IEICE TRANS. INF. & SYST., VOL.E93–D, NO.3 MARCH 2010

PAPER

# An Unsupervised Model of Redundancy for Answer Validation

**Youzheng WU**[†a)], **Hideki KASHIOKA**[†b)], *Nonmembers*, *and* **Satoshi NAKAMURA**[†c)], *Member*

**SUMMARY** Given a question and a set of its candidate answers, the task of answer validation (AV) aims to return a Boolean value indicating whether a given candidate answer is the correct answer to the question. Unlike previous works, this paper presents an unsupervised model, called the U-model, for AV. This approach regards AV as a classification task and investigates how effectively using redundancy of the Web into the proposed architecture. Experimental results with TREC factoid test sets and Chinese test sets indicate that the proposed U-model with redundancy information is very effective for AV. For example, the top@1/mrr@5 scores on the TREC05, and 06 tracks are 40.1/51.5% and 35.8/47.3%, respectively. Furthermore, a cross-model comparison experiment demonstrates that the U-model is the best among the redundancy-based models considered. Even compared with a syntax-based approach, a supervised machine learning approach and a pattern-based approach, the U-model performs much better.

***key words:*** *question answering, answer validation, unsupervised model, Web mining, support vector machine*

## 1. Introduction

Given a question and a set of candidate answers, the task of answer validation (AV) is to return a Boolean value for each candidate answer, indicating whether the candidate is the correct answer to the question. This is an emerging topic in question answering (QA) (`http://nlp.uned.es/QA/ave/`). Automatic techniques for AV are of great interest for the development of open-domain QA systems.Spurred by TREC (`http://trec.nist.gov/`), CLEF (`http://www.clef-campaign.org/`), and NTCIR (`http://research.nii.ac.jp/ntcir/`), manyapproaches have been presented, such as the retrieval-based model [14], the pattern-based model [8], [21], the deep NLP-based model [7], [9], [10], [20], and the machine-learning-based model [2], [12], [15], [18], [19], [27]. Many of these approaches independently process the candidate-bearing snippets and do not use information on data redundancy among these snippets to help select the correct answer from the candidate answers. For example, the machine-learning-based model independently estimates the probability $p(c_i|s)$ of generating answer $c_i$ from each candidate-bearing snippet $s$ and then selects the candidate $c_i^*$ with the highest probability as the correct answer. As a result, answers cannot always be identified in cases in which the answers occur in snippets with low similarities with respect to the question. Our proposed unsupervised model of using redundancy information learned from the Web automatically can partially resolve this problem.

Considering the test questions of TREC, however, we find that candidate-bearing snippets containing the same candidate answer include roughly the same sub-meaning. Table 1 lists some Google snippets in response to the query that is composed of the candidate answer "*1969*" and the keywords from the TREC04 test question *When was the first Crip gang started*. This table indicates that most of these Google snippets (such as $e_1, e_2, e_3, and\ e_4$) express the same meaning, i.e., the time of establishment of *the first Crip gang*. By considering these snippets together and using them to help select answers to questions, an AV system might achieve better performance than that of systems that independently process these candidate-bearing snippets. Approaches based on this observation are called redundancy-based models. Some pioneering studies [3], [4], [16], [22]–[24] have investigated redundancy from the Web for the AV in QA.

This paper presents a novel redundancy-based model incorporating data redundancy from the Web for the AV task. Our approach is an unsupervised model, called the U-model, which has the following characteristics:

- It regards the AV task as a kind of classification task;
- The training data required by the classifier can be learned automatically;
- It is independent of language and can be implemented with limited resources.

To the best of our knowledge, no research on the kind of study we discuss here has been reported. To validate the proposed U-model, we performed extensive experiments in terms of TREC English test data sets and Chinese test data sets. Our major findings include the following:

1). The U-model can achieve very competitive performance with $top@1/mrr@5/top@5$ scores for the TREC05 and 06 sets of 40.1/51.5/71.9%, and 35.8/47.3/ 66.8%, respectively (as described in Sect. 4.2).

2). The overlap, Boolean, candidate, and context features (as described in Sect. 3.2) play more important roles (as described in Sect. 4.3).

Manuscript received May 14, 2009.
Manuscript revised October 6, 2009.

[†]The authors are with Spoken Language Communication Group, National Institute of Information and Communications Technology (NiCT), Kyoto-fu, 619–0289 Japan.
a) E-mail: youzheng.wu@nict.go.jp
b) E-mail: hideki.kashioka@nict.go.jp
c) E-mail: satoshi.nakamura@nict.go.jp

DOI: 10.1587/transinf.E93.D.624

**Table 1** Some Google snippets containing answer candidate "1969" to question *When was the first Crip gang started.*

| | |
|---|---|
| $e_1$ | ...like the Bloods and Crips that are well-known today. It is believed that the first Crip gang was formed in late 1969. During this time in Los Angeles there ... |
| $e_2$ | ...Not long after the first Bloods and Crips gangs started forming in Los Angeles in late 1969, the Island Bloods sprung up in north Pomona and ... |
| $e_3$ | ...formed by 16 year old Raymond Lee Washington in 1969. Williams joined Washington in 1971 ...be called the Crips. It was initially started to eliminate all street gangs ... |
| $e_4$ | ...In three years, after the first Crip gang was established in 1969, the number of black gangs in Los Angeles had grown to 18. Table 1 reveals that in each ... |
| $e_5$ | ...the first writer to win ...the infamous CRIP gang. ...The father gave Tookie a ...started his gang. It appears that the young people .... Hall takeover of 1969 by students demanding ... |

3). The U-model is independent of Web search engines because the performance with different search engines does not significantly vary (as described in Sect. 4.4).

4). A cross-model comparison demonstrates that the U-model statistically significantly outperforms previous redundancy-based models [3], [16] (as described in Sect. 4.5).

5). Even compared with the conventional syntax-based model [9], a supervised machine learning approach and a pattern-based approach, the U-model can achieve much better performance (as described in Sect. 4.6, 4.8 and 4.9).

## 2. Comparison among Models

The characteristic of the redundancy-based approaches is the use of Web redundancy for more reliable answer validation. This section introduces some closely related redundancy-based approaches.

