Identification of relations between answers with global constraints for Community-based Question Answering services

Hikaru Yokono
Precision and Intelligence Laboratory,
Tokyo Institute of Technology
yokono@lr.pi.titech.ac.jp

Takaaki Hasegawa
NTT Cyber Space Laboratories,
NTT Corporation
hasegawa.takaaki@lab.ntt.co.jp

Genichiro Kikui*
NTT Cyber Space Laboratories,
NTT Corporation
kikui.genichiro@lab.ntt.co.jp

Manabu Okumura
Precision and Intelligence Laboratory,
Tokyo Institute of Technology
oku@pi.titech.ac.jp

Abstract

Community-based Question Answering services contain many threads consisting of a question and its answers. When there are many answers for a question, it is hard for a user to understand them all. To address this problem, we focus on logical relations between answers in a thread and present a model for identifying the relations between the answers. We consider that there are constraints among the relations, such as a transitive law, and that it might be useful to take these constraints into account. To consider these constraints, we propose the model based on a Markov logic network. We also introduce super-relations to give additional information for logical relation identification into our model. Through the experiment, we show that global constraints and super-relations make it easier to identify the relations.

1 Introduction

Community-based Question Answering services, such as Yahoo! Answers[1], OKWave[2] and Baidu Zhidao[3], have become popular web services. In these services, a user posts a question and other users answer it. The questioner chooses one of the answers as the best answer. These services have many threads consisting of one question and a number of answers, and the number of threads grows day by day. The threads are stored and anyone can read them. When a user has a question, if there is a similar question in the service, he or she can refer to the answers to the similar question. Herefrom, these services are useful for not only the questioner but also other users having a similar question.

When more answers get posted for a question, the answers in the thread might become more diverse. Some of these answers will be similar or oppositional to each other. Also, when a question tends to have various answers, e.g. the questioner asks for opinions (e.g. “What are your song recommendations?”), it is insufficient to read only the best answer. Generally, as the only one best answer is chosen from the answers, a user may miss other beneficial answers. When a user checks these services with a mobile device, its small display is inefficient to browse all answers.

To alleviate these problems, it would be useful to get an overview of answers in a thread, such as by identifying the relations between the answers or by summarizing them (Jimbo et al., 2010; Liu et al., 2008). The purpose of this study is to identify logical relations between answers with a high degree of accuracy, as a basis of these methods.

We propose an identification model with global constraints on logical relations between answers. Among the relations, there are some constraints like a transitive law. To this end, it is necessary to identify relations in a thread at once, and identified relations need to satisfy as many of these constraints as possible. Our model is based on a Markov logic network and incorporates these constraints as formulas of first order logic.

Also, we group logical relations on the basis of semantic similarity and transitivity and call these grouped relations “coarse relations” and “transi-

*Current affiliation is Faculty of Computer Science and System Engineering, Okayama Prefectural University.

[1] http://answers.yahoo.com/

[2] http://okwave.jp/

[3] http://zhidao.baidu.com/

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 920–928,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

tive relations", respectively. We consider that these relations might be useful for identification of logical relations and that identification of these relations is easier than that of logical relations. Thus, we incorporate identification of these super-relations into our model.

We briefly describe our related work in section two. Then, we show the logical relations between answers in section three and present our model with global constraints using a Markov logic network in section four. We explain the experiment and the results in section five and conclude our paper in section six.

2 Related Work

The growing popularity of Community-based Question Answering services has prompted many researchers to investigate their characteristics and to propose models for applications using them.

Question search and ranking answers are an important application because there are many threads in these services. Jeon et al. discussed a practical method for finding existing question and answer pairs in response to a newly submitted question (Jeon et al., 2005). Surdeanu et al. proposed an approach for ranking the answers retrieved by Yahoo! Answers (Surdeanu et al., 2008). Wang et al. proposed the ranking model for answers (Wang et al., 2009). Wang et al. proposed a model based on a deep belief network for the semantic relevance of question-answer pairs (Wang et al., 2010).

