Answering Yes/No Questions via Question Inversion

Hiroshi Kanayama[1] *Yusuke Miyao*[2] *John Prager*[3]

(1) IBM Research - Tokyo, Koto-ku, Tokyo, Japan
(2) National Institute of Informatics, Chiyoda-ku, Tokyo, Japan
(3) IBM T.J.Watson Research Center, Yorktown Heights, NY, USA
hkana@jp.ibm.com, yusuke@nii.ac.jp, jprager@us.ibm.com

Abstract

This paper investigates a solution to yes/no question answering, which can be mapped to the task of determining the correctness of a given proposition. Generally it is hard to obtain explicit evidence to conclude a proposition is *false* from an information source, so we convert this task to a set of factoid-style questions and use an existing question answering system as a subsystem. By aggregating the answers and confidence values from a factoid-style question answering system we can determine the correctness of the entire proposition or the substitutions that make the proposition false. We evaluated the system on multiple-choice questions from a university admission test on world history, and found it to be highly accurate.

Keywords: question answering, facts validation, yes/no question, question inversion.

Proceedings of COLING 2012: Technical Papers, pages 1377–1392,
COLING 2012, Mumbai, December 2012.

1 Introduction

Yes/no question answering (Green and Carberry, 1994; Hirschberg, 1984) can be equated to the task of determining the correctness of a given proposition. The target of such a mechanism is not limited to explicit interrogative questions since any general declarative sentences can in principle be validated. For example, consider the following two propositions, where (1) is true and (2) false[1]:

(1) Chirac was the president of France in 2000.

*(2) Chirac was the president of Germany in 2000.

As suggested by this example, a false proposition can often be produced by replacing a part of a true proposition. During a conversation, a human with knowledge of the facts might not only say that the utterance (2) is wrong, but respond with something like "not Germany, but France." Therefore we believe this kind of validation system we are proposing here may have application outside of a strict question-answering framework.

In spite of the importance of the process, yes/no-style question answering has not been intensively studied, compared to other type of question answering, such as the factoid-style question answering (Ravichandran and Hovy, 2002; Bian et al., 2008) and definition question answering (Xu et al., 2003; Cui et al., 2005). Even though there are only two possible answers, yes or no, such questions can be quite hard to answer. Search technologies cannot be applied easily because in many cases we can not rely on the existence of explicit negative evidence in the information source, *e.g.* "Chirac was *not* a president of Germany."

This paper tackles yes/no question answering by exploiting an existing system for factoid-style question answering. The key idea, inspired by *question inversion* (Prager et al., 2006), involves generation of factoid questions by replacing some parts of a given proposition with abstract (*i.e.* ungrounded) expressions. For example, to determine the correctness of (1) and (2), two question sentences are generated for each proposition, with abstracted parts in italics and *anticipated answers* in brackets[2].

(1a) *He* was the president of France in 2000. [Chirac]

(1b) Chirac was the president of *this country* in 2000. [France]

(2a) *He* was the president of Germany in 2000. *[Chirac]

(2b) Chirac was the president of *this country* in 2000. *[Germany]

DeepQA (Ferrucci, 2012), the factoid-style question answering system used here accepts these sentences with focal words (*e.g.* nouns with 'this', or pronouns such as 'he') as input questions, generates a hit-list of candidate answers, and assigns confidence values to candidate answers. Usually the system's output to the question is the top-ranked answer, the answer with the highest confidence value. Our hypothesis is that if an original proposition

[1] '*' denotes a false proposition.

[2] '*' before brackets denotes the anticipated answer should not be answered because the original proposition is false.

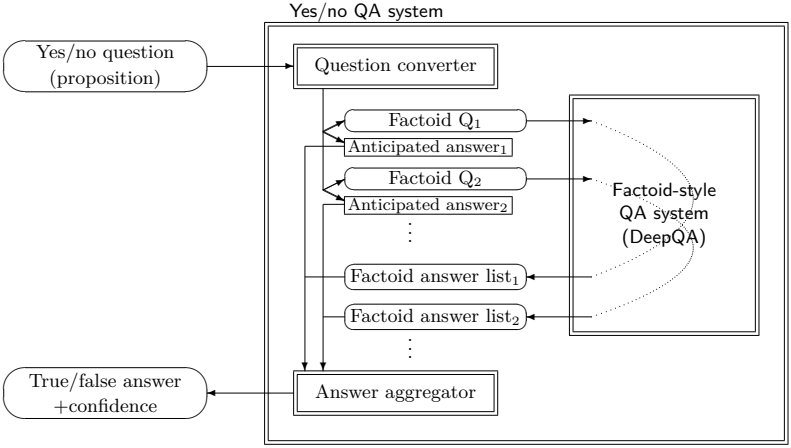

Figure 1: The flow of yes/no question answering using the factoid-style question answering.