### 2.1 Magnini Model

Magnini model [3] incorporates Web data redundancy into an answer validation score that assesses the correctness of a candidate answer $a$ with respect to a question $q$. The best performance is obtained with the answer validation score computed as,

$$CCP(Qsp, Asp) = \frac{h(Qsp\ NEAR\ Asp)}{h(Qsp) \times h(Asp)^{2/3}} \times Mp^{2/3} \quad (1)$$

where $Qsp$ is a question sub-pattern composed by combining keywords in question $q$ by using the proximity operator $NEAR$ and $OR$; $Asp$ is an answer sub-pattern composed by combining keywords in candidate answer $a$ by using $NEAR$ and $OR$; $h(A)$ is the number of pages retrieved by AltaVista, in which pattern $A$ appears, with $A$ denoting $Qsp$, $Asp$, or the answer validation pattern $Qsp\ NEAR\ Asp$; and $Mp$ is the maximum number of pages indexed by the search engine.

### 2.2 Aranea Model

The Aranea model [16] is a complete re-implementation, with additional refinements, of the original askMSR system [23]. Its performance in TREC 2002, 2003, and 2004 tracks was competitive. The main contribution of the Aranea model is a new QA architecture in which there is no need to extract candidates, which is necessary in the traditional QA architecture. Figure 1 illustrates the architecture of the Aranea model. In this architecture, *Rewrite Query* generates a set of weighted rewrites of the question, which are likely substrings of declarative answers to the question. According to the snippets returned by the Web search engine, *Generate N-grams* extracts all N-grams in the snippets as possible answers. *Filter N-grams* scores the N-grams according to the weights of the rewrite rules that generated them and the numbers of unique snippets in which they occurred, and it filters out N-grams whose scores are under a predefined threshold. In *Combine N-grams module*, unigrams are used as evidence to boost the scores of longer answer candidates. Finally, *the Rerank N-grams module* selects the top-ranked N-grams as answers.

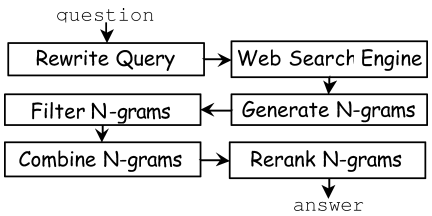


**Fig. 1** Architecture of the aranea model.

The disadvantages of both the Magnini model and the Aranea model include: 1). They heavily depend on the occurrences of the candidates, few other features are incorporated. 2). Much noise is introduced into the models, even though some heuristic rules are used in the Aranea model. The snippet $e_5$ in Table 1, for example, is a noise snippet because it does not express the meaning, i.e, the time of establishment of the first Crip gang. The two models, however, treat it equally as the other four snippets, which will lead to lower performance.

In response to this situation, this paper presents the U-model for answer validation in open-domain QA system that can partially solve the problems.

In addition, [22], [24] incorporated the web-boosting features of exploiting redundancy on the Web to improve their conventional QA system that is based on syntactical and lexical similarity.

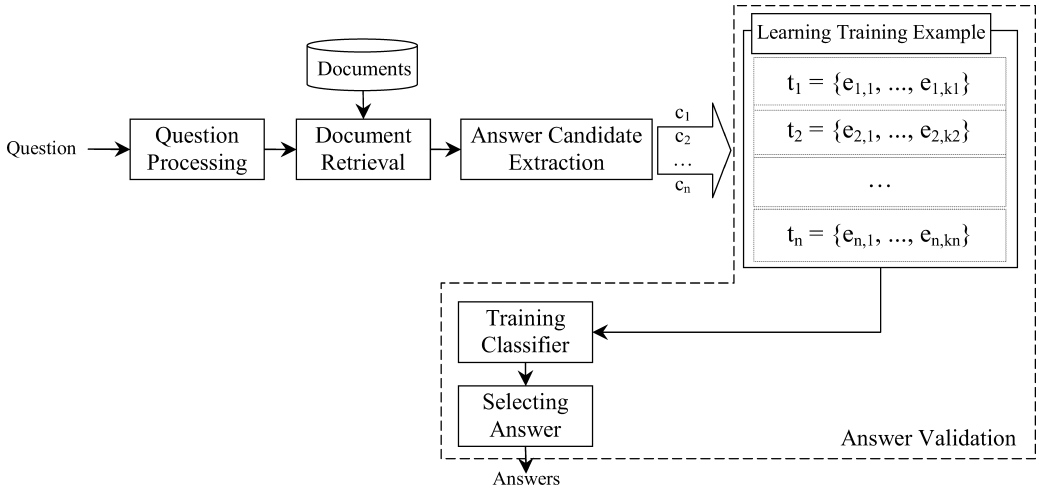


**Fig. 2** Architecture of our question answering system.

## 3. U-Model

Different from the above two models, the proposed U-model proposes a novel way of exploiting the redundancy of Web information. In brief, our-model can partially resolve the noise problem by incorporating multiple features extracted from the candidate-bearing snippets and treating them discriminately.

Figure 2 illustrates the architecture of our QA system that is a cascade of the following modules.

- **Question Processing** analyzes the given natural language question, identifies the question types (or the desired answer types) and the question focus (e.g., "CEO" in question *Who is the CEO of IBM*). We use some handcrafted heuristic rules to identify question types in English QA system. The SVM-based method proposed by [31] is adopted for Chinese QA. To identify question focus, both Chinese and English QA systems adopt some handcrafted heuristic rules.
- **Document Retrieval** uses keywords from the question to retrieve documents related to the question from a large-scale document set. In our implementation, Indri toolkit (`http://www.lemurproject.org/`) is adopted for document retrieval.
- **Answer Candidate Extraction** extracts answer candidates $\{c_i|i = 1, 2, \ldots, n\}$ from the retrieved documents that match the question types, $n$ is the number of candidates.
- The other three modules, i.e., **Learning Training Examples**, **Training Classifier**, and **Selecting Answer**, are designed to select a candidate answer $c_i^*$ from candidates $\{c_i, i = 1, 2, \ldots, n\}$ as the correct answer to the given question, which are called answer validation (AV).

The AV is the kernel of question answering system and the research focus. We refer to our proposed solution to the AV as the U-model that differs from the previous approaches. In the following sections, we will focus on the details of the U-model.

### 3.1 Main Idea of the U-Model

Given a natural language question $q$ and a set of its candidate answers $\{c_i|i = 1, 2, \ldots, n\}$, the U-model states the problem of selecting a candidate answer $c_i^*$ as the correct answer as follows.

