acquired causal expression patterns.

Categories and Subject Descriptors: H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Question-answering (fact retrieval) systems*; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Natural Language Interfaces*; H.5.2 [**Information interfaces and presentation**]: User interfaces—*Natural Language*

General Terms: Languages, Human Factors

Additional Key Words and Phrases: Question answering, causal expression, relation-annotated corpus, pattern mining.

1. INTRODUCTION

Following the trend of non-factoid QA, we are seeing the emergence of work on why-QA; i.e., 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. Recently, NTCIR-6[1] Question

This paper is a modified and augmented version of our earlier reports [Higashinaka and Isozaki 2007; 2008].

[†] 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan. {rh,isozaki}@cslab.kecl.ntt.co.jp

[1] NTCIR stands for NII-NACSIS Test Collection for Information Retrieval Systems; http://research.nii.ac.jp/ntcir/ntcir-ws6/ws-en.html

Answering Challenge (QAC-4) was held to encourage the research and development of Japanese non-factoid QA systems, attracting 14 systems from eight groups.

In why-QA, it is important to accurately extract passages/phrases bearing causes as answer candidates. The most common approach for this is to use hand-crafted patterns that comprise typical cue phrases or POS-tag sequences related to causality. For example, [Fukumoto et al. 2007] utilizes a number of hand-crafted patterns to extract cause-bearing passages. Such patterns include *tame* and *node*, which are typical cue words in Japanese corresponding to "because" in English, as well as words and phrases expressing causality, such as *genin* (cause) and *shiin* (cause of death). In fact, almost all systems presented at QAC-4 relied on such hand-crafted patterns for answer extraction.

One exception was our system, which used causal expression patterns automatically extracted from corpora. We considered this approach to be necessary because, as noted in [Inui and Okumura 2005], causes are expressed in various forms, which makes it difficult to cover all causal expressions by hand. Hand-crafting is also very costly if we want to increase the coverage of causal expressions.

This paper describes the approach we introduced at QAC-4. The approach automatically acquires causal expressions from relation-annotated corpora to derive causal expression patterns in order to improve why-QA. We also utilize a machine learning technique to train an answer-candidate ranker on the basis of the features created from the causal expression patterns together with other possible features related to causality so that the QA performance can be maximized with regards to a corpus of why-questions and answers.

This paper is organized as follows: Section 2 describes previous work on why-QA, and Section 3 describes our approach. Section 4 describes the actual procedure for automatically acquiring causal expression patterns from relation-annotated corpora. Section 5 describes the implementation of our approach, and Section 6 presents evaluation results. Section 7 presents a detailed analysis of the results by examining instances of successful and unsuccessful cases. Section 8 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. 2000], or 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 even if we include those at QAC-4. 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; Shima and Mitamura 2007]. We argue in this paper that this hand-crafting effort can be reduced using automatic methods.

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 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

We propose automatically acquiring causal expression patterns in order to reduce the hand-crafting effort that is currently necessary. We first collect causal expressions from corpora and convert them into causal expression patterns. We use these patterns to create features to represent answer candidates. 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 whose incorporation in the training improves 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. Although pinpointing exact answers may be desirable, there has not been an agreed-upon answering unit for why-QA. In definition QA, which is also a type of non-factoid QA, concise phrases or information nuggets are extracted as answers [Dang and Lin 2007]; however, they may not be suitable because causes may be expressed differently from definitions. Therefore, in this study, we start with sentences/paragraphs, both of which are basic units in natural language. We also assume that a document retrieval module of a system returns top-N documents for a question on the basis of conventional information retrieval (IR) related metrics and regard all sentences/paragraphs extracted from them 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 features that encode how it complies with the three conditions. Here, the causal expression features are those based on the causal expression patterns we aim to acquire 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 how we extract causal expression patterns from corpora, the three types of features, the corpus creation, and the ranker

3.1 Extracting Causal Expression Patterns from Corpora

With the increasing attention paid to SRL, we currently have several corpora, such as PropBank [Palmer 2005] and FrameNet [Baker et al. 1998], that are tagged with semantic relations including a causal relation. We came up with two ways to automatically derive causal expression patterns from such corpora (See Section 4 for the instances of the patterns we derived).

One is to use text spans annotated with a causal relation. Since such text spans are guaranteed to be expressing a cause, they provide good instances of causal expressions. By abstracting such causal expressions, we can create causal expression patterns. For example, if we have a causal expression "for the suspicion of fraud" in a corpus, we can create a causal expression pattern such as `[for the NN of NN]` by converting it into a POS-tag sequence. Note that, in this example, the prepositions and the definite article were preserved in order to avoid making the pattern too generic.

The other is to create causal expression patterns by pattern mining techniques. In the relation-annotated corpora, there are sentences that are annotated with a causal relation as well as those that are not. We can use the patterns that frequently occur in the sentences annotated with a causal relation as our causal expression patterns. For example, if a POS-tag sequence such as `[for the NN of NN]` occurs frequently in sentences annotated with a causal relation to a significant degree, we can accept it as a causal expression pattern. This process corresponds to pattern mining, and, for this purpose, we could apply one of the existing pattern mining methods. Compared to the patterns based on abstracted text spans, the resulting patterns of this approach could offer better precision because counter examples (sentences without a causal relation) are taken into account in the mining process.

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

3.2 Features

3.2.1 *Causal Expression Features.* Having derived causal expression patterns, we create n binary features from n causal expression patterns with each feature representing whether an answer candidate matches each pattern. In addition, some why-QA systems may already possess some good hand-crafted patterns for the detection of causal expressions. Since there is no reason not to use them if we know they are useful for why-QA, we can create a binary feature indicating whether an answer candidate matches existing hand-crafted patterns.

3.2.2 *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.

As mentioned, we regard *all* sentences/paragraphs in the retrieved top documents as answer candidates because the existence of query terms in the answer candidates does not necessarily mean that they are solely eligible as answers. We want to include as our answer candidates sentences/paragraphs that may have implicit references to the content of the question by, for example, reference/anaphoric expressions. Therefore, we also need to be able to calculate the content similarity between a question and an answer candidate even if they do not share the same words.

When a question and an answer candidate concern the same topic, they are likely to be similar in content. 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.

3.2.3 *Causal Relation Features.* There are semantic lexicons where a semantic relation between concepts is indicated. For example, the EDR concept 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 expression in a pair 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

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

techniques such as [Marcu and Echihabi 2002; Girju 2003; Chang and Choi 2004].

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

3.4 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. ACQUIRING CAUSAL EXPRESSION PATTERNS

We used the EDR corpus as our relation-annotated corpus. The EDR corpus is a part of the EDR dictionary, which is a suite of corpora and dictionaries and includes the EDR corpus, the EDR concept dictionary (hierarchy and relation of word senses), and the EDR Japanese word dictionary (sense to word mappings). As far as we know, it is one of the most commonly used Japanese corpus annotated with semantic relations. We first briefly describe the EDR corpus and then describe how we derived our causal expression patterns.

4.1 The EDR Corpus

The EDR corpus is a collection of independent Japanese sentences taken from various sources, such as newspaper articles, magazines, and dictionary glosses. The corpus has a semantic representation for each sentence and this information can be used as a relation annotation. For example, the EDR corpus has the following sentence:[4]

(1)

Obon	*no*	*kiseikyaku*	*wo*	*hakobu*	*koukuubin*	*de*
Bon Festival	-GEN	homecoming guests	-ACC	carry	aircrafts	by
konzatsu suru	*manatsu*	*no*	*sora*	*de,*	*hiyarito suru*	*dekigoto*
crowd	midsummer	-GEN	sky	in	fearful	incident
ga	*okita.*					
-NOM	occur-PAST.					