is true, then the anticipated answer should be the top-ranked answer from DeepQA, but for false propositions, the anticipated answer should either have a low confidence value, or not be ranked high, or both. Since DeepQA outputs "France" for (1b) and (2b) as the top-ranked answer, we can use that as positive evidence for proposition (1), and negative evidence for proposition (2). The exact formula for processing the results of the generated questions is discussed in Section 4.

There is no natural repository of true and false facts which can be used for evaluation of the fact validation system. It is even difficult to artificially generate wrong facts from the set of correct facts without bias. Therefore we chose to use questions of multiple-choice of true (or false) statements in a university admission test as a reliable benchmark. In this context, wrong statements are carefully produced by the examining board by mixing a correct statement with partially wrong information.

The main contribution of this paper is to provide a novel framework for benchmarking and discussion of a method of processing yes/no questions, and propose a method to appropriately make use of existing technologies of factoid-style question answering and its underlying information sources.

Figure 1 illustrates the overall architecture to solve yes/no questions with an existing factoid-style question answering system, DeepQA. The yes/no question answering system takes a proposition as an input and outputs the correctness of the proposition with a confidence value. DeepQA is used as a subsystem, which takes a factoid-style question as an input and outputs an answer list with confidence values.

The key methods are conversion from yes/no questions into factoid-style questions, and aggregation of the results of factoid answers to determine yes or no. Rather than the implementation of fully automatic system, this study focuses on the methodologies and the headroom analysis of this fact validation. Thus the conversion of questions is done

manually, but it is conducted by using specific algorithms and can be automated using standard techniques and resources. The subsequent stages are automatic and accuracies of several methods will be compared.

Section 2 describes some of the prerequisites for the discussion in this paper, the yes/no type question answering, the existing factoid-style question-answering system, the original idea of question inversion, and the Japanese university admission test that was used for the evaluations. Section 3 shows the method of question conversion and problems in question generation. Section 4 describes the answer aggregation module, which selects the most appropriate answer lists for the fact validation. The experimental results appear in Section 5, and Section 6 provides the further discussion on the results and comparison with another approach. Section 7 concludes this paper.

2 Background

2.1 Yes/no question answering

From linguistic point of view, a yes/no question is defined as an interrogative sentence that is answered by either yes or no. A yes/no question in English language is often generated by placing a positive or negative form of an auxiliary verb (*e.g.* does, can, isn't, etc.) before the subject.

However, yes/no questions are not always correctly answered by just one of two answers, yes or no. Hirschberg (Hirschberg, 1984) pointed out that yes/no questions can have several functions, including *scalar implicature* such as (3Q), where the answer (3A) makes the correction to the proposition in the question.

(3Q) Did you invite Mary?

(3A) I invited Bob.

Green et al. (Green and Carberry, 1994) investigated such indirect answers (*i.e.* other than yes/no) to yes/no questions from discourse representation based on Rhetorical Structure Theory (Mann and Thompson, 1987). They provided an algorithm to generate appropriate answers that fill the gap between the question and the discourse contexts. The process is designed to provide the automatic responses in a dialog system.

This paper aims at fact validation of propositions, so the main outputs are yes or no. Though the initial input in our test domain is not in the form of interrogative question but declarative expressions, the intrinsic task is the same; indeed we will show that our approach has the capability of suggesting the true fact from a factually incorrect input.

2.2 Factoid-style question answering system

We use an open-domain factoid-style question answering system, DeepQA (Ferrucci, 2012), which returns multiple answers with each confidence value ranging from 0 to 1. DeepQA first analyzes the input question sentence, identifies its answer type amongst other question processing, generates candidate answers from searches performed corpora including encyclopedia, news paper articles and thesauri, gathers evidences for each candidate answer by evaluating a collection of feature scorers on it, and calculates each candidate's final confidence value by accumulating these scores and weighting them via a model of learned on trained data.