Most sentences express a sequence of sub-topical discussions that can be characterized by highly correlated terms [5]. In this paper, we assume that the $n$ candidates $\{c_i|i = 1, 2, \ldots, n\}$ generated by *the answer candidate extraction module* represent that there exists $n$ topics $\{t_i|i = 1, 2, \ldots, n\}$ related to the given question; and the $n$ candidates are the topic signatures of the $n$ defined topics. Namely, this assumption defines $n$ topics $\{t_i|i = 1, 2, \ldots, n\}$ and their topic signatures $\{c_i|i = 1, 2, \ldots, n\}$.

We assume that there exists an ideal/pseudo answer-bearing snippet to any given question $q$. This ideal/pseudo answer-bearing snippet is also assumed to express a topic (tagged as $t_a$) and the correct answer (tagged as $a^*$) can be used to represent its topic signature. If the topic $t_a$ is same as one of the defined topic $t_k$, it is logical to assume that the topic signature $c_k$ of topic $t_k$ is equal to the topic signature $a^*$ of topic $t_a$, that is, $a^* = c_k$. Therefore, $c_k$ is the correct answer to the question. That is, we define the following inference rule:

$$t_a = t_k \Rightarrow a^* = c_k \tag{2}$$

Accordingly, the AV task becomes the task of classifying the ideal/pseudo answer-bearing snippet into one of $n$ topics $\{t_i|i = 1, 2, \ldots, n\}$.

We employ SVM-based classifier to classify the ideal/pseudo answer-bearing snippet into one of $n$ topics. Unfortunately, there are no training examples for the $n$ defined topics. Here, we present the Algorithm-1 below to learn training examples for each topic.

**Table 2** Some training examples of selected topics.

| topic signature | training examples |
|---|---|
| 1969 | It is believed that the first Crip gang was formed in late 1969. During this time in Los Angeles there were ... |
| | ... the first Bloods and Crips gangs started forming in Los Angeles in late 1969, the Island Bloods sprung up in north Pomona ... |
| | ... formed by 16 year old Raymond Lee Washington in 1969. Williams joined Washington in 1971 ... had come to be called the Crips. It was initially started to eliminate all street gangs ... |
| August 8, 2005 | High Country News – August 8, 2005: The Gangs of Zion |
| 2004 | 2004 main 1 Crips 1.1 FACTOID When was the first Crip gang started? 1.2 FACTOID What does the name mean or come ... |
| 1972 | One of the first-known and publicized killings by Crip gang members occurred at the Hollywood Bowl in March 1972. |
| 1971 | Williams joined Washington in 1971, forming the westside faction of what had come to be called the Crips. |
| | The Crips gang formed as a kind of community watchdog group in 1971 after the demise of the Black Panthers. ... |
| | ... formed by 16 year old Raymond Lee Washington in 1969. Williams joined Washington in 1971 ... had come to be called the Crips. It was initially started to eliminate all street gangs ... |
| 1982 | Oceanside police first started documenting gangs in 1982, when five known gangs were operating in the city: the Posole Locos ... |
| mid-1990s | Street Locos; Deep Valley Bloods and Deep Valley Crips. By the mid-1990s, gang violence had ... |
| 1970s | The Blood gangs started up as opposition to the Crips gangs, also in the 1970s, and the rivalry stands to this day ... |

Algorithm-1

*for each candidate* $c_i$ *do*

Combine candidate $c_i$ and the question keywords $\{q_j | j = 1, 2, \ldots, k\}$ to form a Web search query;

Submit this Web query to a Web search engine and download the top $M$ snippets returned by the search engine;

Retain those snippets $\{s_{i,1}, s_{i,2}, \ldots, s_{i,k_i}\}$ that contain candidate $c_i$ and at least one question keyword as the Web redundancy data of candidate $c_i$, $k_i$ is the number of snippets for $c_i$, and $k_i \leq M$.

*endfor*

For better understanding, Table 2 shows some training examples of selected topics for TREC 2004 test question *When was the first Crip gang started*. Using the training examples $\{s_{i,1}, s_{i,2}, \ldots, s_{i,k_i} | i = 1, 2, \ldots, n\}$ of the topics $\{t_i | i = 1, 2, \ldots, n\}$ learned by the Algorithm-1, we train a $n$-topic classifier. Section 3.2 introduces the classification features. This step is implemented by *the training classifier module*.

*The selecting answer module* constructs the vector of the ideal/pseudo answer-bearing snippet (described in Sect. 3.3) and determines the topic of the ideal/pseudo answer-bearing snippet by using the $n$-topic classifier. The classifier outputs $n < t_k, p_k >$ pairs [†], meaning that the probability of the ideal/pseudo answer-bearing snippet belonging to topic $t_k$ is $p_k$. The U-model finally selects candidate $c_i^*$ with the largest probability as the correct answer.

To summarize, the U-model considers answer validation as a kind of classification task. The hypothesis underlying this model is that candidate-bearing snippets containing the same candidate answer express the same submeaning and thus belong to the same topic. That is, candidates are topic signatures. Similarly, there exists an ideal/pseudo answer-bearing snippet that also expresses a topic that can also be characterized by the answer. Hence, if the ideal/pseudo answer-bearing snippet belongs to one topic, a candidate answer in this topic is the correct answer to the question.

The U-model adopts an SVM as a classifier, and extracts multiple features to form training vectors. For our implementation, we select LIBSVM toolkit (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) and one-against-one strategy for multi-class classification, where the kernel is the radical basis function with the parameter *gamma* = 0.001.

### 3.2 Features

The U-model extracts the following six categories of features for the SVM classifier.

- **Overlap features** include three features, i.e., percentage of matched question keywords (KWs); percentage of mismatched question KWs; and percentage of matched question Bi-grams.
- **Semantic feature** is the percentage of matched thesaurus words. To compute the value of this feature, synonyms and immediate hypernyms of KWs in WordNet are used for English QA system; TONGYICICILIN (a Chinese Thesaurus Lexicon) is employed for Chinese QA system to find synonyms of Chinese words.
- **Boolean features** capture the similarities between the test question and the snippets in terms of certain specific keywords. The Boolean features includes: 1. a focus word [††] or its thesauruses of the test question match or do not match; 2. the test question does or does not match the word, which forms a bi-gram with the candidate contained in the snippet; 3. the capitalized keywords, and time and numeric keywords in the question match or do not match. The value of each feature is computed as,

$$score_{B_i} = \begin{cases} \theta_1 & \text{if } B_i \text{ matches in snppets} \\ 0 & else \end{cases} \quad (3)$$

where $B_i$ means a Boolean feature, $\theta_1$ is set to 0.5 according to the development set.