The user's qualifications affect the quality of his or her answer. For example, an IT expert may provide a good answer to a question about computers. Jurczyk and Agichtein proposed a model to estimate the authority of users as a means of identifying better answers (Jurczyk and Agichtein, 2007). Pal and Konstan proposed the expert identification model (Pal and Konstan, 2010) .

Each user has a background. If a user is an amateur in some field, he or she cannot understand a difficult question of the field. For a user-oriented question ranking, Chen and Kao proposed a model to classify a question as easy or difficult (Chen and Kao, 2010).

When there are many answers in a thread, multi-answer summarization is a good way to understand the answers. Liu et al. proposed taxonomies for a question and answers, and automatic summarization model for answers (Liu et al., 2008). Their best answer taxonomy is based on reusability for similar questions, factuality and so on, and their question type taxonomy is based on the expected answer. Achananuparp et al. proposed a model to extract a diverse set of answers (Achananuparp et al., 2010). Their approach is based on a graph whose edges have weight about similarity and redundancy.

Meanwhile, identification of discourse relations in meetings or dialogs was tackled by some researchers. Hillard et al. demonstrated that automatic identification of agreement and disagreement is feasible by using various textual, durational, and acoustic features (Hillard et al., 2003). Galley et al. described a statistical approach for modeling agreements and disagreements in conversational interaction, and classified utterances as agreement or disagreement by using the adjacency pairs and features that represent various pragmatic influences of previous agreements or disagreements to the target utterance (Galley et al., 2004).

Jimbo et al. proposed a model of relation identification for Community-based Question Answering services (Jimbo et al., 2010). Their model identified relations using Support Vector Machines with various features. We think considering constraints among relations might contribute to improve the performance of identifying relations. Therefore, we realize it with a Markov logic network.

Relation identification is considered as a problem to find labeled edges between pairs of nodes, where a node is an answer in a thread. Structured output learning is a method to predict such a structure (Tsochantaridis et al., 2004; Crammer et al., 2006). Morita et al. proposed a model based on structured output learning to identify agreement and disagreement relations in a discourse (Morita et al., 2009). Yang et al. used structured Support Vector Machines to extract contexts and answers for questions in threads of online forums (Yang et al., 2009).

3 Logical Relations between Answers

A thread consists of a question and some answers and the answers are sorted in order of posted time. Thus, in this paper, we try to identify to which preceding answer and in what relation an answer is related. However, some answers are irrelevant to a question and these answers might be unnecessary for an overview of a thread. Therefore, we consider not only answer-answer relations, but also question-answer relations. We consider two relations for question-answer pairs, and seven relations for answer-answer pairs.

3.1 Relations for Question-Answer Pairs

The relations for question-answer pairs are "answer" and "unrelated". The "answer" relation is that the answer answers the question directly and is beneficial for the questioner. The "unrelated" relation is that the answer has no relation with the expected answer for the question.

The reason why we consider the "unrelated" relation is that some answers are replies to other answers or questions to the original questioner to ask for further details.

3.2 Relations for Answer-Answer Pairs

We define the logical relations for answer-answer pairs according to Radev's work that defines 24 types of relations between texts for multi-document summarization (Radev, 2000). Table 1 shows the relations we consider.

Table 1: Logical relations between answers

Relation	Description
equivalence	Two answers have same contents.
elaboration	The content of the latter includes that of the former.
subsumption	Two answers do not have same contents directly, but they are related to each other.
summary	The content of the latter is a summary of the former.
partial	Two answers have partially-duplicated contents, and they also have mutually different contents.
contradiction	Two answers are completely inconsistent with each other.
unrelated	The contents of two answers have no relation. For example, two answers are different answers for the question.

Figure 1 shows an example of a thread.

Answers (a1) and (a2) include the same content, i.e. they recommend XXX, while answer (a3) expresses a different opinion. Answer (a4) mentions about XXX as well as answers (a1) and (a2), but contains the opposite opinion.