In the candidate answer generation phase, the candidates are not restricted to terms which have the answer type determined by question analysis, but any term found in returned documents which passes certain filtering criteria can be selected as a candidate. The coincidence between the type of the candidate and the type of question is used as one of the features that support the candidate answer to be correct.

A confidence value is calculated using multiple features computed from some analysis that the candidate answer should be correct. Each feature type is weighed by a model output from a machine learning process trained on a repository of several thousand question-answer pairs. The confidence values, calculated independently for each candidate, estimate the likelihood to be the correct answer (so the sum of confidence values across all answers may exceed 1, reflecting the fact that multiple answers may be correct). Thus there is a strong correlation between the confidence value and the precision of the answer.

According to (unpublished) experiments, DeepQA achieved an accuracy of 67% on TREC 2002 (Voorhees, 2002) question set, trained using about 8,000 QA pairs from other years of TREC and another data source.

2.3 Question Inversion

Question inversion (Prager et al., 2006) is a method to generate new questions from a different point of view to rerank the candidate answers in factoid-style question answering. It can be used with any QA system. For example, when a QA system solves question (4),

(4) What was the capital of Germany in 1985?

the system first generates several candidate answers, like "Berlin", "Moscow", etc. The next step is to generate the abstract inverted question (4i), by removing Germany, the *pivot term*.

(4i) Of what country was *CANDIDATE* the capital in 1985?

A series of grounded inverted questions is generated by replacing *CANDIDATE* in turn with each of the candidate answers to (4i) (*e.g.* "Of what country was Berlin the capital in 1985?"). These inverted questions are processed by the QA system, and the "inverted" answers (*e.g.* "Germany", "Soviet Union" etc.) along with their scores are gathered. Original candidate answers ("Berlin", "Moscow") are given credit when the corresponding inverted question generates the pivot ("Germany", in this case). The candidate's final score is calculated as a function of its original score and the credit from inversion.

2.4 Examination for university admission

For this study we use the National Center Test for University Admission[3], which is the standardized achievement test for high school students who wish to enter universities in Japan. The examination on world history was selected because most of the questions are solvable as fact-validation problems with general knowledge. The questions are provided in a multiple-choice style as exemplified in Table 1, which shows the description of the question and the four candidate statements. In this case only one of four is a correct statement,

[3]http://www.dnc.ac.jp/

From 1-4 below, choose the one sentence that is correct in regard to the person/people that it describes.	
1	Ouyang Xiu and Su Shi were the best known writers of the Tang dynasty.
2	Yen Chen-ching was the best known chirographer in the Sung dynasty.
3	Wang Anshi of the Sung era implemented the reform called the New Policies.
4	Qin Hui fought with the war hawks on the relationship with Yuan.

Table 1: An example of questions in the world-history examination (from 2009 National Center Tests on World History). The correct answer is 3.

and the others are incorrect, so we can frame the individual choices as test cases for fact validation. The original examinations are provided in Japanese, and this paper used their English translation, which was translated by a native speaker with domain knowledge.

This particular examination is also used for evaluating a task of recognition of textual entailment (Miyao et al., 2012; Shima et al., 2011). These true or false statements can be test cases of hypotheses which can be entailed from relevant texts found in encyclopedia articles. The multiple-choice solution using textual entailment will be compared to our approach in Section 6.2.

3 Question conversion

The question conversion module generates multiple factoid-style questions to be input to DeepQA from the original proposition. As described in Section 1, because the component to perform the inversion was not completed in this time, this module's function is performed manually in this study so that we can discuss the intrinsic features of yes/no question answering, but this section gives algorithms for each operation.

3.1 Conversion method

First, *pivot terms*, the key entities to be abstracted are selected from the original proposition. While the original work chose a single pivot per question, there is nothing about the question inversion process that requires there be only one pivot/abstract inverted question per input question, and in fact we use multiple. Since most of questions in the test domain contain proper nouns, often multiple proper nouns, we decided to use proper nouns as pivot terms. A back-off method of choosing common nouns is employed when there are no proper nouns. For example, two pivot terms (underlined) are found in proposition (5), which is the second row in Table 1.