[†]Usually, support vector classification predicts only class label but not probability information. To extend SVM for probability estimates, the approach proposed in [25] is adopted.

[††]When no focus word can be identified using the heuristic rules, this feature does not fire.

- **Candidate feature** indicates that candidate has or does not have desired answer type. For example, answers to numerical or time questions should contain numeric or time expressions; answers to personal, locational, and organizational questions should contain at least one word starting with a capital letter [†] . The value of this feature is computed as,

$$score = \begin{cases} \theta_1 & \text{if candidate feature fires} \\ 0 & else \end{cases} \quad (4)$$

  where $\theta_1$ is also set to 0.5.
- **Context features** is a set of words preceding $\{w_{i-m}, \ldots w_{i-1}\}$ and following $\{w_{i+1}, \ldots, w_{i+m}\}$ the candidate answer. Each context feature is weighted as,

$$score(w_j, C_i) = \frac{N(w_j, t_i) + \delta}{N(w_j) + \delta} \quad (5)$$

  where $N(w_j)$ is the total number of snippets containing word feature $w_j$, and $N(w_j, t_i)$ is the number of snippets in topic $t_i$ that contain $w_j$, $\delta$ is used for smoothing.
- **Other features** include distance feature (DIST) and frequency feature (FREQ). The DIST denotes the normalized distance between candidates and question keywords in snippets. The value is computed as,

$$score_{DIST} = \frac{\prod_{j=1,\ldots,k} 2^{\frac{1}{1+dist(q_j,c_i)}}}{2^k} \quad (6)$$

  where $dist(q_j, c_i)$ means the number of words between question KW $q_j$ and candidate $c_i$, which is set to the length of the snippet when $q_j$ does not appear, and $k$ is the number of question KWs.
  The FREQ denotes the normalized number of training examples in topic $t_i$ containing candidate $c_i$, which is weighted as,

$$score_{FREQ} = \frac{count(t_i)}{\sum_{t_j} count(t_j)} \quad (7)$$

  where $count(t_i)$ is the number of the learned training examples in topic $t_i$.

### 3.3 Selecting Answer

We assume that the ideal/pseudo answer-bearing snippet contains all of the question words and the words in the context features. Therefore, the values of the matched question KWs and the matched bi-grams in the overlap features, and the semantic feature are set to 1; the mismatched question KWs in the overlap features is set to 0; the values of the context features are set using Eq. (5). Similarly, we assume that all of the Boolean features and the candidate feature fire, thereby their values are set to $\theta_1$. About the other features, we set the values to the maximum estimated in Eq. (6) and (7).

## 4. Experiments and Results

In our experiments, we validate the U-model in terms of English TREC test data sets and Chinese test data sets.

The experimental results are measured in terms of two kinds of scores, *top@n* and *mrr@n*. Here, *top@n* is the rate at which at least one correct answer is included in the top $n$ answers, while *mrr@n* indicates the average reciprocal rank ($1/n$) of the highest rank of the correct answer to each question. In addition, we measure performance by using Ken Litkowski's answer patterns (`http://trec.nist.gov/data/`) and NIST's scoring script. ‡ indicates a statistically significant difference in performance at the 1% level according to a two-sided paired-sample t-test, while † indicates a statistically significant difference at the 5% level.

### 4.1 English Test Sets

The English test data sets consists of factoid test questions from the TREC 2002, 2003, 2005, 2006 QA tracks. The TREC 2001 test questions are used as development data set. For each test question, the candidates come from the TREC QA participants' systems. These candidates are available on the TREC website (`http://trec.nist.gov/data/`). Table 3 summarizes the test sets, including the average number of candidates. Note that NIL questions are excluded from our test sets, because TREC does not supply answer patterns for them, $\#^1$ and $\#^2$ denote the number of test questions and the average number of candidates to each question, respectively. From Sect. 4.1 to Section 4.5, the experiments are based on English test sets.

### 4.2 Overall Performance of the U-Model

Unless specified, otherwise, the search engine is Google. The value of $M$ in the Algorithm-1 of learning training examples (in Sect. 3.1) is very important. If $M$ is too large, the quality of training examples decreases, otherwise, the quantity of training examples can be insufficient. According to our experiments, the best value of $M$ is set to 50. A frequency-based AV approach, selecting candidates as correct answers simply according to the FREQ feature in Sect. 3.2, is implemented as the baseline. Table 4 lists the performance results of the baseline, U-model.

Table 4 demonstrates that the U-model significantly outperforms the baseline. The improvements of all metrics with respect to the baseline system are over 20%, which is

**Table 3** Statistic of English test data sets.

| tracks | TREC02 | TREC03 | TREC05 | TREC06 |
|---|---|---|---|---|
| $\#^1$ | 444 | 380 | 352 | 386 |
| $\#^2$ | 25.0 | 19.1 | 25.1 | 20.7 |

[†]Here, personal, locational and organizational entities are not disambiguated because Named Entity Tagger is not employed.

**Table 4** Performance of U-model.

| | | TREC05 | TREC06 |
|---|---|---|---|
| top@1 | Baseline | 10.2% | 10.9% |
| | U-model | 40.1%‡ | 35.8%‡ |
| mrr@5 | Baseline | 22.7% | 23.6% |
| | U-model | 51.5%‡ | 47.3%‡ |
| top@5 | Baseline | 47.7% | 48.4% |
| | U-model | 71.9%‡ | 66.8%‡ |

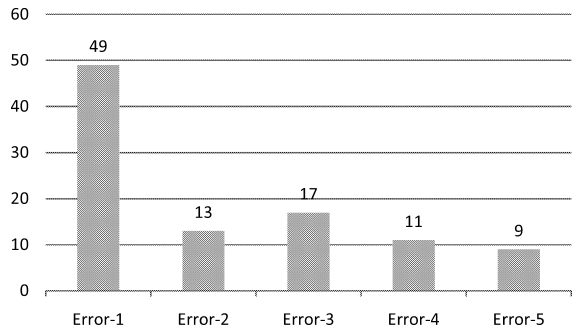


**Fig. 3** Wrongly answered questions.

statistically significant at the 1% level. In fact, there is much noise in the candidates from all of the participants' systems. If more semantic information, like named entities, is incorporated into this architecture, the performances could be further improved.