Hence, the relation between (a1) and (a2) is "equivalence" and the relations between (a1) and (a3) and between (a2) and (a3) are "unrelated". The relations between (a1) and (a4) and between (a2) and (a4) are "contradiction" and the relation between (a3) and (a4) is "unrelated".

Question (q). Can anyone recommend me a good internet provider?
Answer 1 (a1). XXX is good, though I use only this provider.
Answer 2 (a2). I prefer XXX. It is cheap and fast.
Answer 3 (a3). I use YYY and it is no problem.
Answer 4 (a4). The customer support of XXX is the worst.

Figure 1: Example of a QA thread

Here, since the relation between (a1) and (a2) is "equivalence" and the relation between (a1) and (a4) is "contradiction", we expect that the relation between (a2) and (a4) will be the same as the one between (a1) and (a4). This type of constraint is what we incorporate into the model.

4 Relation Identification Model with Global Constraints

We propose a joint identification model of logical relations between answers in a thread.

We consider that there are some constraints between logical relations. However, since not all relations satisfy a same constraint, we group logical relations into two types of super-relations on the basis of two kinds of commonality; transitivity and semantic similarity.

To incorporate constraints between relations, we try to identify relations for all pairs in a thread jointly. For these purposes, we take an approach with a Markov logic network.

4.1 Super-Relations for Answer-Answer Relations

We consider two kinds of super-relations for answer-answer relations; coarse relations and transitive relations. Coarse relations are based on semantic similarity and transitive relations are based on transitivity, that is, whether a transitive law is satisfied for relations.

Tables 2 and 3 show the correspondences between coarse relations and logical relations and between transitive relations and logical relations, respectively.

It might be easier to identify these two super-relations than logical relations, because these relations are coarser than logical relations. Furthermore, these information might be useful for identification of logical relations.

Table 2: Coarse relations

Relation	Logical relation
similar	equivalence, subsumption, elaboration, summary, partial
contradiction	contradiction
unrelated	unrelated

Table 3: Transitive relations

Relation	Logical relation
transitive	equivalence, subsumption, elaboration, summary
intransitive	contradiction, partial, unrelated

4.2 Markov Logic Network

A Markov logic network (MLN, in short) is a model combining first order logic and a Markov network (Richardson and Domingos, 2006). In this framework, we can take account of domain knowledge as formulas of first order logic. In first order logic, if there is at least one predicate that violates formulas in a possible world[4], the world is not valid. Therefore, a model with first order logic can only have strict constraints.

On the other hand, a MLN assigns a weight to each formula and can tolerate violation of the formula.

A MLN L is defined as a set of tuples $(\{F_i, w_i\})$, where F_i is a formula of first order logic and w_i is its weight. The distribution for a possible world x is as follows:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$$

where $n_i(x)$ is the number of F_i's ground formulas which are true on x, and Z is a normalization factor. A ground formula is a formula whose terms are constants.

The advantage of this model is that it can treat not only multiple relation identification tasks but also constraints between relations.

In this paper, according to the terminology of *markov thebeast*[5] (Riedel, 2008), which is one of the implementations of a MLN, we call the predicates that indicate aspects obtained from input data as **observed predicates**, the predicates that indicate aspects to be estimated as **hidden predicates**. The observed predicate corresponds to a feature in a general machine learning framework, and the hidden predicate corresponds to a label.

In addition, we call the formulas that express relations between observed predicates and a hidden predicate **local formulas** and formulas that express relations between hidden predicates **global formulas**.

4.3 Proposed Model

To identify the logical relation, our model consists of five subtasks: identification of question-answer relations, identification of whether two answers have a relation, identification of coarse relations, transitive relations, and logical relations. Table 4 shows the hidden predicates corresponding to these subtasks. For the question-answer relation, the relation between question i and answer j is "answer" if the predicate $hasqarelation(i, j)$ is true, and "unrelated" otherwise. For the answer-answer relation, the term l of the predicate $aarelation(i, j, l)$ corresponds to the logical relation to be identified originally.