(5) Yen Chen-ching was the best known chirographer in the Sung dynasty.

Next, a *type* is determined for each pivot term. For this algorithm, we define a type to be a common noun which is a hypernym of the pivot term. A thesaurus such as WordNet can be used, but choosing the right level of abstraction can be a problem. An effective approach to making this selection involves mining the corpus to determine co-occurrence counts with each possible hypernym (Prager et al., 2001). A simpler solution, which we adopt here as a guideline, is to select the type from the first sentence of Wikipedia articles about the subject term, from the observation that many opening sentences are of the form "X is Y" where X is the title of the article and Y is its type (Kazama and Torisawa, 2007). In the

example above, “calligrapher” and “dynasty” are assigned as types of “Yen Chen-ching” and “Sung dynasty”, respectively.

The pivot term is then replaced with a noun phrase consisting of “this” and the type. When the type is Person and the pivot term is the subject of the head predicate of the proposition, just “he” or “she” is used. From (5), two factoid questions (5a) and (5b) are generated, the substitutions denoted by italic.

(5a) *He* was the best known chirographer in the Sung dynasty.

(5b) Yen Chen-ching was the best known chirographer in *this dynasty*.

For each question there is the *anticipated answer*, that corresponds to the pivot term in the original proposition. The questions are input to DeepQA and the results will be compared to the anticipated answers, “Yen Chen-ching” for (5a) and “Sung dynasty” for (5b).

This idea of generation of new questions is inspired by question inversion, described in Section 2.3. However, the method in this paper is different from the original question inversion (Prager et al., 2006) mainly in three points:

Multiple inverted questions. Instead of a single pivot term, multiple pivot terms are selected and multiple questions are generated in this study. All of them are input to DeepQA and the results are aggregated in a subsequent module. This usage of multiple questions makes the reliance on DeepQA (both its competence and the coverage of the test domain in its corpora) less brittle.

Generation of sentences. In our method the inverted sentences are generated as natural-language questions, instead of internal structures for a specific system. This process is more general.

No need of candidate answers. When question inversion is used for answer reranking in factoid-style question answering, candidate answers must be generated to fill the slot of the original question. In our method, possible answers for a yes/no question are just yes or no, thus traditional candidate generation is not required in the question conversion module.

The questions generated by the method above are input to DeepQA, and the multiple answers with confidence values are returned for each question. Table 2 shows an example.

Ideally the anticipated answer is at the top of the DeepQA’s answer list when the original proposition is correct, although, we cannot rely on this since DeepQA is not perfect. For example, the third question in Table 2 is an unlucky case: the original proposition is correct, but the anticipated answer was at the second place in DeepQA’s answer list, due to complexity of the responsive text in the corpus. The anticipated answer was not returned for the fourth question in Table 2. This problem will be addressed by the answer aggregation module described in Section 4.

	Generated questions	Anticipated answer	1st	2nd	3rd
q_1	In *this country*, Xuanzang traveled to India and brought home the Buddhist scriptures during the Tang period.	[China]	China 0.702	Nepal 0.513	Burma 0.129
q_2	In China, *he* traveled to India and brought home the Buddhist scriptures during the Tang period.	[Xuanzang]	Xuanzang 0.593	Tang 0.147	Buddhism 0.11
q_3	In China, Xuanzang traveled to *this country* and brought home the Buddhist scriptures during the Tang period.	[India]	monk 0.23	India 0.216	Faxian 0.135
q_4	In China, Xuanzang traveled to India and brought home the Buddhist scriptures during *this period*.	[Tang period]	Buddhism 0.441	monk 0.219	Faxian 0.127

Table 2: Examples of answer lists output by DeepQA, for the original (true) proposition "In China, Xuanzang traveled to India and brought home the Buddhist scriptures during the Tang period." The numbers under the answers indicate the confidence values.

3.2 Problematic questions

Besides failures by DeepQA due to just lack of information in the corpus or inability to process it correctly, there are intrinsic problems in the use of DeepQA as the evidence of a fact validation. We identified two major problematic phenomena: *multiple-answer questions* (*i.e.* questions with more than one answer) which may not return the anticipated answer as the first answer for a true proposition, and *attributive questions* which may produce the anticipated answer for a false proposition.