A fearful incident occurred in the midsummer sky crowded with aircrafts carrying homecoming guests during the Bon Festival.

[4] GEN, ACC, NOM, and PAST indicate the genitive, accusative, and nominative cases, and the past tense, respectively.

```
[[main 10:konzatsu(crowd):0f1e00]
 [object 14:sora(sky):0ff656]
 [cause [[main 8:koukuubin(aircrafts):3bd65b]
         [modifier [[main 6:hakobu(carry): 1e85e6]
                    [object [[main 3-4:kiseikyaku(homecoming guests)]
                             [modifier 1:obon(Bon Festival):0e800f]]]]]]]]
```

Fig. 1. Example of a semantic representation in the EDR corpus. Symbols such as 3bd65b and 0f1e00 indicate word sense IDs (called concept IDs) and numbers before the words (e.g., 10 before konzatsu) indicate word positions in the sentence.

Figure 1 shows an excerpt of the meaning representation that the EDR corpus has for this sentence. The semantic representation has a tree structure. The leaf nodes are composed of words that have corresponding concept IDs in the EDR concept dictionary; that is, words that do not bear concept IDs, such as functional words, do not appear in the tree structure. Nodes headed by `object`, `cause`, and `modifier` indicate that they hold object, causal, and modifying relations to the `main` nodes in their siblings. For example, *sora* (sky) and *koukuubin* (aircrafts) have `object` and `cause` relations to *konzatsu* (crowd), meaning that the sky is crowded by reason of aircrafts.

4.2 Patterns by Abstracting Text Spans

From the semantic representation, it is possible to identify the text spans corresponding to the node that has a causal (`cause`) relation. To find the text span for the `cause` node in Fig. 1, we first extract word positions under that node; namely 1, 3–4, 6, and 8. By taking its minimum and maximum word positions, we can obtain a text span by concatenating words 1–8; namely, *Obon no kiseikyaku wo hakobu koukuubin.* Although text spans created this way may be sufficient, considering that *bunsetsu*[5] is a commonly used linguistic unit in Japanese language, we expand each end of the span to its bunsetsu boundaries. In this example, there is no need to expand the left end of the span since it is already at the beginning of the sentence, but there is a bunsetsu boundary one word after *koukuubin*; between *de* and *konzatsu.* Therefore, we expand the text span to include *de* to obtain *Obon no kiseikyaku wo hakobu koukuubin de* as our desired text span. To detect bunsetsu boundaries, we used Cabocha,[6] a Japanese dependency analyzer

In this way, we can reasonably extract text spans annotated with a causal relation although the structure of the EDR corpus is slightly different from those of PropBank and FrameNet. Out of all 207,802 sentences in the EDR corpus, there are 8,379 sentences annotated with a causal relation. Since 82 sentences required elements outside them to complete words under a causal relation node due to anaphora and ellipsis, we used the remaining 8,297 sentences to obtain the causal text spans. From the 8,297 sentences, we found 8,774 text spans, with each sentence producing one or more text spans.

To make the text spans into causal expression patterns, in a manner similar to the previous pattern-based approaches on QA [Ravichandran and Hovy 2002; Cui

[5] Bunsetsu is a Japanese linguistic unit consisting of one or more content words followed by zero or more functional words.

[6] http://chasen.org/˜taku/software/cabocha/

#	Causal Expression Pattern	Frequency
1	*de* (by)	985
2	*no* (-GEN) * *de* (by)	659
3	*ni* (-DAT)	328
4	*no* (-GEN)	233
5	*ga* (-NOM)	164
6	*niyoru* (because of)	160
7	*no* (-GEN) * *ni* (-DAT)	158
8	*wa* (-TOPIC)	135
9	*niyotte* (because of)	131
10	*no* (-GEN) * *niyotte* (because of)	88
11	*no* (-GEN) * *ga* (-NOM)	76
12	*kara* (from)	71
13	*no* (-GEN) * *no* (-GEN) * *de* (by)	64
14	*no* (-GEN) * *wa* (-TOPIC)	60
15	*wo* (-ACC)	60
16	*no* (-GEN) * *kara* (from)	53
17	*no* (-GEN) * *niyoru* (because of)	53
18	*niyori* (because of)	51
19	*tame* (because of)	49
20	*ga* (-NOM) * *de* (by)	41

Table I. Top-20 causal expression patterns created from `cause`-annotated text spans. DAT and TOPIC indicate the dative case and a topic marker.

et al. 2007], we abstracted them by leaving only the functional words (auxiliary verbs and case, aspect, tense markers) and replacing others with wild-cards '*'. We chose this abstraction because functional words indicate important grammatical functions in Japanese and because including content words such as nouns and verbs could jeopardize the generality of the patterns when considering the relatively small number of text spans found in the corpus. By this abstraction, we obtain a pattern [no * wo * de] from *Obon no kiseikyaku wo hakobu koukuubin de.*[7] We also used Cabocha to perform this abstraction. From the 8,774 text spans, we obtained 402 distinct causal expression patterns after filtering out those that occurred only once. We call the patterns acquired by abstracting text spans the **ATS** (abstracted text span) patterns.

Although we acknowledge that this abstraction is Japanese-dependent because it mainly utilizes the characteristics of Japanese, in which tenses, aspects, and modalities are expressed mainly with functional affixes, we believe we can reasonably perform a similar abstraction for other languages, for example, by making use of verbal inflections and auxiliary verbs in the case of English.

Table I shows the top-20 ATS patterns with their frequency. It is noticeable that the patterns that have *de* (by) are very frequent. Although we can also observe many typical causal cue words such as *niyoru*, *niyotte*, *kara*, and *tame* (all of which correspond to "because" in English), it is interesting that patterns that do not contain such cue words are also frequent; e.g., [*ni* (-DAT)], [*no* (-GEN) * *ni* (-DAT)] and [* *wa* (-TOPIC)], re-confirming the variety of causal expressions as pointed out in [Inui and Okumura 2005] and also indicating possible insufficiency

[7] Asterisks at both ends are omitted for conciseness.

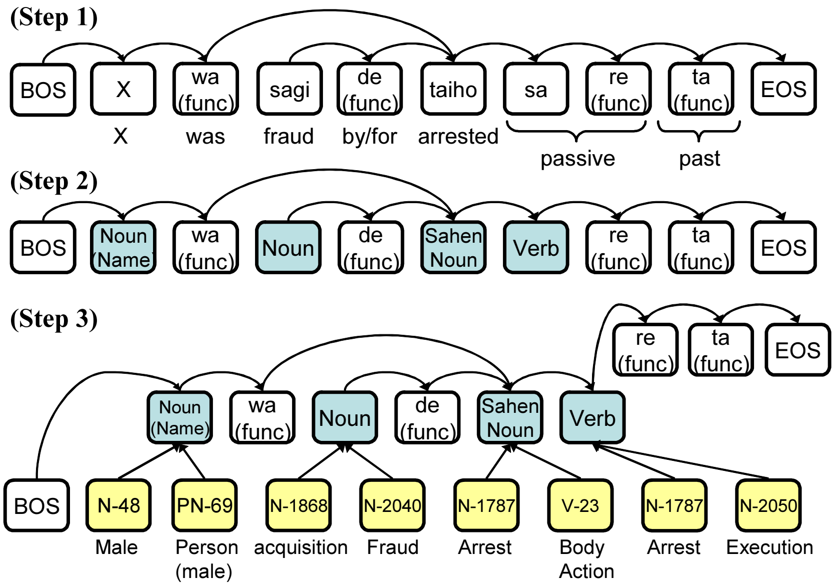

Fig. 2. The three steps for making a syntactic structure for the sentence "*X wa sagi de taiho sa re ta.* (X was arrested for fraud.)". BOS and EOS denote the beginning and the end of a sentence, respectively. 'Func' denotes that they are functional words.

of hand-crafting approaches.