This table also indicates that 28.1% of the questions are not ranked in the top five on the TREC05 test data set. We analyze the reasons for this failure and classify them into five categories, as showed in Fig. 3.

This figure shows that:

- **Error-1** refers to questions that are probably answered wrongly because of data sparseness. Even though we retrieve candidate-bearing snippets from the Web, data sparseness is still possible. Thirty of the questions classified in **Error-1** have no answer-bearing snippets, while the remaining nineteen questions have less than five answer-bearing snippets.
- The learned answer-bearing snippets by using the Algorithm-1 sometimes express different meanings from their questions. For example, most of the retrieved snippets containing the answer *New York* to the question *Where did Woody Guthrie die* do not express the place of his death. As a result, our model fails to answer this kind of questions, classified as **Error-2**.
- **Error-3** refers to questions wrongly answered because of different phrases in the question and the candidate-bearing snippets. For a question like *What is the company's web address*, most snippets contain answer using expressions like *for more info about this company, visit www.merck.com*. Answering this kind of questions requires incorporating prior knowledge.
- **Error-4** refers to questions that are not correctly answered because the current features cannot distinguish the answers from noise candidates.
- All other wrongly answered questions are classified into **Error-5**.

**Table 5** Contributions of different features.

| | TREC05(%) | TREC06(%) |
|---|---|---|
| overlap | 20.2/33.0/55.4 | 16.3/30.5/53.1 |
| +Boolean | 27.6‡/39.1‡/60.5‡ | 22.0‡/36.4‡/61.4‡ |
| +candidate | 34.4‡/45.9‡/66.8‡ | 26.7‡/40.7‡/64.5† |
| +context | 39.5‡/50.8‡/70.5† | 34.7‡/46.4‡/66.1 |
| +semantic | 40.1/51.5†/71.9† | 35.8/47.3†/66.8 |
| +other | 38.1/49.6†/70.5 | 32.6†/45.8†/66.8 |

### 4.3 Impact of Features

Classification features play very important roles in our model. Table 5 lists the contributions of different features to top@1/mrr@5/top@5 scores by gradually adding them, in order to understand the effectiveness of these features.

This table demonstrates that:

- Using the overlap features only achieves *top*@1/*mrr*@5 scores of 20.2%/33.0% for TREC05 and 16.3%/30.5% for the TREC06.
- The Boolean, candidate, and context features are statistically significant at either the 1% or 5% level, which improves *top*@1 scores by 7.4%, 6.8%, and 5.1%, respectively, for the TREC05, and by 5.7%, 4.7%, and 8.0%, respectively, for the TREC06.
- The semantic feature gives quite limited improvement. For example, adding this feature only increases *top*@1 score for the TREC05 from 39.5% to 40.1%, and top@1 improvement for the TREC06 is only 1.1%. This might be because the semantic feature is incorporated after adding the Boolean, candidate and context features, which dominate the similarities between the snippets and the questions. Consequently, the semantic feature does not largely contribute.
- The other features have negative influences that decrease top@1 scores for TREC05 and TREC06 by 2.0% and 3.2%, respectively. Therefore, the other features will not be used in our final QA systems.

### 4.4 Impact of Search Engines

The above experiments are based on retrieving results from Google. The performance might be not exactly the same if the underlying search engine is changed. To understand this, we compare the performance of the U-model with AltaVista (A, `http://www.altavista.com`), Google (G), and both search engines (G + A). G + A means that the snippets learned from Google and AltaVista are simply combined †. Table 6 summarizes the top@1/mrr@5/top@5 scores.

This experimental result demonstrates that the U-model is independent of the Web search engines, because

†This combination of the snippets does not mean that we simply double the snippets from Google or AltaVista, because the search engines and the snippet-generation approaches used in Google and AltaVista are different.

**Table 6** U-model with different search engines.

| | TREC05 | TREC06 |
|---|---|---|
| G | 40.1/51.5/71.9 | 35.8/47.3/66.8 |
| A | 40.6/51.2/68.8 | 35.0/46.6/67.4 |
| G+A | 41.8/52.4/72.2 | 38.6/48.5/66.1 |

**Table 7** Comparison of redundancy-based models.

| | | top@1 | mrr@5 | top@5 |
|---|---|---|---|---|
| TREC02 | Aranea | 45.0% | 51.2% | 61.7% |
| | Magnini | 34.2%$^{\ddagger}$ | 45.9%$^{\dagger}$ | 66.2% |
| | U-model | 54.7%$^{\ddagger}$ | 65.7%$^{\ddagger}$ | 82.7%$^{\ddagger}$ |
| TREC03 | Aranea | 30.8% | 36.0% | 44.5% |
| | Magnini | 30.5% | 43.9%$^{\ddagger}$ | 66.6%$^{\ddagger}$ |
| | U-model | 43.4%$^{\ddagger}$ | 55.1%$^{\ddagger}$ | 71.1%$^{\ddagger}$ |

the performance does not significantly differ between AltaVista and Google. There is no doubt that the combination of results from different search engines achieves better performance than when the search engines are used individually. The improvement, however, is not statistically significant. This might be because the top 50 results returned by different search engines complement each other to a certain degree.

### 4.5 Cross-Model Comparison

In Sect. 2, we introduced two related redundancy-based AV models. To investigate the effectiveness of these models, we conduct a cross-model comparison of the U-model, the Magnini model, and the Aranea model. [16] provides the source code for the Aranea model, and our experimental setup uses basic.google.nolookup.modules configuration. For the Magnini model, we implement it ourselves. Because the Aranea model cannot be directly evaluated on the TREC05 and 06 series questions, the TREC02 and 03 test data sets are employed in this comparison. The other setups are as follows:

- The U-model and Aranea model are based on retrieval results from Google [†] , while the Magnini model uses AltaVista, which supports the proximity operators *NEAR* and *OR*.
- The U-model and Magnini model adopt candidates from the TREC participants' systems, but the Aranea model does not use these candidates, because it can generate candidates itself. This is the main advantage of the Aranea model over the other models.