Multiple-answer questions. Proposition (1) in Section 1 has enough context in it to be correctly validated as-is; however, suppose the phrase "in 2000" were dropped, producing proposition (6), with inverted questions (6a) and (6b).

(6) Chirac was the president of France.

(6a) *He was the president of France.* $^{?}$[Chirac]

(6b) Chirac was the president of this country. [France]

(6b) is not problematic, but (6a) has several possible answers ("Mitterand", "Chirac", "Sarkozy", "Hollande", ...), so even if the original proposition (6) is true, "Chirac" may not be the DeepQA's first answer to (6a).

Attributive questions. Proposition (7) below is false. Structurally, (7) asserts two properties of an entity ("Carlos I"), and whether one uses said entity or one of the properties as a pivot can have very different consequences. Using the entity as a pivot is generally a good choice in such situations, since the two (or however many are present) properties can together triangulate a unique answer. This does not happen so readily when a property is chosen as a pivot.

*(7) Carlos I, the King of Spain, was also the King of Portugal.

(7a) He, the King of Spain, was also the King of Portugal. *[Carlos I]

(7b) *Carlos I, the King of this country, was also the king of Portugal.* *?[Spain]

(7a) is confirmed to be a desirable inverted question because DeepQA actually returns "Philip II of Spain" as the first answer. This generates a correct statement, knowledge of which is the point of the question. However, if (7b) is used as an inverted question, "Spain" is returned by DeepQA as the first answer. This is because there is no country which is more associated with "Carlos I" than "Spain", regardless of other contexts in the sentence. This suggests that the kind of pivot term which is an attribute of a specific named entity may not be suitable for generation of inverted questions. This issue will be addressed in the answer aggregation phase discussed in Section 4.

4 Aggregation of answers

This section describes the answer aggregator which processes the multiple answer lists from the generated questions and decides the correctness of the input proposition.

4.1 Answer matching

The initial operation by the answer aggregator is to compare the anticipated answer and the answer lists output by DeepQA. Here we introduce a function $A(q)$, which stands for the rank where the anticipated answer appears in the answer lists for a question q. If the anticipated answer is not found in the answer list for q, $A(q) = \text{nil}$. For example, the third question in Table 2 is expressed as $A(q_3) = 2$.

There is clearly an important signal in whether the anticipated answer appears on the top of the answer list. When the first answer is the anticipated answer (*i.e.* $A(q) = 1$) it will provide evidence that the original proposition is true. On the other hand, when the first answer is different from the anticipated answer (*i.e.* $A(q) \neq 1$), it provides evidence that the original proposition is false.

Given the importance of this test, the matching of the anticipated answer and each candidate answer should be more robust than exact match. We relax the matching criteria in the following ways:

- Case insensitive matching.
- The existence of type name is optional (*e.g.* "Sung dynasty" and "Sung" can match).
- Synonyms, or transliteration variations (*e.g.* "Sung dynasty" and "Song dynasty") can match. In this study a synonym list is created manually by seeing the pairs of the anticipated answers and the first answers of DeepQA, without seeing the result. This can be replaced by an automatic process, such as using Wikipedia redirects.

4.2 Question/answer selection

Now the question is how to interpret multiple answer lists output by DeepQA to determine the correctness of a given proposition. An ultimate goal of this work is to develop an algorithm that combines the support for/against the truth of the different propositions

from all of the inverted questions used. However, we have found that selecting just one question per proposition requires a simpler algorithm, and can give good results.

For simplicity, we use only the top-ranked answer from the desired inverted question, that is, the system outputs true if $A(q) = 1$ and outputs false if $A(q) \neq 1$. Therefore the problem is reduced to selection of q^*, the question for which DeepQA generates the most representative answer list, from the set of questions Q. If we define $R(q)$ to be the *reliability* of question q, then the decision of true or false of the given proposition is formalized as

$$TF(Q) = \begin{cases} \mathsf{true} & A(q^*) = 1 \\ \mathsf{false} & otherwise \end{cases} \\ where \quad q^* = \underset{q \in Q}{argmax}(R(q)) \tag{8}$$

Then the problem is the design of the reliability function $R(q)$. The basic idea is the use of confidence value, relying on the correlation between the confidence value and the precision of the answer in DeepQA's outputs. Define

$$R_1(q) = C(q, 1) \tag{9}$$

where $C(q, n)$ is DeepQA's confidence value for n-th answer to question q. In Table 2, $q^* = q_1$ because q_1 has the highest confidence value for its first answer, then $TF(Q) = \mathsf{true}$ because q_1's first answer, "China" matches the anticipated answer. The correct answer is true, so it shows the method works well for the decision.