4.3 Mining Syntactic Patterns by BACT

Since syntactic patterns have been found useful for extracting causal expressions [Khoo et al. 2000], we decided to mine syntactic patterns. We mine such patterns by finding syntactic structures that dominantly occur in the causally annotated sentences in the EDR corpus. There are 8,379 sentences that have causal relations and 199,423 sentences that do not.

For this purpose, we adopted BACT [Kudo and Matsumoto 2004], which is a machine learning algorithm based on a tree-mining technique. BACT mines, from positively or negatively labeled trees, subtrees that are useful for the classification of the trees using a boosting-based algorithm [Breiman 1999]. Since a syntactic structure has a tree representation, BACT can be directly applied. BACT has been used to mine syntactic patterns useful for text classification [Kudo and Matsumoto 2004] and anaphora resolution [Iida et al. 2006], in which syntactic information plays an important role.

Before the mining, as we did to the ATS patterns, we performed abstraction to the syntactic structures to improve generality. The abstraction was performed in the following three steps (See Fig. 2 for an example of how we make a syntactic structure for "X was arrested for fraud").

Step 1. Parse a sentence with Cabocha to create a word dependency tree. In this step, the leaves are the surface words (not base forms).

Step 2. Replace all words except functional ones with their POS tags. Functional

#	Causal Expression Pattern	Weight
1	*de* (by) General-Noun N-1398 [doubt] – *no* (-GEN)	0.0062
2	*niyori* (because of)	0.0031
3	, *yori ni* (because of)	0.0028
4	*niyotte* (because of)	0.0028
5	*node* (because of)	0.0026
6	*niyoru* (because of)	0.0021
7	*tame* (because)	0.0020
8	*aru de* (it is that . . .)	0.0020
9	。(Japanese punctuation mark)	0.0018
10	N-2455 [reason]	0.0015
11	*tame kono* (because of this)	0.0013
12	N-1265 [wonder/astonishment]	0.0013
13	*de* (by) Sahen-Noun	0.0013
14	N-2115 [shaking]	0.0012
15	Adjective *de* (by)	0.0012
16	N-2558 [activity]	0.0012
17	, *de* (by) *no* (-GEN)	0.0011
18	, *kara* (from)	0.0011
19	, *de* (by)	0.0010
20	Verb *wo* (-ACC) – , *wa* (-TOPIC) Sahen-Noun	0.0010

Table II. Top-20 syntactic patterns with their weights given by BACT. The meaning of semantic categories are shown in brackets. English translations are given in parentheses. The '–' denotes a sibling relation between nodes; otherwise, the relation is mother-daughter from left to right.

words are preserved as in the ATS patterns.

Step 3. Parse the sentence with *morph* [Ikehara et al. 1991] and JTAG [Fuchi and Takagi 1998] parsers. The *morph* is a morphological analyzer that comes with ALT/J-E (a Japanese-English machine translation system [Ikehara et al. 1991]) and outputs semantic categories (2,715 types) defined by Nihongo Goi-Taikei for most Japanese content words. JTAG is another morphological analyzer developed for information extraction and outputs verbal categories (36 types) and proper noun categories (130 types). Semantic categories are prefixed by N, verbal categories by V, and proper noun categories by PN. The nodes representing these categories are added as daughters to the POS-tag node. Nodes representing semantic categories are added to compensate for the semantic information lost in the abstraction in Step 2. Note that compared to the number of words, the number of semantic categories is much smaller. If no semantic categories can be assigned to the POS-tag node, we revert it back to a surface word node. In Japanese, the word boundaries of parsers may be different. Therefore, we used character-based matching to find out which category belongs to which POS-tag node.

After creating abstracted syntactic structures for all sentences and labeling them as positive or negative on the basis of whether they have a causal relation, we processed the structures with BACT, which mined useful subtrees and produced 669 syntactic patterns. We call these patterns the **BACT** patterns.

Table II shows the top-20 BACT patterns with their weights given by BACT. Note that the patterns are to be read from right to left following the structure of our dependency tree (See Fig. 2).

Similarly to the ATS patterns, we also observe a number of causal cue words in the top-10 together with many other patterns having POS-tags and semantic categories. The pattern with the heaviest weight has N-1398 [doubt] probably because of the many crime-related sentences in the EDR corpus, which is mainly composed of newspaper articles. We also see semantic categories such as N-1265 [wonder/astonishment], N-2115 [shaking] (e.g., earthquakes), N-2419 [types of illness], and N-2558 [activity] that we can intuitively recognize as sources of causes. Surprisingly, the Japanese punctuation mark (。) was found to be a useful indication of a cause (See pattern # 9).[8] This is probably because English-style punctuations were used in the sentences from dictionary glosses where causes are less likely to be expressed.

5. IMPLEMENTATION

We created a Japanese why-QA system that implements our approach. The system is called **NAZEQA** ("Naze" means "why" in Japanese). The system was built by extending our factoid QA system, SAIQA [Isozaki 2004; 2005]. The system 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. Query terms are also extracted from a question by removing from it functional words, such as auxiliary verbs and ending suffixes, and interrogative words. Canonicalization, such as base form conversion, is also applied to content words (i.e., nouns, verbs, and adjectives) using the dictionary that comes with ALT/J-E.

(2) Using the disjunction of query terms as a query, 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. The system can be configured to use sentences or paragraphs as answer candidates.

(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. The causal expression patterns (the ATS and BACT patterns) presented in Section 4 are utilized to create the causal expression features.

(4) The SVM ranker trained by a QA corpus ranks the answer candidates on the basis of the features.

(5) The top-N answer candidates are presented to the user as answers.

In the following sections, we show our list of features and describe the QA corpus and ranker.

5.1 Features

Causal Expression Features:

—AUTO-ATS-Causal Expression Features: We have 402 ATS patterns. Therefore, we create 402 binary features representing the existence of each pattern

[8] The Japanese punctuation mark appears in the table because semantic categories cannot be assigned to it (See Step 3 of the three steps to create our abstracted syntactic structures).

in the answer candidate. Currently, we do not take into account the frequency of the patterns in creating the features.

—AUTO-BACT-Causal Expression Features: We have 669 BACT patterns. Therefore, we create 669 binary features representing the existence of each pattern in the answer candidate. Currently, we do not utilize the weights given to the patterns in creating the features.

—MAN-Causal Expression Feature: 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.

Content Similarity Features:

—Question-Candidate Cosine Similarity Feature: 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.

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

—Synonym Pair Feature: This is a binary feature that indicates whether a word and its synonym (including the same word) 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 synonyms. We have 133,486 synonym entries.

Causal Relation Feature:

—Cause-Effect Pair Feature: 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 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. We have this large number of word pairs due to the multiplicity effects from synonyms on both sides and because of our augmenting word senses with its sub word senses (sub-concepts) using the hierarchical structure of the EDR concept dictionary.

5.2 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 the 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

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.

Fig. 3. 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.)

in the collection are guaranteed to have answers. Figure 3 shows a sample question and answer sentences in the collection.

5.3 Training a Ranker by Ranking SVM

We trained ranking models using the idea of ranking SVM [Joachims 2002]. The ranking SVM learns ranking by utilizing the preferences in the pairs of training samples. Suppose that an answer candidate $\boldsymbol{x}_i$ is ranked higher than $\boldsymbol{x}_j$. This preference can be represented by $\boldsymbol{w} \cdot \boldsymbol{x}_i > \boldsymbol{w} \cdot \boldsymbol{x}_j$, i.e., $\boldsymbol{w} \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j) > 0$, making it possible for the SVM to obtain $\boldsymbol{w}$ from training data $\{+1, \boldsymbol{x}_i - \boldsymbol{x}_j\}$. In the ranking phrase, input samples are simply ranked by their scores $\boldsymbol{w} \cdot \boldsymbol{x}$. Although SVM-light[9] is a widely used implementation for the ranking SVM, we implemented the equivalent using Pegasos [Shalev-Shwartz et al. 2007], which is an efficient linear kernel SVM training algorithm.