Table 4: Hidden predicates

Predicate	Description
$hasqarelation(i, j)$	Answer j replies question i directly.
$hasaarelation(i, j)$	There is a relation between answer i and answer j.
$coarserelation(i, j, c)$	The coarse relation between i and j is c.
$transrelation(i, j, t)$	The transitive relation between i and j is t.
$aarelation(i, j, l)$	The logical relation between i and j is l.

In our model, when an answer is "unrelated" to the question, we exclude it from the identification of the answer-answer relation.

4.3.1 Local Formulas

In a MLN, there is only one hidden predicate in a local formula. We describe the relation between observed predicates and a hidden predicate on local formulas. For observed predicates, based on Jimbo et al.'s model (Jimbo et al., 2010), we consider thread features (e.g. order of answers and question type), n-gram features (e.g. word unigram and word bigram), semantic features, named-entity features, and similarity features. Table 5 lists some of the observed predicates. Since our model uses these features both for question-answer relations and answer-answer

[4] A possible world is a set of ground atoms. A ground atom is a predicate whose term is constant.

[5] http://code.google.com/p/thebeast/

relations, we use a term "article" for a question and an answer in this section.

Table 5: Some of the observed predicates

Predicate	Description
$question(i)$	Article i is a question.
$questiontype(q)$	The question type of the thread is q.
$first(i)$	Article i is the first answer in the thread.
$neighbor(i,j)$	Article j adjoins article i.
$longer(i,j)$	Article i is longer than article j.
$timegap(i,j,t)$	A time interval between article i and j is t.
$antonym(i,j)$	Article i and j contain antonyms.
$sameurl(i,j)$	Article i and j include a same URL.
$unigram(i,u)$	Article i contains word unigram u.
$bigram(i,b)$	Article i contains word bigram b.
$questionfocus(i)$	Article i contains the question-focus.
$samefocus(i,j)$	Article i and j contain the same question-focus.
$focusedneclass(i)$	Article i contains a word of the focused NE class.
$namedentity(i)$	Article i contains a named entity.
$samene(i,j)$	Article i and j contain the same named entities.
$samequoted(i,j)$	Article i and j contain the same quoted expression.
$scosine(i,j,c)$	Cosine similarity between article i and j in terms of sentences is c

For identification of relations between articles, the information obtained from the question, such as a class of an expected answer for a question, might be useful.

A question usually consists of multiple sentences such as in Figure 2. In this example, the essential question is the latter sentence. Therefore, we extract the core sentence and the question-focus and estimate the question type, on the basis of Tamura et al.'s model (Tamura et al., 2005).

Question.
I'm planning a journey to Hokkaido.
Can you suggest some good sightseeing places?

Figure 2: Example of a question

The core sentence is the most important sentence, that is, the one requiring an answer in the question. Usually, the core sentence is an interrogative one such as "Where is the most famous place in Hokkaido?". But questioners sometimes ask questions without an interrogative form, such as "Please tell me some good sightseeing places in Hokkaido.". To cope with this diversity, a machine learning approach is taken to extract the core sentence.

The question-focus is then extracted from the core sentence. The question-focus is the word that determines the answer class of the question. For example, the question-focus in Figure 2 is "places".

We extract the question-focus according to the following steps[6]:

step 1. Find the phrase including the last verb of the sentence or the phrase with "?" at the end.

step 2. Find the phrase that modifies the phrase found in step 1.

step 3. Output the nouns and the unknown words in the phrase found in step 2.

Question types are categorized in terms of the expected answer into the thirteen types in table 6.

Table 6: Question types

Nominal answer	Non-nominal answer
Person	Reason
Product	Way
Facility	Definition
Location	Description
Time	Opinion
Number	Other (text)
Other (noun)	

We use a model based on Support Vector Machines (SVM, in short) to identify the question type for the question. In this model, the feature vector is obtained from the core sentence.