Table 7 [††] reports the comparative performance of these redundancy-based AV models. This experiment indicates that:

- The U-model significantly outperforms both the Aranea and Magnini models at the 1% level, with $top@1$ improvements for the TREC03 of 12.6% and 12.9%, respectively.
- The performance of the Aranea and Magnini models do not appear to be consistently stable. For the TREC02, the improvement in $top@1/mrr@5$ score of Aranea over Magnini is statistically significant, while $mrr@5/top@5$ scores of the Magnini model for the TREC03 increase, outperforming the Aranea.
- This comparison proves that our model can resolve the problems of the other models mentioned in Sect. 2. Our model, however, is slower, because we have to train a classifier for each question.

**Table 8** U-model vs. Shen, et al.'s model.

| | | TREC05 | TREC06 |
|---|---|---|---|
| top@1 | Shen, et al. | 24.7% | 20.7% |
| | U-model | 28.1% | 25.6% |
| | TREC-b | 71.3% | 57.8% |
| mrr@5 | Shen, et al. | 32.5% | 26.9% |
| | U-model | 34.6% | 30.2% |
| top@5 | Shen, et al. | 43.5% | 38.6% |
| | U-model | 45.2% | 37.6% |

### 4.6 U-Model vs. Syntax-Based Model

Shen, et al. presented a syntax-based model [9] representing the conventional deep NLP-based technique in the AV task. They did not use any redundancy information and independently estimated the similarities between each candidate-bearing sentence and test question from the viewpoint of a syntax tree. We thus seek to experimentally compare our redundancy-based model with this traditional syntax-based model. To make the comparison more effective, both models use the same candidates provided by [9]. Table 8 reports the results. As a reference, TREC-b (the best TREC QA participant's system) on TREC05 and 06 [11], [13] are also listed.

The $top@1/mrr@5$-score increase of 3.4%/2.1% for the TREC05 and of 4.9%/3.3% for the TREC06 indicate that the U-model is more effective than the model of Shen, et al. This comparison shows that a redundancy-based model can even achieve better performance than that of a conventional NLP-based model if we can effectively mine and use the redundancy information. The TREC-b [6], [24] is a deep NLP-based QA system, which make it outperform all participates's systems and our U-model significantly. However, it is hard for us to follow because the TREC-b incorporates many deep NLP information such as syntactic, semantic information, lexical chain, etc.

### 4.7 Experiments of Chinese Test Sets

To prove that the proposed U-model is a language-independent model, we conduct extensive experiments in terms of Chinese test data sets. In Chinese QA systems, all candidates are extracted automatically using our Chinese

[†]As Sect. 4.4 indicated, our model based on Google is slightly better than that based on AltaVista. While the source code of the Aranea model is based on Google.

[††]We re-run the source code provided by [16], which make the results different from [16].

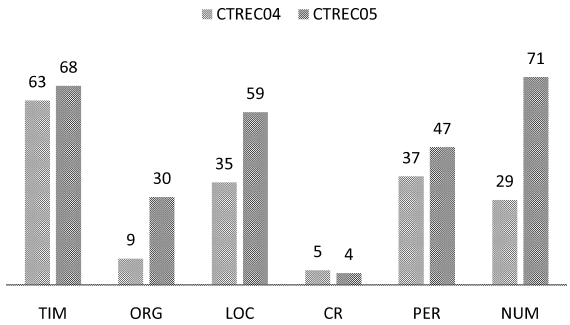


**Fig. 4** Statistics of CTREC04 and CTREC05.

NER tool [32].

Three Chinese data sets, i.e., CTREC04, CTREC05, and CTEST05, are employed. CTREC04 is a set of 178 Chinese questions translated from TREC 2004 FACTOID testing questions. CTREC05 is a set of 279 Chinese questions translated from TREC 2005 FACTOID testing questions. Although the U-model is independent of the question types, for convenience in the answer candidate extraction [32], only those questions whose answers are named entities are selected. Figure 4 breaks down the types of questions (manually assigned) in the CTREC04 and CTREC05 data sets. Here, PER, LOC, ORG, TIM, NUM, and CR refer to questions whose answers are a person, location, organization, time, number, and book or movie, respectively. CTEST05 is a set of 178 Chinese questions found in [30] that are similar to TREC testing questions except that they are written in Chinese.

### 4.8 U-Model vs. S-SVM

As we mentioned, our U-model is a kind of unsupervised algorithm for AV. This U-model differs from the supervised techniques for AV [1], [2], [12], [15], [18], [19], [27], in which a large number of hand-tagged training <question, answer-bearing snippet> pairs are required. This experiment is to compare our unsupervised U-model with supervised model (S-SVM) in terms of the CTREC04 and CTREC05 test sets.

To collect <question, answer-bearing snippet> training data for the S-SVM, we submitted 807 Chinese questions to Google and extracted the candidates for each question from the top 50 Google snippets. We then manually selected the snippets containing the correct answers as positive snippets, and designated all of the other snippets as negative snippets. Finally, we collected 807 hand-tagged Chinese <question, answer-bearing snippet> pairs as the training data of S-SVM called CTRAINDATA. The training data includes 140 LOC, 251 PER, 45 ORG, 90 NUM, and 281 TIM questions.

To explore the effectiveness of our unsupervised model as compared with the supervised model, we conduct a cross-model comparison of the S-SVM and the U-model. Note that the context and other features are hard to incorporate into the S-SVM [2], [18], therefore, this comparison is based on the overlap, semantic, Boolean, and candidate features

**Table 9** Comparison of U-model and S-SVM on the CTREC04 and CTREC05.

| | | CTREC04 | CTREC05 |
|---|---|---|---|
| top@1 | S-SVM | 39.18% | 33.33% |
| | U-model | 53.61% | 50.00% |
| mrr@5 | S-SVM | 53.54% | 48.67% |
| | U-model | 66.25% | 62.38% |
| top@5 | S-SVM | 79.38% | 74.67% |
| | U-model | 88.66% | 82.67% |