To create our training data, we first extracted all sentence and paragraph-level answers as well as non-answers from the WHYQA collection. Here, non-answers are the sentences/paragraphs that were not selected as answers by the analyst in the top-20 documents as described in Section 5.2. When we regard sentences as answers, there are 4,849 answers and 521,177 non-answers in the WHYQA collection. In the case of paragraphs, there are 4,371 answers and 261,215 non-answers.

Then, we extracted features for all answers and non-answers. We defined four feature sets; namely, NOAC, ATS, BACT, and ALL. The NOAC feature set does not use any features related to the automatically acquired causal expression patterns; namely, it uses the MAN-Causal Expression, Question-Candidate Cosine Similarity, Question-Document Relevance, Synonym Pair, and Cause-Effect Pair features. The ATS feature set uses all features without AUTO-BACT-Causal Expression features and the BACT feature set uses all features without AUTO-ATS-Causal Expression features. The ALL feature set uses all features. The NOAC, ATS, BACT, and ALL feature sets comprise 5, 407 (402 + 5), 674 (669 + 5), and 1076 (402 + 669 + 5) features, respectively.

[9] http://svmlight.joachims.org/

Having derived the features, we generated the set of difference vectors $\boldsymbol{x}_+ - \boldsymbol{x}_-$ for all pairs of answers $\boldsymbol{x}_+$ and non-answers $\boldsymbol{x}_-$ for each question in the WHYQA collection. Finally, using the difference vectors for all questions as training data, we trained our ranking models using Pegasos.

6. EVALUATION

For evaluation, we compared the proposed system (**NAZEQA**) with three baselines. We created three versions of NAZEQA (**NAZEQA-ATS**, **NAZEQA-BACT**, and **NAZEQA-ALL**) depending on the feature set used to train the ranker (the ranking model). The aim of having these three versions is to examine the effects of each type of automatically extracted causal expression patterns and also to examine whether the combination of the two types of patterns can lead to improvement.

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. Baseline-3 (**NOAC**) uses the ranking models trained by using the NOAC feature set. Since NOAC does not utilize the automatically extracted causal expression patterns, we do not regard it as a version of NAZEQA. The aim of having this baseline is to see how the system optimizes the ranking performance by the ranking SVM without the automatically extracted causal expression patterns.

NAZEQA and the three baselines used the same document retrieval engine to obtain the top-20 documents from Mainichi newspaper articles (1998–2001) and ranked the sentences or paragraphs in these documents.

6.1 QA Performance in MRR and Coverage

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 at least one of the top-N answer candidates. Tables III and IV show the MRRs and coverage for the baselines and NAZEQA. A 10-fold cross validation was used for the evaluation of NAZEQA and NOAC; that is, we first split the WHYQA collection into ten sets and then trained ranking models using nine of the ten sets and evaluate the performance using the remaining set; this was repeated ten times in a round-robin fashion.

We can see from the table that NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL are better in all comparisons to the baselines. A statistical test (a sign test that compares the number of times one system places the correct answer before the other) showed that they are significantly better ($p<0.01$) than all baselines for all top-Ns in the sentence and paragraph-levels. In addition, NOAC, which does not use the automatically extracted causal expression patterns, performs significantly worse than the NAZEQA systems, which do use them, showing the effectiveness of the automatically acquired causal expression patterns.

	Baselines			**NAZEQA**		
top-N	COS	FK	NOAC	ATS	BACT	ALL
Sentences as answer candidates						
top-1	0.036	0.091^{**}	0.071^{**}	0.123	**0.133**	**0.133**
top-5	0.086	0.138^{**}	0.141^{**}	0.206	0.222	$\mathbf{0.223}^{\ddagger\ddagger}$
top-10	0.102	0.148^{**}	0.161^{**}	0.228	$\mathbf{0.244}^{\ddagger}$	$0.243^{\ddagger}$
top-20	0.115	0.152	0.174^{**}_{++}	0.239	$\mathbf{0.255}^{\ddagger}$	$\mathbf{0.255}^{\ddagger\ddagger}$
Paragraphs as answer candidates						
top-1	0.065	0.152^{**}	$0.118^{**}_{\dagger}$	0.195	**0.206**	0.194
top-5	0.140	0.245^{**}	$0.225^{**}_{\dagger\dagger}$	0.316	**0.326**	0.318
top-10	0.166	0.257^{**}	0.249^{**}	0.339	**0.349**	0.340
top-20	0.181	0.262^{**}	0.262^{**}	0.350	**0.359**	0.350

Table III. Mean Reciprocal Rank (MRR) for the baselines (COS, FK, and NOAC) and the proposed NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL (ATS, BACT, and ALL in the table) systems for the entire WHYQA collection (1,000 questions). The top-1, 5, 10, and 20 mean the numbers of topmost candidates used to calculate the MRR. Although not marked for the sake of simplicity, NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL show statistical significance over all baselines (p<0.01). Asterisks indicate FK and NOAC's statistical significance (p<0.01) over COS, '++' NOAC's over FK (p<0.01), '†' FK's over NOAC (†† p<0.01, † p<0.05), and '‡' NAZEQA-BACT and NAZEQA-ALL's over NAZEQA-ATS (‡‡ p<0.01, ‡ p<0.05). **Bold font** indicates the current best performance.

	Baselines			**NAZEQA**		
top-N	COS	FK	NOAC	ATS	BACT	ALL
Sentences as answer candidates						
top-1	3.6%	9.1%	7.1%	12.3%	13.3%	13.3%
top-5	19.1%	23.1%	27.4%	35.6%	39.7%	39.4%
top-10	31.3%	30.7%	42.4%	52.0%	56.0%	54.6%
top-20	54.1%	35.5%	60.8%	67.9%	71.7%	71.5%
Paragraphs as answer candidates						
top-1	6.5%	15.2%	11.8%	19.5%	20.6%	19.4%
top-5	29.2%	41.6%	43.0%	54.1%	54.9%	54.0%
top-10	48.8%	50.5%	61.6%	71.4%	71.7%	70.7%
top-20	70.7%	56.4%	78.9%	85.2%	85.9%	85.2%

Table IV. Coverage for the baselines (COS, FK, and NOAC) and the proposed NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL (ATS, BACT, and ALL in the table) systems for the entire WHYQA collection. The top-1, 5, 10, and 20 mean the numbers of topmost candidates used to calculate the coverage.