The question types whose answers are nouns require a specific class of named entity for the answer, e.g. the class of named entity, PERSON for the question type, Person. We call this class **focused NE class**. Table 7 shows the correspondence between a question type and a named entity class.

Table 7: Question type and focused NE class

Question type	Focused NE class
Person	PERSON
Product	ARTIFACT
Facility	LOCATION, ORGANIZATION
Location	LOCATION
Time	DATE, TIME
Number	MONEY, PERCENT

For n-gram features, we also consider unigram and bigram for the first five words and the last five

[6] This procedure is specific to Japanese.

words in each article. For similarity features, we also consider cosine similarities in terms of word unigram, word bigram, phrase unigram, noun unigram, and noun category unigram.

The score c for sentence-based cosine similarity is calculated as follows:

$$c = \max_{s_{i_m} \in S_i, s_{j_n} \in S_j} \mathrm{sim}_{\cos}(s_{i_m}, s_{j_n})$$

where s_{i_m} is a m-th sentence of article i and S_i is a set of sentences in article i. $\mathrm{sim}_{\cos}(x, y)$ means a cosine similarity between sentence x and y. For the predicate about similarity like $scosine(i, j, c)$, we do not use the score itself for c but rather a value from 0 to 1 divided up into tenths.

For t in the predicate $timegap(i, j, t)$, we choose the one from {"less than an hour", "an hour $\sim$ 3 hour", "3 hour $\sim$ 6 hour", "6 hour $\sim$ 12 hour", "12 hour $\sim$ 24 hour", "24 hour $\sim$ 48 hour", "48 hour $\sim$ 72 hour", "more than 72 hour"} for the actual time gap between article i and j.

We consider the same pattern of local formulas for each hidden predicate. Some examples of local formulas are:

$$question(i) \land first(j) \Rightarrow hasqarelation(i, j) \quad (1)$$

$$bigram(i, +w_1) \land bigram(j, +w_2) \Rightarrow hasaarelation(i, j) \quad (2)$$

$$samequoted(i, j) \Rightarrow coarserelation(i, j, +c) \quad (3)$$

$$scosine(i, j, +v) \Rightarrow transrelation(i, j, +t) \quad (4)$$

$$lastunigram(i, +u_1) \Rightarrow aarelation(i, j, +l) \quad (5)$$

where "+" indicates that the weight of the formula depends on each constant to be grounded. For example, formula (1) has one weight in spite of values for i and j, but formula (3) has a separate weight for each value of c.

4.3.2 Global Formulas

Global formulas have more than one hidden predicates, and we can use these formulas to incorporate constraints between hidden predicates into the model. Figure 3 shows some global formulas that we consider.

There are two kinds of formulas in a MLN: hard constraints and soft constraints. Hard constraints are formulas that must be satisfied in a possible world. This kind of constraint is realized by assigning a huge value for its weight. For a possible world where a formula of hard constraints is false, its probability is almost zero. In our model, formulas from (6) to (12) are hard constraints.

We describe preconditions for each hidden predicate (formulas (6)-(10)) and correspondences between super-relations and logical relations (formulas (11) and (12)) as hard constraints. For example, formula (10) represents that if there is any transitive relation between answer i and j, there needs to be any logical relation between the answers.

Soft constraints are formulas that are allowed to be false in a possible world. It is obvious that a possible world where soft constraints are satisfied is more probable than a world where soft constraints are not satisfied. Thus, the model identifies relations to satisfy as many of these constraints as possible. In our model, formulas from (13) to (16) are soft constraints. We describe soft constraints for relations among three answers i, j, and k.

Formula (13) represents a semantic relevancy and formula (14) represents a transitive law. As shown in the example in Figure 1, when a relation between two answers is "equivalence", it is reasonable to assume that logical relations from these answers to other answers are identical with each other. Formula (15) represents this situation, and formula (16) represents the opposite direction of formula (15), that is, relations from other answers to these answers.

5 Experiment

To evaluate our model, we conducted an experiment with annotated question-answers threads in Japanese.