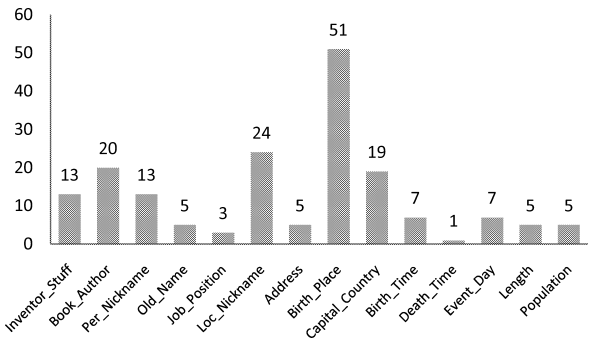


**Fig. 5** Statistics of CTEST05.

described in Sect. 3.2. The U-model results are compared with the S-SVM results for the CTREC04 and CTREC05 in Table 9. The S-SVM is trained on CTRAINDATA. This table shows that the proposed U-model significantly outperforms the S-SVM for all measurements and all test data sets. For the CTREC04, the $top@1$ improvement is about 14.4%. For the CTREC05, the $top@1$ score increases from 33.3% to 50.0%. In the S-SVM, all questions share the same training data, while the U-model uses the unique training data learned by the Algorithm-1 (in Sect. 3.1) for each question. This is the main reason why the U-model performs better than the S-SVM does.

### 4.9 U-Model vs. Pattern-M vs. S-SVM

The pattern-based models (Pattern-M) [8], [21], [26] are also widely used for QA system. This approach first classifies the question into predefined categories, and then validates whether the extracted candidates are the exact answers or not by using answer patterns learned off-line. The pattern-based model can obtain high precision for some predefined types of questions, it is difficult to define question types in advance for open-domain question answering. The CTEST05 contains 14 different question types that are suitable for the pattern-based models. Figure 5 lists the statistic of the CTEST05, and Table 10 gives examples of each type of questions, which is labeled manually.

The Pattern-M uses the dependency syntactic answer patterns learned in [28] to extract the answer. Each syntactic pattern is associated with a probability indicating the precision of the pattern. When multiply answers are matched by the patterns, the probability of the pattern is used to rank them.

This experiment is to compare the U-model with the

**Table 10** Examples of questions in the CTEST05.

| Question type | Examples |
|---|---|
| Inventor_Stuff | Who invented telephone? |
| Book_Author | Who is the author of Harry Potter? |
| Per_Nickname | Who is the father of music? |
| Old_Name | What is the full name of Newton? |
| Job_Position | Who is the CEO of Microsoft? |
| Loc_Nickname | What is the alias of New York State? |
| Address | Where is the Statue of Liberty? |
| Birth_Place | Where was Michael Jordan born? |
| Capital_Country | Which city is the capital of Japan? |
| Birth_Time | When was Michael Jordan born? |
| Death_Time | When did Marilyn Monroe die? |
| Eventday | When is World Day for Water? |
| Length | What is the height of Mount Everest? |
| Population | What is the population of china? |

**Table 11** Comparison of U-model, Pattern-M and S-SVM on CTEST05.

| | S-SVM | Pattern-M | U-model |
|---|---|---|---|
| $top@1$ | 44.89% | 53.14% | 59.09% |
| $mrr@5$ | 56.49% | 61.28% | 67.34% |
| $top@5$ | 74.43% | 73.14% | 81.82% |

Pattern-M and the S-SVM in terms of the CTEST05 data set. Table 11 summarizes the performances of the U-model, Pattern-M, and S-SVM models on the CTEST05. The results in the table show that the U-model significantly outperforms the S-SVM and Pattern-M, while the S-SVM underperforms the Pattern-M. Compared with the Pattern-M, the U-model increases the $top@1/mrr@5/top@5$ scores by 5.95%/ 6.06%/8.68%, respectively. The reasons may lie in the following:

- The Chinese dependency parser influences dependency syntactic answer-pattern extraction, and thus degrades the performance of the Pattern-M model.
- From the cross-model comparison, we conclude that the performance ranking of these models is: U-model > Pattern-M > S-SVM.

## 5. Discussion

Even though the U-model can achieve satisfying performance, there are some problems with this framework. First, the training data for the SVM classifier are learned automatically by the Algorithm-1 in Sect. 3.1, which inevitably introduces noise that can negatively influence the performance. Second, we have to train the classifier for each test question, which can result in a heavily time-consuming model. This section discusses these two problems.

### 5.1 Noise Reduction

Some candidates $\{c_j\}$ cannot be used to represent topics, in which cases the corresponding topics $\{t_j\}$ are noise, called topic noise. To reduce this noise, such candidates must be filtered out. The second noise is snippet noise, meaning that snippets in correct topics are noise. For the example in Table 1, not all snippets containing the candidate *1969* express the time of establishment of *the first Crip gang*. We thus conduct an experiment to examine the impacts of these noise.

**Table 12** U-model after noise reduction.

| | *topic noise reduction* | *snippet noise reduction* |
|---|---|---|
| *top@1* | 52.8% | 38.9% |
| *mrr@5* | 65.8% | 50.2% |
| *top@5* | 86.9% | 70.1% |

To overcome topic noise, candidates from the TREC top ten participants' systems are adopted. As compared with the candidates from all participants' systems, the noise in the candidates from the top ten systems is greatly reduced, because the average number of candidates in TREC05 is 5.8. Actually, it is impossible to have candidates from top 10 groups for any new test question, this experiment, however, helps us find the future work.

To reduce snippet noise, we adopt one additional condition in the Algorithm-1, requiring that the snippets must include at least one capitalized question keyword or time or numeric keyword when the question contains one of them. With this restriction, the problem of snippet noise can be partially resolved.

Table 12 reports the performance of our model on TREC05 after reducing the topic noise and snippet noise. Comparing this table with Table 4, we observe that the performance on the snippet-noise-reduced data falls off somewhat, because 14 questions' answers are filtered out from candidates. Before the snippet noise reduction, 141 questions are correctly answered; however, only 137 questions are correctly answered after snippet noise is reduced. This means that although some noise can be removed, at the same time, correct answers can also be filtered out. Perhaps snippet noise does not greatly, adversely impact the performance, at least reducing snippet noise in a simply way is certainly not helpful. Table 12 also shows that topic noise greatly impacts the performance. Reducing of the topic noise sharply increases top@1 score from 40.1% to 52.8%, and the improvements in mrr@5/top@5 scores are 14.3%/15.0%.

### 5.2 Time Problem

The experiment investigating how much time our model requires shows that an average time per question used in training the classifier is about 3.5 seconds. Compared with NLP-based approaches, the U-model is not a time-consuming model. This is because our model does not require any deep natural language processing. Moreover, training data is not huge: about 20 candidates multiplied by 25 training instances for each question (the average $k_i$ in the Algorithm-1 is 25).