When we compare NAZEQA-ATS and NAZEQA-BACT, NAZEQA-BACT performs better in all cases, showing the effectiveness of mining causal expression patterns. We see no remarkable difference between NAZEQA-BACT and NAZEQA-ALL. They are mostly tied in the sentence-level and NAZEQA-BACT leads slightly in the paragraph-level. It seems that using the ATS patterns in addition to the BACT patterns does not contribute greatly to the QA performance. Our analysis revealed that this limited performance of NAZEQA-BACT comes from its inability to assign appropriate scores to answer candidates with multiple pattern matches, because, in our current implementation, combinations of the patterns are not taken

	Baselines			**NAZEQA**		
top-N	COS	FK	NOAC	ATS	BACT	ALL
Sentences as answer candidates						
top-1	0.000	0.152	0.121	**0.273**	0.121	0.121
top-5	0.049	0.226^{+}	$0.191^{+}_{\dagger}$	**0.328**	0.214	0.212
top-10	0.068	0.242^{+}	$0.199_{\dagger}$	**0.348**	0.245	0.237
top-20	0.085	0.247^{+}	$0.218^{+}_{\dagger}$	**0.358**	0.256	0.245
Paragraphs as answer candidates						
top-1	0.061	$\mathbf{0.273}^{+}$	$0.091_{\dagger}$	0.182	0.182	0.091
top-5	0.125	$\mathbf{0.339}^{++}$	$0.185_{\dagger}$	0.303	0.238	0.205
top-10	0.144	$\mathbf{0.364}^{++}$	$0.208^{+}_{\dagger}$	0.320	0.268	0.248
top-20	0.155	$\mathbf{0.366}^{++}$	0.222^{++}	0.331	0.280	0.255

Table V. Mean Reciprocal Rank (MRR) for the baselines (COS, FK, and NOAC) and the proposed NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL (ATS, BACT, and ALL in the table) systems for the QAC-4 why-questions (33 questions). Pluses indicate FK and NOAC's statistical significance over COS (++ $p<0.01$, + $p<0.05$), and '†' FK's over NOAC ($p<0.05$). **Bold font** indicates the current best performance.

into account due to the limited expressiveness of the linear kernel used in our training of the ranking models using the ranking SVM. Polynomial kernels might solve this problem. The coverage is also high for NAZEQA, making it possible to find correct answers within the top-10 sentences and top-5 paragraphs for more than 50% of the questions.

Tables V and VI show the MRRs and coverage for the baselines and the NAZEQA systems when we evaluated them using the why-questions in the 100 questions of the QAC-4 formal run. We identified 33 questions as why-questions; namely, Q2, 5–8, 10, 12, 15, 26, 29–31, 35, 37, 40–41, 51, 53, 61–62, 64–65, 68, 70–71, 73, 79, 87, 90, 92, 96, and 98–99. We chose them as why-questions because two independent labelers, who are not the authors, agreed that they are asking for causes. The agreement ratio (Cohen's κ) was high with 0.91. The answers for the 33 questions were created in the same manner as the WHYQA collection by the same analyst. The ranking models used by NOAC, NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL are those trained using the entire WHYQA collection.

In the table, no significant difference was observed between the baselines and the NAZEQA systems due to the small number of questions; however, in terms of figures, NAZEQA-ATS shows the best performance in the sentence-level and FK performs best in the paragraph-level. It is also noticeable that NAZEQA-BACT does not perform as well as it does for the WHYQA collection. Since FK and NAZEQA-ATS, which rely mainly on surface patterns, show good performance compared to NAZEQA-BACT, which utilizes syntactic and semantic information, we suspect that the 33 questions in QAC-4 were those that can be answered by focusing on their surface expressions, especially causal cue words, rather than their syntactic structures or semantic information. This does not mean that NAZEQA-BACT cannot identify causal cue words. Remember that our automatically mined causal expression patterns include many such cue words (Table II). Our brief analysis of NAZEQA-BACT's answers revealed that, for the particular questions of QAC-4, NAZEQA-BACT finds many irrelevant matches with its complex patterns

	Baselines			**NAZEQA**		
top-N	COS	FK	NOAC	ATS	BACT	ALL
Sentences as answer candidates						
top-1	0.0%	15.2%	12.1%	27.3%	12.1%	12.1%
top-5	15.2%	36.4%	33.3%	42.4%	42.4%	42.4%
top-10	30.3%	48.5%	39.4%	57.6%	63.6%	60.6%
top-20	54.6%	54.5%	63.6%	72.7%	81.8%	72.7%
Paragraphs as answer candidates						
top-1	6.1%	27.3%	9.1%	18.2%	12.1%	9.1%
top-5	24.2%	48.5%	36.4%	51.5%	45.5%	39.4%
top-10	39.4%	66.7%	51.5%	63.6%	69.7%	72.7%
top-20	57.6%	69.7%	72.7%	78.8%	87.9%	81.8%

Table VI. Coverage for the baselines (COS, FK, and NOAC) and the proposed NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL (ATS, BACT, and ALL in the table) systems for the QAC-4 why-questions.

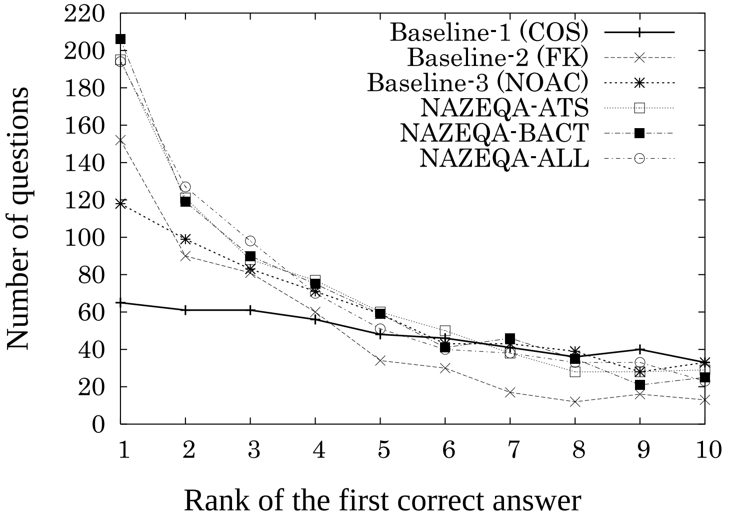

Fig. 4. Distribution of the ranks of first correct answers for all questions in the WHYQA collection. Paragraphs were used as answers. A 10-fold cross validation was used to evaluate NOAC, NAZEQA-ATS, NAZEQA-BACT, and NAZEQA-ALL.

to come up with answers, while the correct answer can be simply obtained by using a few explicit causal cue words.

Figure 4 shows the distribution of the ranks of the first correct answers for all questions in the WHYQA collection for the baselines and the NAZEQA systems. The distribution of COS is almost uniform, indicating that lexical similarity cannot be directly translated into causality. NOAC shows a similar tendency due to its heavy reliance on the content similarity features. The figure also shows that the NAZEQA systems consistently outperform FK. Among the NAZEQA systems, NAZEQA-BACT leads slightly in the number of top-1 answers.

6.2 Impact of the features

It is interesting to know how each type of feature contributes to the QA performance. Table VII shows how the performance of NAZEQA-ATS and NAZEQA-BACT in MRR (top-5) changes when one type of feature is excluded in the ranker training.

We have already mentioned the effectiveness of AUTO-ATS-Causal and AUTO-BACT-Causal Expression features in NAZEQA's comparison to NOAC in the previous section. Note that the performance without these features is the same as that of NOAC (See Table III). In addition, we see significant drops in performance when we remove the Question-Candidate Cosine Similarity and Document-Question Relevance features, showing the effectiveness of lexical and topic similarity.

The MAN-Causal Expression and Synonym 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 due 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.

It is difficult to interpret the effect of the Cause-Effect Pair feature. This is because, although the performance does not change when this feature is removed from NAZEQA-BACT, the performance seems to degrade in the sentence-level and vice versa in the paragraph-level when it is removed from NAZEQA-ATS. Our interpretation is that, although the Cause-Effect Pair feature is generally effective, overfitting to the training data occurred in the sentence-level; i.e., the comparative impact of the existence of a cause-effect pair is likely to be bigger for sentences than for paragraphs considering their short length.