5.1 Experimental Settings

We used 299 threads of three genres from Yahoo! Chiebukuro[7], which is a Japanese Community-based Question Answering service. Table 8 shows the statistics of the data we used.

Table 8: Statistics of the data

Genre	the number of threads	the number of answers	Average number of answers in a thread
Cook	99	776	7.83
PC	100	618	6.13
Love	100	813	8.13

For each question-answer pair and answer-answer pair, five annotators annotated a relation. In annotating relations, answers whose question-answer relation had been annotated with "unre-

[7] http://chiebukuro.yahoo.co.jp/

$$\neg hasqarelation(i,j) \Rightarrow \neg hasaarelation(j,k) \tag{6}$$

$$\neg hasqarelation(i,k) \Rightarrow \neg hasaarelation(j,k) \tag{7}$$

$$hasaarelation(i,j) \Rightarrow \exists t(transrelation(i,j,t)) \tag{8}$$

$$hasaarelation(i,j) \Rightarrow \exists c(coarserelation(i,j,c)) \tag{9}$$

$$transrelation(i,j,t) \Rightarrow \exists l(aarelation(i,j,l)) \tag{10}$$

$$\begin{aligned} coarserelation(i,j,\text{“similar”}) \Rightarrow\ & \neg(aarelation(i,j,\text{“contradiction”}) \lor \\ & aarelation(i,j,\text{“unrelated”})) \end{aligned} \tag{11}$$

$$\begin{aligned} transrelation(i,j,\text{“transitive”}) \Rightarrow\ & \neg(aarelation(i,j,\text{“partial”}) \lor \\ & aarelation(i,j,\text{“contradiction”}) \lor \\ & aarelation(i,j,\text{“unrelated”})) \end{aligned} \tag{12}$$

$$\begin{aligned} coarserelation(i,j,\text{“similar”}) \land\ & coarserelation(j,k,\text{“similar”}) \\ \Rightarrow\ & coarserelation(i,k,\text{“similar”}) \end{aligned} \tag{13}$$

$$\begin{aligned} transrelation(i,j,\text{“transitive”}) \land\ & transrelation(j,k,\text{“transitive”}) \\ \Rightarrow\ & transrelation(i,k,\text{“transitive”}) \end{aligned} \tag{14}$$

$$aarelation(i,j,\text{“equivalence”}) \Rightarrow aarelation(i,k,+l) \land aarelation(j,k,+l) \tag{15}$$

$$aarelation(j,k,\text{“equivalence”}) \Rightarrow aarelation(i,j,+l) \land aarelation(i,k,+l) \tag{16}$$

Figure 3: Some of the global formulas

lated” were excluded from the annotation of the answer-answer relation. We considered only relation labels that more than two annotators agreed on the experiment. The number of pairs that we used and the distributions of relations are shown in Tables 9 and 10, respectively.

Table 9: Number of pairs

	Cook	PC	Love
question-answer	775	616	811
answer-answer	2194	1012	1626

Table 10: Distribution of relations

	Relation	Cook	PC	Love
question-answer	answer	0.925	0.924	0.905
	unrelated	0.075	0.076	0.095
answer-answer	equivalence	0.115	0.186	0.192
	elaboration	0.026	0.083	0.033
	subsumption	0.000	0.009	0.008
	summary	0.012	0.033	0.025
	partial	0.073	0.078	0.113
	contradiction	0.055	0.084	0.187
	unrelated	0.716	0.528	0.442

To acquire the semantic category for nouns, we utilized a Japanese thesaurus, Nihongo-Goi-Taikei (Ikehara et al., 1997). For antonyms, we used the Japanese dictionary, Kadokawa-Ruigo-Shin-Jiten (Ohno and Hamanishi, 1981).

For dependency parsing, we used Japanese dependency parser CaboCha[8]. For named entities, we utilized CaboCha's output.

We used SVM^{light}[9] for the implementation of SVM and *markov thebeast* for a MLN.