Now remember that in Section 3.2 we found two types of questions that are not suitable for fact validation. To consider multiple-answer questions, we can penalize questions where the first answer and second answer have similar confidence values. We revise (9) to

$$R_2(q) = \begin{cases} p_2 R_1(q) & \frac{C(q,2)}{C(q,1)} > \theta_2 \\ R_1(q) & otherwise \end{cases} \tag{10}$$

where p_2 is a penalty value less than 1.0, and θ_2 is a threshold value between 0.0 and 1.0 to estimate whether q is a multiple-answer question.

To take attributive questions into consideration, another penalty value should be multiplied to $R_2(q)$, when the pivot term of q is an *indirect entity*, which is one of the followings:

- A constituent of a phrase modifying other proper noun or content noun (*e.g.* "Akbar of the Mughal Empire")
- Appositive modifiers (*e.g.* "Tenochtitlan, the capital of Aztec empire")

The updated formula is (11). For example, the pivot term of question (7b) meets the second condition above, so the reliability of the question will be penalized with p_3.

$$R_3(q) = \begin{cases} p_3 R_2(q) & q\text{'s pivot term is indirect entity} \\ R_2(q) & otherwise \end{cases} \tag{11}$$

Besides $TF(Q)$, the decision of true or false, the system returns the confidence value $C^*(Q)$ as well, simply defined as $C^*(Q) = C(q^*, 1)$ *i.e.* DeepQA's confidence value of the first answer for the most reliable question.

Number of proper nouns	0	1	2	3	4
Frequency	0	1	53	48	2

Table 3: Distribution of number of proper nouns in statements of the development set.

5 Experiments

5.1 Experimental setup

For the evaluation of yes/no question answering, we used two sets of world history examination data from National Center Test for University Admission, the set in the year of 2009 for development data, and set in the year of 2007 for open test data. They contain 26 and 23 multiple-choice questions each of which has four sentential statements, thus 104 and 92 predicates can be used for true/false validation as development and test data, respectively. 84% of questions require the answerer to choose a correct statement out of four, and rest of them are to choose an incorrect statement out of four.

Some statements need preprocessing to be self-contained statements, because sometimes necessary information is found in the description of the question, instead of the statements to be chosen. This preprocessing is same as that in the use of the same data for textual entailment (Miyao et al., 2012). For example, when a question description says "choose the most appropriate sentence concerning events that occurred during the period of Ming dynasty", the statement (12) need to be updated to (12m) to determine its correctness.

(12) A Buddhist sect called Zen was created.

*(12m) A Buddhist sect called Zen was created during the period of Ming dynasty.

Table 3 shows the distribution of number of proper nouns in statements of the development set. This shows that most of statements have multiple proper nouns to be ungrounded to produce multiple inverted questions.

The main metrics of this experiment is the accuracy of true/false determination, and we also evaluated the accuracy of 4-way multiple-choice questions, which actually examinees answer. To answer such question types, we introduce the *correctness score* $CS(Q)$ defined as (13).

$$CS(Q) = \begin{cases} C^*(Q) & TF(Q) = \mathsf{true} \\ (-1)C^*(Q) & TF(Q) = \mathsf{false} \end{cases} \tag{13}$$

With this score, "select the correct statement" questions can be solved by selecting the statement with the highest $CS(Q)$, that is, the proposition judged true with the highest confidence value, or the one with the lowest confidence when all propositions are judged false. Also "select the wrong statement" questions can be answered by just selecting the lowest $CS(Q)$.

5.2 Experimental results

First we evaluated the accuracies of true/false determination and 4-way selection using the development data (examination in the year of 2009). Baseline for true/false evaluation is

Method	true/false		p	4-way	
Baseline	65.4%	(68/104)		25%	
Model 1	77.8%	(81/104)	.043	50%	(13/26)
Model 2	79.8%	(83/104)	.029	58%	(15/26)
Model 3	81.7%	(85/104)	.017	62%	(16/26)

Table 4: The result of closed test with the development set (2009).