To account for this ambivalence and the ineffectiveness of the Cause-Effect Pair feature when it is removed from NAZEQA-BACT, we also need to verify the quality of our cause-effect word pairs because we blindly expanded concepts holding a causal relation into corresponding words in creating the pairs, when the concepts have broad senses, their lexicalizations may not necessarily hold a causal relation. For example, we have "*satsujin* (murder)" and "*taiho suru* (arrest)" as a cause-effect word pair, but we also have "*keru* (kick)" and "*taiho suru* (arrest)". Here, *taiho suru* corresponds to a concept ID 3ce77a (an act of seizing a person who breaks the law; arrest) and *keru* 3cf10d (to refuse one's request). Although *keru* is one lexicalization of a refusal (e.g., "*hito no iken wo keru* (kick one's opinion)"), the word itself has other meanings (e.g., kicking a ball) and would not necessarily lead to an arrest.

Furthermore, we analyzed the trained ranking models to examine the weights given to the features by the ranking SVM. Table VIII shows the weights of the top-10 features. We also include in the table the weights of the MAN-Causal Expression and Cause-Effect Pair features so that the role of all types of features

	ATS		**BACT**	
Feature Set	Sent.	Para.	Sent.	Para.
All features	0.206	0.316	0.222	0.326
w/o AUTO-ATS-Causal Expression	**0.141****	**0.225****	N/A	N/A
w/o AUTO-BACT-Causal Expression	N/A	N/A	**0.141****	**0.225****
w/o MAN-Causal Expression	0.204	0.317	0.221	0.325
w/o Question-Candidate Cosine Similarity	**0.149****	**0.217****	**0.160****	**0.210****
w/o Document-Question Relevance	**0.182****	0.299	**0.193***	0.308
w/o Synonym Pair	0.201	0.310	0.216	0.328
w/o Cause-Effect Pair	*0.210*††	**0.309***	0.222	0.326

Table VII. Performance changes in MRR (top-5) when we exclude one type of feature. Asterisks indicate a statistically significant drop in performance (** $p<0.01$, * $p<0.05$) and '††' indicates a statistically significant improvement ($p<0.01$) from the case where all features are used. Ten-fold cross validation was used for the evaluation.

Rank	Feature Name	Weight
1	Question-Candidate Cosine Similarity	3.97
2	Document-Question Relevance	0.82
3	ATS-Exp.[*koto niyotte* (by the fact that)]	0.47
4	Synonym Pair	0.44
5	ATS-Exp.[*ga* (-NOM) * *no* (-GEN) * *wo* (-ACC) * *teiru* (-GERUND)]	0.31
6	ATS-Exp.[*wo* (-ACC)]	0.31
7	ATS-Exp.[*na* (AUX) *no* (-GEN) *ni* (-DAT)]	0.31
8	ATS-Exp.[*kara* (from) *ga* (-NOM) *ta* (-PAST)]	0.28
9	ATS-Exp.[*no* (-GEN)]	0.27
10	ATS-Exp.[*ga* (-NOM)]	0.26
	⋮	
43	MAN-Causal Expression	0.16
61	Cause-Effect Pair	0.13

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

in our approach can be seen. The analyzed model was the one trained trained with all 1,000 questions in the WHYQA collection with paragraphs as answers using the ATS feature set. Just as indicated in Table VII, the Question-Candidate Cosine Similarity feature plays the key role, followed by the Document-Question Relevance feature and the ATS patterns.

The same tendency was observed in the model trained using the BACT feature set (See Table IX). Here, the analyzed model was the one trained with all 1,000 questions in the WHYQA collection with sentences as answers. It is noticeable that many semantic categories, such as N-2329 [pollution], N-2522 [confusion], N-1246 [hunger and thirst], and N-2419 [types of illness], are included in the table. We also have semantic categories, such as N-1702 [invitation] (e.g., invoking of events), N-1301 [detest/dislike], N-1321 [respect/value/a high regard], and N-2265 [increase] in the top 20. Since they represent events that are likely to generate an effect, it strongly suggests the importance of having some prior knowledge about sources of causes for better why-QA.

Rank	Feature Name	Weight
1	Question-Candidate Cosine Similarity	3.52
2	Document-Question Relevance	0.83
3	BACT-Exp.[。 (Japanese punctuation mark)]	0.67
4	BACT-Exp.[N-2329 [pollution]]	0.48
5	BACT-Exp.[Verb N-2522 [confusion]]	0.44
6	BACT-Exp.[N-1246 [hunger and thirst]]	0.40
7	BACT-Exp.[N-2419 [types of illness]]	0.38
8	BACT-Exp.[N-1350 [shiver]]	0.36
9	Synonym Pair	0.36
10	BACT-Exp.[*de* (by) *mono* (being/thing) N-5 [human] – ,]	0.35
	⋮	
295	Cause-Effect Pair	0.04
355	MAN-Causal Expression	0.01

Table IX. Weights of features learned by the ranking SVM. 'AUTO-BACT-Causal Expression' is denoted as 'BACT-Exp.' for lack of space.

Although not listed in Table IX, we also have many highly ranked BACT patterns that have semantic categories together with functional words, such as BACT-Exp.[*de* (by) General-Noun N-2419 [types of illness]], BACT-Exp.[Verb N-1259 [pain/hardship] – *ni* (-DAT)], and BACT-Exp.[*ga* (-NOM) General-Noun N-2518 [situation/prospects]]. We consider that these semantic categories are used to disambiguate the usage of the functional words. For example, *de* is known to have more than ten usages (e.g., by, for, with, at, etc.) [Ishiwata 1999; Kiyota and Kurohashi 2001], and this ambiguity makes it difficult to decide when it is used for a causal relation. The BACT patterns seem to use the semantic categories around functional words to distinguish contexts in which *de* can be used as a causal cue.

6.3 Effects of Training Data Size on QA Performance

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 ··· Q1000}. Figure 5 shows the learning curve. We used the BACT feature set for the training, and sentences were used as answers. Naturally, the performance improves as we increase the data. However, the performance gains begin to decrease when the training data size exceeds 500, possibly indicating a limitation of our current implementation.

7. ANALYSIS OF ANSWERABLE QUESTIONS

Although it has been shown that the NAZEQA systems consistently outperform the baselines for the 1,000 questions in the WHYQA collection, when we evaluated them using the 33 why-questions of QAC-4 in the paragraph-level answers, FK showed better performance than the NAZEQA systems. Although this difference was not statistically significant, examining these cases closely could lead to further improvement of our approach.

Figure 6 is a Venn diagram showing the question IDs correctly answered by the

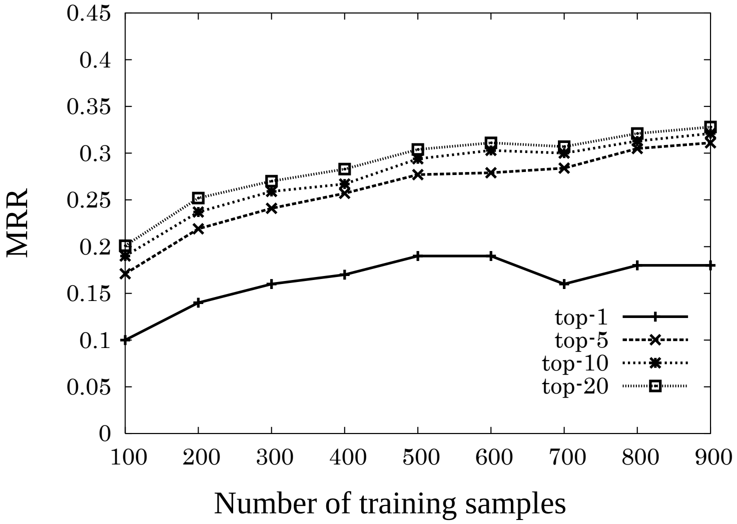

Fig. 5. Learning curve: Performance changes when answering Q1–Q100 with different sizes of training samples. Sentences are used as answer candidates. The BACT feature set was used for the training of the ranking models.

top-1 sentence-level answers by FK, NAZEQA-ATS, and NAZEQA-BACT. Q7 was answered correctly by all systems, indicating that it was probably the easiest why-question in QAC-4. In our analysis, we look in detail at Q2 where only FK succeeds and Q92 where only NAZEQA-BACT succeeds.