5.2 Results

For each genre, we performed 10-fold cross validation and evaluated the F-value.

The baseline model was one using SVM based on (Jimbo et al., 2010). In this model, we used the observed predicates for our model as features and trained a binary classifier for the question-answer relation and a one-versus-rest classifier for the answer-answer relation. The algorithm for the baseline model is as follows:

step 1. Identify the question-answer relation between the question and each answer.

step 2. For answers to be identified as an “answer”, identify the answer-answer relation.

Our work is different from Jimbo et al.'s work with respect to the number of relations. We considers seven relations for answer-answer pairs, while they consider four relations. Also, we used a different data from their experiment. Therefore, we could not conduct an accurate comparison experiment between our model and their model.

Table 11 shows the results. Bold face indicates that F-value of our model was higher than the baseline model and symbols $^{**}(p < 0.01)$ and $^{*}(p < 0.05)$ indicate the F-value was significantly different from the baseline with a sign test. Compared with the baseline model, our model was better for most relations.

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

[9] http://www.cs.cornell.edu/people/tj/svm_light/

Table 11: Results for each relation (F-value)

	Cook		PC		Love	
Relation	SVM	MLN	SVM	MLN	SVM	MLN
QArelation	0.961	0.956*	0.959	0.958	0.949	0.945*
AArelation	0.793	**0.796****	0.470	**0.653****	0.326	**0.612****
Coarse (similar)	0.018	**0.246****	0.315	**0.326****	0.176	**0.266****
Coarse (contradiction)	0.000	**0.025**	0.000	**0.033**	0.000	**0.091****
Coarse (unrelated)	0.636	**0.642****	0.423	0.378**	0.301	**0.312****
Trans (transitive)	0.000	**0.094****	0.000	**0.309****	0.000	**0.120****
Trans (intransitive)	0.712	0.712	0.498	**0.506****	0.497	0.495
Logical (equivalence)	0.000	**0.062****	0.164	**0.245****	0.019	**0.116****
Logical (elaboration)	0.000	0.000	0.000	**0.022**	0.000	0.000
Logical (subsmption)	0.000	0.000	0.000	0.000	0.000	0.000
Logical (summary)	0.000	0.000	0.000	**0.014**	0.000	0.000
Logical (partial)	0.000	**0.141****	0.000	**0.098****	0.000	**0.102****
Logical (contradiction)	0.000	**0.025**	0.000	**0.033**	0.000	**0.094****
Logical (unrelated)	0.636	**0.637****	0.406	0.380**	0.311	0.306**

While the baseline model considered only the target pair in identification, our model considers the relations of all pairs at the same time and identifies relations to satisfy as many constraints as possible. These constraints on relations contribute to improve the performance for the identification of relations.

Also, to evaluate the effectiveness of introducing the super-relations, we evaluated the model without coarse relations and transitive relations (w/o-super).

Table 12 shows the results for the data for PC. Bold face indicates the best value for each relation. For most relations, the w/o-super model was better than the baseline model. Although the w/o-super model does not leverage global formulas about super-relations (e.g. formula (13)), it leverages global formulas about logical relations (e.g. formula (16)) as well as the MLN model. We consider that the reason why the w/o-super model outperformed the SVM model is that these constraints worked well.

Table 12: Results for the model w/o super (F-value)

Relation	SVM	w/o-super	MLN
QArelation	**0.959**	0.952**	0.958
AArelation	0.470	0.546**	**0.653****
Coarse (similar)	0.315	–	**0.326****
Coarse (contradiction)	0.000	–	**0.033**
Coarse (unrelated)	**0.423**	–	0.378**
Trans (transitive)	0.000	–	**0.309****
Trans (intransitive)	0.498	–	**0.506****
Logical (equivalence)	0.164	**0.264****	0.245**
Logical (elaboration)	0.000	0.000	**0.022**
Logical (subsmption)	0.000	0.000	0.000
Logical (summary)	0.000	0.000	**0.014**
Logical (partial)	0.000	0.044	**0.098****
Logical (contradiction)	0.000	0.018	**0.033**
Logical (unrelated)	**0.406**	0.251**	0.380**

Furthermore, the MLN model is better than the w/o-super model. Because super-relations focus on transitivity and semantic similarity, identifying these relations is easier than the logical relations and the information about these relations is useful for identifying the logical relations.