Method	true/false		p	4-way	
Baseline	68.4%	(63/92)		25%	
Model 1	69.6%	(64/92)	.500	57%	(13/23)
Model 2	71.7%	(66/92)	.418	57%	(13/23)
Model 3	79.3%	(73/92)	.046	65%	(15/23)

Table 5: The result of open test with 2007 question set.

the accuracy when returning always false, and baseline for 4-way evaluation is the random choice (theoretically 25%). The proposed method was evaluated with three models. Model 1 uses $R_1(q)$ in equations (8) and (9). Model 2 addresses the problem of multiple-answer questions with $R_2(q)$ in equation (10), where both p_2 and θ_2 were empirically set to 0.5. Model 3 cares also attributive questions with $R_3(q)$ in equation (11). p_3 is empirically set to 0.1 here.

Table 4 shows the results. According to p-value between the baseline and each model in the true/false determination, all models significantly outperformed the baseline. We set parameters so that Model 3 shows the highest accuracies, but the differences amongst the three models were not significant, due to the small test set.

Table 5 shows the result using the test data (the examination in 2007), using parameters set from the development data. Though Model 1 and Model 2 did not show a significant difference from the baseline in terms of the yes/no determination, Model 3 shows highest accuracy and significantly outperformed both the baseline and Model 1. In the 4-way test, all models apparently outperformed the random choice.

The results with the test set showed that referring to both the multiple-answer questions and attributive questions can be appropriately discouraged. For example, when a wrong statement (14) is converted to (14a), the country which ruled Vardhana dynasty, India, was returned as the first answer by DeepQA with high confidence, regardless of the role of Angkor Wat. When Model 3 is used, (14a) was penalized because the pivot term is a modifier to another proper noun. This gave another question (14b) relatively higher priority, resulting in the answer "Chalukya", which is actually the ruling dynasty when Angkor Wat was built. The resulting mismatch with the anticipated answer successfully supports the conclusion that (14) is false.

*(14) Angkor Wat was built during the same period as the Vardhana dynasty ruled in India.

(14a) Angkor Wat was built during the same period as the Vardhana dynasty ruled in this country. ?[India]

(14b) Angkor Wat was built during the same period as this dynasty ruled in India. *[Vardhana dynasty]

5.3 Error analysis

From unsuccessful cases, we found some difficulties. Here is a typical example of DeepQA's limitation. For the correct statement (15), three questions (15a) to (15c) were generated, but DeepQA's results were $A(q) \neq 1$ for all of them. The difficulty comes from ambiguity in "February Revolution", so "Russia" is answered for (15a) due to more popular revolution, and (15c) is difficult to answer.

(15) French provisional government formed after the February Revolution created National Workshops.

(15a) Provisional government of this country formed after the February Revolution created National Workshops. [France]

(15b) French provisional government formed after this revolution created National Workshops. [February Revolution]

(15c) French provisional government formed after the February Revolution created this place. [National Workshops]

Another interesting case was (16). "Agricultural Adjustment Act" and "United States" were answered for (16a) and (16b), respectively, but this statement was wrong because the prices were *raised*, not *lowered*. This is the limitation of DeepQA that answers the most likely proper nouns assuming there is an answer. To overcome this problem, other questions with opposite meaning (*e.g.* The prices of agricultural products were *raised*...) could be generated and submitted to DeepQA. If DeepQA performs well enough, it might return higher confidence values with questions derived from the correct statement.

*(16) The prices of agricultural products were lowered by the Agricultural Adjustment Act in United States.

(16a) The prices of agricultural products were lowered by this law in United States. ?[Agricultural Adjustment Act]

(16b) The prices of agricultural products were lowered by the Agricultural Adjustment Act in this country. ?[United States]

6 Discussion

6.1 Beyond yes/no answer

This system does not just return yes/no answers. For false propositions, the answer by DeepQA can suggest what element makes the proposition false, as we found in the example (7a) and (14b). We call this task *false trigger identification*.