Figure 7 shows the top-1 sentence-level answers given by FK, NAZEQA-ATS, and NAZEQA-BACT for Q2. Contributions of the features identified by the ranking SVM are also shown for NAZEQA-ATS and NAZEQA-BACT. In this question, FK used the relatively high cosine similarity of 0.424 and the existence of a cue word *tame* to come up with the correct answer, whereas NAZEQA-ATS and NAZEQA-BACT chose answer candidates with a similar level of cosine similarity, but with the many matching ATS or BACT patterns. Since many automatically derived patterns are observed dominantly as their contributing factors, we consider that the weights for them might have been over-tuned to the corpus, causing an adverse effect. For example, "。", the Japanese punctuation mark, shows a significant contribution. This is mainly because the headlines of newspaper articles, which do not end with the Japanese punctuation mark, were seldom selected as answers in the QA corpus, showing heavy reliance on the training data. It is possible that the 1,000 questions may not be enough for training ranking models or that more abstraction of the patterns, such as generalizing over stylistic variations, may be necessary to suppress this overfitting.

Figure 8 shows the top-1 sentence-level answers by FK, NAZEQA-ATS, and NAZEQA-BACT for Q92. Here, although the cosine similarity is low with 0.137, NAZEQA-BACT came up with the correct answer on the basis of many matches with the BACT patterns, including the one with a semantic category corresponding to a cause (N-2450 [cause]). This semantic category came from a word "*genin*

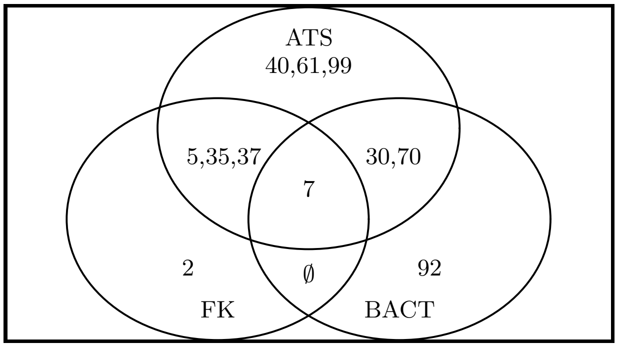

Fig. 6. Venn diagram showing the question IDs correctly answered by top-5 sentence-level answers by FK, NAZEQA-ATS, and NAZEQA-BACT for QAC-4 why-questions.

(cause)" in the candidate that strongly expresses a cause. Although FK uses a list of causal words for answer extraction and the list contains "*genin*", it could not come up with this candidate because the patterns of FK select candidates where such cue words appear only after specific functional words such as *ni, ga, wo, toiu*, and *toitta*. By restricting the context of causal words, FK aims to extract answer candidates with high precision, but seemingly at the cost of recall. This re-confirms the difficulty of cyclopaedically covering all causal expressions by hand.

Figure 9 is a Venn diagram showing the number of questions that can be answered by the top-5 sentence and paragraph-level answers by FK, NAZEQA-ATS, and NAZEQA-BACT for all questions in the WHYQA collection. From the good number of questions that can only be answered by NAZEQA-ATS and NAZEQA-BACT, the effectiveness of our approach can be seen. It is also interesting to see a big overlap between NAZEQA-ATS and NAZEQA-BACT compared to their small overlaps with FK, showing that NAZEQA-ATS and NAZEQA-BACT have similar ranking models and that such models greatly differ from a ranking process conceived and implemented by humans.

8. SUMMARY AND FUTURE WORK

This paper described our approach for why-QA, which we initially introduced at QAC-4. We automatically obtained causal expression patterns from relation-annotated corpora by abstracting text spans that are annotated with a causal relation and also by mining syntactic patterns that are useful in distinguishing sentences annotated with a causal relation from those annotated with other relations.

We used these automatically acquired patterns to derive features for answer candidates, and used these features together with other possible features related to causality 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 baselines with a MRR (top-5) of 0.223 when sentences are used as answers and with a MRR (top-5) of 0.326 when paragraphs are used as answers, making it presumably the best-performing fully implemented why-QA system. The usefulness of the automatically acquired causal expression patterns was also verified.

QAC4-00002-00: *Douiu mokuteki de NPO hou wa tsuku rare mashita ka?* (For what purpose was the NPO law made?)

Top-1 answer sentence by FK <**correct answer**>.

Shimin katsudou no kiban seibi no tame, sakunen 12 gatsu ni shikou sareta NPO hou. (In order to maintain the infrastructure of civic activity, the NPO law was passed last December.) [Cosine similarity: 0.424]

Top-1 answer sentence by NAZEQA-ATS.

"Undou wa oowaku kara kimete saibu ni iku toki ni ashimoto wo sukuware yasui. Tokuni houritsu wa, risouteki na mokuteki wo kakagete mo honenuki ni sa reru. Hossoku mae ni hou no saibu made tsukutte shimaou to omotta" to Matsubara san. ("In a campaign, it is easy for a project to fail when deciding from the outline and going into the details. Especially, laws are likely to be watered down even if they have ideal purposes. I thought that I needed to fix the details before the campaign." said Mr. Matsubara.)

#	Contribution	Feature
1	1.528	Question-Candidate Cosine Similarity (0.430)
2	0.430	Synonym Pair ("*hou* (law)" is shared.)
3	0.419	Document-Question Relevance (document rank=2)
4	0.350	ATS-Exp.[*na* (AUX)]
5	0.295	ATS-Exp.[*wa* (-TOPIC) * *na* (AUX)]
6	0.293	ATS-Exp.[*wo* (-ACC)]
7	0.212	ATS-Exp.[*no* (-GEN)]

Top-1 answer sentence by NAZEQA-BACT.

NPO sentâ nado NPO no shien, kouryuu wo mokuteki to shita NPO mo tsugitsugi to tsukurare teiru shi "shimin katsudou wo sasaeru seido wo tsukuru kai" nado wa, shakaiteki ni mitome rareru NPO no kaikei gensoku ya teikan manyuaru wo jibun tachi no te de tsukurou to shiteiru. NPOs for supporting and enhancing inter-communication among NPOs, such as the NPO center, have been increasingly organized, and the "Coalition for Legislation to Support Citizens' Organizations" is currently aiming to create, by itself, the accounting principle and an manual for articles of incorporation that are socially acceptable.

#	Contribution	Feature
1	1.940	Question-Candidate Cosine Similarity (0.552)
2	0.676	BACT-Exp.[。 (Japanese punctuation mark)]
3	0.355	Synonym Pair ("*tsukuru* (to make)" is shared.)
4	0.290	BACT-Exp.[General-Noun]
5	0.224	BACT-Exp.[Verb]
6	0.199	BACT-Exp.[*wo* (-ACC) General-Noun General-Noun]
7	0.165	BACT-Exp.[*ta* (-PAST)]

Fig. 7. Top-1 answers by FK, NAZEQA-ATS, and NAZEQA-BACT for Q2 in QAC-4. Here, FK's answer is the correct one. Contributions of the features used are shown for NAZEQA-ATS and NAZEQA-BACT. English translations by the authors are shown in parentheses.

As future work, we are planning to improve our features and also to investigate other possible features that may be useful for why-QA. For example, lexical similarity can be more accurately calculated by incorporating IDF values of the words

QAC4-00092-00: *2000-2001 nen no ariake nori fusaku no genin wa nan nano desuka?* (What was the reason for the bad harvest of Ariake seaweed from 2000 to 2001?)

Top-1 answer sentence by FK.