In our model, there are some constraints between the predicate $aarelation$ and the other predicates. When the performances for the identifications of auxiliary relations (i.e. $hasaarelation$, $coarserelation$, $transrelation$) are worse, the performance of the identification of logical relations would be worse too. Therefore, in order to improve the performance of logical relation identification, it is necessary to improve the performance of identifying these auxiliary relations.

6 Conclusion

We proposed a logical relation identification model with a Markov logic network. There are constraints between relations and we incorporated them into the model. These constraints may be violated and a MLN permits violation of them.

Through the experiment, we showed that our model is better than a baseline model using SVM and that incorporating super-relations improves the performance. However, since the accuracy was not so high, we need to improve our model.

The relation between answers is the effective information for understanding the overview of a thread. Our future work is to propose answers summarization model and thread visualization model, based on these logical relations.

References

Palakorn Achananuparp, Xiaohua Hu, Tingting He, Christopher C. Yang, Yuan An, and Lifan Guo. 2010. Answer diversification for complex question answering on the web. In *Proceedings of The 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 375–382.

Ying-Liang Chen and Hung-Yu Kao. 2010. Finding hard questions by knowledge gap analysis in question answer communities. In *Proceedings of the 6th Asia Information Retrieval Societies Conference.*, pages 370–378.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Michel Galley, Kathleen McKeown, Julia Hirchberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of baysian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 669–676.

Dustin Hillard, Mari Ostendorf, and Elizabeth Shriberg. 2003. Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 34–36.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentarou Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Nihongo-Goi-Taikei*. Iwanami Shoten. (In Japanese).

J. Jeon, W.B. Croft, and J. Lee. 2005. Finding semantically similar questions based on their answers. In *Proceedings of the SIGIR'05*.

Kazuki Jimbo, Hiroya Takamura, and Manabu Okumura. 2010. Identification of relations between utterances in question answering communities. In *Proceedings of The 24th Annual Conference of the Japanese Society for Artificial Intelligence 3D3-3*. (In Japanese).

Pawel Jurczyk and Eugene Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of 16th ACM International Conference on Information and Knowledge Management*, pages 919–922.

Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. 2008. Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 497–504.

Hajime Morita, Hiroya Takamura, and Manabu Okumura. 2009. Structured output learning with polynomial kernel. In *Proceedings of the International Conference RANLP-2009*, pages 281–286.

Susumu Ohno and Masato Hamanishi. 1981. *Kadokawa-Ruigo-Shin-Jiten*. Kadokawa Shoten. (In Japanese).

Aditya Pal and Joseph A. Konstan. 2010. Expert identification in community question answering: Exploring question selection bias. In *Proceedings of 19th ACM International Conference on Information and Knowledge Management*, pages 1505–1508.

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of 1st SIGdial Workshop on Discourse and Dialogue*, pages 74–83.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of the 24th Annual Conference on Uncertainty in AI (UAI '08)*, pages 468–475.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *Proceedings of the ACL'08*, pages 719–727.

Akihiro Tamura, Hiroya Takamura, and Manabu Okumura. 2005. Classification of multiple-sentence questions. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, pages 426–437.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, pages 823–830.

Xin-Jing Wang, Xudong Tu, Dan Feng, and Lei Zhang. 2009. Ranking community answers by modeling question-answer relationships via analogical reasoning. In *Proceedings of the 32nd Annual ACM SIGIR Conference*, pages 179–186.

Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, and Lin Sun. 2010. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1230–1238.

Wen-Yun Yang, Yunbo Cao, and Chin-Yew Lin. 2009. A structural support vector method for extracting contexts and answers of questions from online forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 514–523.