Table 6 shows the frequency of such triggers found in the answer lists. Among the 68 false propositions in the development data, 11 false triggers were found as the first answer of

Question rank	Answer rank	Frequency
1st	1st	11
2nd to 3rd	1st	6
1st	2nd to 5th	2
2nd to 3rd	2nd to 5th	5

Table 6: The frequencies of false triggers found in the answer lists.

q^*, 6 false triggers were found as the first answer to other questions, and 7 were in the lower-ranked answers to one of the questions. For these incorrect statements, our system thus provides useful hints to make corrections to the input.

In 12 out of 46 unsuccessful cases the time period was incorrect as in Question (17) when the 15th century is the correct date. In these cases it is difficult to identify the false trigger because DeepQA was not particularly good at answering questions seeking the proper century.

*(17) In the 11th century, the ruler of the Malacca Sultanate converted to Islam.

6.2 Comparison with entailment techniques

Recently recognition of textual entailment (RTE) (Dagan et al., 2005) is intensively studied. The same world history question dataset was used in the evaluation of entailment task (Miyao et al., 2012). In their experiments, for a statement H in the choices for a question, a relevant sentence T was manually selected among Wikipedia articles so that H entails T only if statement H is true. They reported that the accuracy of 4-way multiple choice question in a world history examination using the best system was 54% (13/24). Since the prerequisites were different and the languages were also different (they used the original Japanese questions while we used English translations) we cannot directly compare the results. However, the accuracy using entailment is lower than that by our closed test in Table 5, and our method does not require the careful selection of a relevant single text from the corpus, so our approach can be argued that it is closer to a real application scenario.

7 Conclusion

This paper proposed a method to determine the correctness of propositions by leveraging the power of information sources accessed through a factoid-style question answering system. This was done by generating questions to be input to a subsystem and postprocessing the answer list produced by the system. This achieved quite encouraging results in both true and false determination, and hence in the overall examination goal of answering multiple choice questions. In this study, although the first stage of the process was manual, we identified kind of questions that are suitable or problematic for fact validation, and we found that reliable questions can be automatically selected by relying on the confidence values output by the factoid-style question-answering system. Besides yes/no determination, the system indicated an ability to identify why an input statement might be incorrect, suggesting our method can be applied in a dialog system that continuously validates the input utterances, or for semantic document content correction based on general knowledge, amongst others.

References

Bian, J., Liu, Y., Agichtein, E., and Zha, H. (2008). Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th international conference on World Wide Web*, pages 467–476.

Cui, H., Kan, M.-Y., and Chua, T.-S. (2005). Generic soft pattern models for definitional question answering. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 384–391.

Dagan, I., Glickman, O., and Magnini, B. (2005). The PASCAL recognizing textual entailment challenge. In *MLCW*, pages 177–190.

Ferrucci, D. A. (2012). Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15.

Green, N. and Carberry, S. (1994). Generating indirect answers to yes-no questions. In *Proceedings of the seventh International Workshop on Natural Language Generation*, pages 189–198.

Hirschberg, J. (1984). Toward a redefinition of yes/no questions. In *Proceedings of 10th COLING and 22nd ACL*, pages 48–51.

Kazama, J. and Torisawa, K. (2007). Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of EMNLP-CoNLL 2007*, pages 698–707.

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: Description and construction of text structures. In Kempen, G., editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pages 85–95. Nijhoff.

Miyao, Y., Shima, H., Kanayama, H., and Mitamura, T. (2012). Evaluating textual entailment recognition for university entrance examinations. *ACM Transactions on Asian Language Information Processing*, 11(4). (to appear).

Prager, J., Radev, D., and Czuba, K. (2001). Answering what-is questions by virtual annotation. In *Proceedings of Human Language Technologies Conference*, pages 1–5.

Prager, J. M., Duboué, P. A., and Chu-Carroll, J. (2006). Improving QA accuracy by question inversion. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47.

Shima, H., Kanayama, H., Lee, C., Lin, C., Mitamura, T., Miyao, Y., Shi, S., and Takeda, K. (2011). Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In *Proceedings of NTCIR-9 Workshop Meeting*.

Voorhees, E. M. (2002). Overview of the TREC 2002 question answering track. In *Proceedings of the 11th Text REtrieval Conference (TREC)*, pages 115–123.

Xu, J., Licuanan, A., and Weischedel, R. M. (2003). Trec 2003 QA at BBN: Answering definitional questions. In *Proceedings of the 12th TREC*, pages 98–106.