> *Daisanshai dewa nougyou keizai gaku wo senmon to suru fukusuu no iin kara "hitotsu no moderu to shi teiru 20 hekutâru no daikibo nougyou dewa, tochidai dake de 1 oku 4000 man en kakaru. Sonna kingaku wo haratte doredake nyuushoku kibou sha ga irunoka" nado no gimon ga deta.* (Several committee members of the third-party panel, who specialize in agricultural economics, expressed doubts saying "In one large-scale farming of 20 hectares, which we assume to be a model, it costs 140 million yen just for the land fee. Would there be any immigration applicants who can afford that amount of money?") [Cosine similarity: 0.547]

Top-1 answer sentence by NAZEQA-ATS.

> *Soko ni sumu gokai ya chigai nado no seibutsu wa, yaku 30 nen mae niwa 1 heihou mêtoru atari 2000-10,000 ita ga, konkai chousa dewa suuhyaku kotai shika mitsukara nai basho ga 8 chiten atta.* (Although the number of living things at the bottom of the sea such as sand worms and the fry of shellfish was 2000-10,000 per square meter about 30 years ago, this investigation revealed that there are eight points where only hundreds of individuals are found.)

#	Contribution	Feature
1	2.207	Question-Candidate Cosine Similarity (0.622)
2	0.430	Synonym Pair ("*nen* (year)" is shared.)
3	0.283	ATS-Exp.[*ga* (-NOM)]
4	0.271	ATS-Exp.[*ya* (or)]
5	0.246	ATS-Exp.[*nai* (not)]
6	0.232	ATS-Exp.[*nado* (such as) *no* (-GEN)]
7	0.211	ATS-Exp.[*wa* (-TOPIC)]

Top-1 answer sentence by NAZEQA-BACT <**correct answer**>.

> *Purankuton mo nori mo kaichuu no eiyouso (omo ni chisso ya rin no eiyoubun) wo totte ikite iru ga, eiyou wo suikomu chikara ga tsuyoi purankuton no tairyou hassei de, nori no eiyou ga tarinaku nari, iroochi ya fusaku no genin ni natta to mirare teiru.* (Although plankton and the seaweed live on nourishment in the sea (mainly, nutrient content of nitrogen and phosphorus), because of the mass generation of plankton with strong power to intake nourishment, the seaweed became nutritionally deprived, resulting in the color fade and bad harvest.)

#	Contribution	Feature
1	0.831	Document-Question Relevance (document rank=1)
2	0.676	BACT-Exp.[。 (Japanese punctuation mark)]
3	0.482	Question-Candidate Cosine Similarity (0.137)
4	0.355	Synonym Pair ("*nori* (sea weed)" is shared)
5	0.290	BACT-Exp.[General-Noun]
6	0.259	BACT-Exp.[N-2450 [cause]]
7	0.224	BACT-Exp.[Verb]

Fig. 8. Top-1 answers by FK, NAZEQA-ATS, and NAZEQA-BACT for Q92 in QAC-4. Here, NAZEQA-BACT's answer is the correct one.

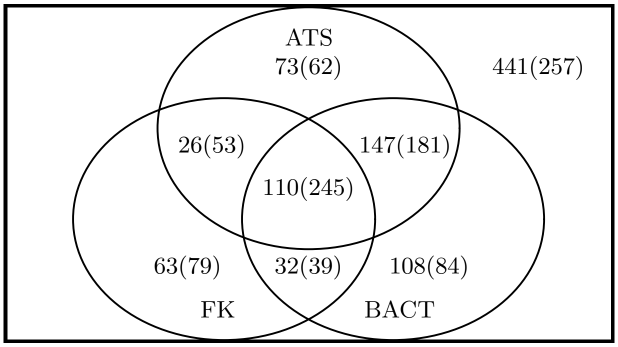

Fig. 9. Venn diagram showing the number of questions correctly answered by top-5 sentence and paragraph-level answers by FK, NAZEQA-ATS, and NAZEQA-BACT for the entire WHYQA Collection. The number of paragraph-level answers are shown in the parentheses.

as well as N-gram overlaps. We also need to investigate the quality of our synonyms and cause-effect word pairs because their usefulness was found to be limited in our analyses. 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 also be worth verifying it by creating an English version of NAZEQA based on causal expression patterns 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 comments and discussions. We also thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

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

Breiman, L. 1999. Prediction games and arching algorithms. *Neural Computation 11,* 7, 1493–1518.

Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N., and Schoenberg, S. 1997. Question answering from frequently asked question files: Experiences with the FAQFinder system. *AI Magazine 18,* 2, 57–66.

Chang, D.-S. and Choi, K.-S. 2004. Causal relation extraction using cue phrase and lexical pair probabilities. In *Proc. IJCNLP*. 61–70.

Cui, H., Kan, M.-Y., and Chua, T.-S. 2007. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems 25,* 2, 8.

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

Dang, H. T. and Lin, J. 2007. Different structures for evaluating answers to complex questions: Pyramids won't topple, and neither will human assessors. In *Proc. ACL*. 768–775.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research 4*, 933–969.

Fuchi, T. and Takagi, S. 1998. Japanese morphological analyzer using word co-occurrence –JTAG–. In *Proc. 17th COLING and 36th ACL*. Vol. 1. 409–413.

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

Fukumoto, J., Kato, T., Masui, F., and Mori, T. 2007. An overview of the 4th question answering challenge (QAC-4) at NTCIR workshop 6. In *Proc. NTCIR*. 483–440.

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

Higashinaka, R. and Isozaki, H. 2007. NTT's question answering system for NTCIR-6 QAC-4. In *Proc. NTCIR*. 460–463.

Higashinaka, R. and Isozaki, H. 2008. Corpus-based question answering for why-questions. In *Proc. IJCNLP*. Vol. 1. 418–425.

Iida, R., Inui, K., and Matsumoto, Y. 2006. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proc. 21st COLING and 44th ACL*. 625–632.

Ikehara, S., Shirai, S., Yokoo, A., and Nakaiwa, H. 1991. Toward an MT system without pre-editing –effects of new methods in ALT-J/E–. In *Proc. Third Machine Translation Summit: MT Summit III*. 101–106.

Inui, T. and Okumura, M. 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*.

Ishiwata, T. 1999. *Gendai Gengo Riron to Kaku (Case and Modern Linguistic Theories)*. Hitsuji Shobo. (in Japanese).

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

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

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

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

Kiyota, Y. and Kurohashi, S. 2001. Automatic summarization of Japanese sentences and its application to a WWW KWIC index. In *Proc. SAINT*. 120–127.

Kudo, T. and Matsumoto, Y. 2004. A boosting algorithm for classification of semi-structured text. In *Proc. EMNLP*. 301–308.

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

Marcu, D. and Echihabi, A. 2002. In *Proc. 40th ACL*. 368–375.

Màrquez, L., Comas, P., Giménez, J., and Català, N. 2005. Semantic role labeling as sequential tagging. In *Proc. CoNLL*. 193–196.

Mizuno, J., Akiba, T., Fujii, A., and Itou, K. 2007. Non-factoid question answering experiments at NTCIR-6: Towards answer type detection for realworld questions. In *Proc. NTCIR*. 487–492.

Mori, T., Sato, M., Ishioroshi, M., Nishikawa, Y., Nakano, S., and Kimura, K. 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*. 469–476.

Palmer, M. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics 31,* 1, 71–106.

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

Ravichandran, D. and Hovy, E. 2002. Learning surface patterns for a question answering system. In *Proc. 40th ACL*. 41–47.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. 2007. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. ICML*. 807–814.

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

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

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

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

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

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

Verberne, S., Boves, L., Oostdijk, N., and Coppen, P.-A. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proc. SIGIR (Posters and Demonstrations)*. 735–736.