## 6. Conclusion

Given the observation that candidate-bearing snippets with the same candidate usually express the same sub-meaning,

this paper has presented an unsupervised SVM classifier for the AV task. The essence of our approach is to exploit data redundancy information from the Web for the AV task. Our model, called U-model, achieves satisfactory performance on TREC test sets. Cross-model experiments indicate that the U-model outperforms the related redundancy-based models with statistical significance. The top@1/mrr@5/top@5 improvements over the Aranea model and the Magnini model on TREC03 are 12.6/19.1/21.0% and 12.9/11.2/4.5%, respectively. Moreover, our model achieves better performance than do the syntactic-based model, a supervised machine learning model, and a pattern-based model. From the results reported in this paper, we can conclude that effectively exploiting redundancy information can greatly improve the performance of an AV task.

In our future work, we plan to improve the U-model by removing noise in the training data, which is the main problem of this architecture. Furthermore, we intend to adapt our model for other types of questions, such as definition and biography questions.

**References**

[1] A. Echihabi and D. Marcu, "A Noisy-channel approach to question answering," Proc. ACL-2003, 2003.
[2] A. Ittycheriah and S. Roukos, "IBM's statistical question answering system-trec 11," Proc. TREC-11.
[3] B. Magnini, M. Negri, R. Prevete, and H. Tanev, "Is it the right answer? Exploiting web redundancy for answer validation," Proc. ACL-2002, pp.425–432, 2002.
[4] C.L.A. Clarke, G.V. Cormack, and T.R. Lynam, "Exploiting redundancy in question answering," Proc. SIGIR-2001, pp.358–365, 2001.
[5] C.Y. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," Proc. COLING-2000, pp.495–501, 2000.
[6] D. Moldovan, M. Bowden, and M. Tatu, "A temporally-enhanced PowerAnswer in TREC 2006," Proc. TREC-2006, 2006.
[7] D. Moldovan, S. Harabagiu, and G. Roxana, "LCC tools for question answering," Proc. TREC-2002, 2002.
[8] D. Ravichandran and E. Hovy, "Learning surface text patterns for a question answering system," Proc. ACL-2002, 2002.
[9] D. Shen and D. Klakow, "Exploring correlation of dependency relation paths for answer extraction," Proc. COLING/ACL-2006, pp.889–896, 2006.
[10] E. Hovy, U. Hermjakob, and C.Y. Lin, "The use of external knowledge of factoid QA," Proc. TREC-2001, 2001.
[11] E.M. Voorhees and H.T. Dang, "Overview of the trec 2005 question answering track," Proc. TREC-2005, 2005.
[12] H.T. Ng, J.L.P. Kwan, and Y. Xia, "Question answering using a large text database: A machine learning approach," Proc. EMNLP-2001, pp.66–73, 2001.
[13] H.T. Dang, J. Lin, and D. Kelly, "Overview of the trec 2006 question answering track," Proc. TREC-2006, 2006.
[14] H. Yang and T.S. Chua, "Qualifier: Question answering by lexical fabric and external resources," Proc. EACL-2003, pp.363–370, 2003.
[15] I. Zukerman and E. Horvitz, "Using machine learning techniques to interpret WH-questions," Proc. ACL-2001, pp.547–554, 2001.
[16] J. Lin, "An exploration of the principles underlying redundancy-based factoid question answering," ACM Trans. Information Systems, vol.27, no.2, pp.1–55, 2007.
[17] J. Lin and B. Katz, "Question answering from the web using knowledge annotation and knowledge mining techniques," Proc. CIKM-2003, pp.116–123, 2003.
[18] J. Suzuki, Y. Sasaki, and E. Maeda, "SVM answer selection for open-domain question answering," Proc. COLING-2002, pp.974–980, 2002.
[19] L.V. Lita and J. Carbonell, "Instance-based question answering: A data-driven approach," Proc. EMNLP-2004, pp.396–403, 2004.
[20] M. Pasca, "A relational and logic representation for open-domain textual question answering," Proc. ACL (Companion Volume) 2001, pp.37–42, 2001.
[21] M.M. Soubbotin and S.M. Soubbotin, "Use of patterns for detection of likely answer strings: A systematic approach," Proc. TREC-2002, 2002.
[22] M. Kaisser, S. Scheible, and B. Webber, "Experiments at the University of Edinburgh for the TREC 2006 QA track," Proc. TREC-2006, 2006.
[23] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng, "Web question answering: Is more always better?," Proc. SIGIR-2002, pp.291–298, 2002.
[24] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang, "Employing two question answering systems in TREC-2005," Proc. TREC-2005, 2005.
[25] T.F. Wu, C.J. Lin, and R.C. Weng, "Probability estimates for multi-class classification by pairwise coupling," J. Machine Learning Research, vol.5, pp.975–1005, 2004.
[26] Y. Du, X.J. Huang, X. Li, and L.D. Wu, "A novel pattern learning method for open domain question answering," Proc. IJCNLP-2004, pp.81–89, 2004.
[27] Y. Sasaki, "Question answering as question-biased term extraction: A new approach toward multilingual QA," Proc. ACL-2005, pp.215–222, 2005.
[28] Y. Wu, H. Kashioka, and J. Zhao, "Using clustering approaches to open-domain question answering," Proc. CICLING-2007, pp.506–517, 2007.
[29] Y. Wu, R.Q. Zhang, X.H. HU, and H. Kashioka, "Learning unsupervised SVM classifier for answer selection in question answering," Proc. EMNLP-2007, pp.33–41, 2007.
[30] Y. Wu, J. Zhao, X.Y. Duan, and B. Xu, "Building an evaluation platform for chinese question answering systems," Proc. First NCIRCS, 2004. (in Chinese)
[31] Y. Wu, J. Zhao, and B. Xu, "Chinese question classification from approach and semantic view," Proc. 2nd Asia Information Retrieval Symposium (AIRS-2005), LNCS 3689, pp.485–490, 2005.
[32] Y. Wu, J. Zhao, B. Xu, and H. Yu, "Chinese named entity recognition model based on multiple features," Proc. HLT/EMNLP-2005, pp.427–434, 2005.

**Youzheng Wu** received his B.S. and M.S. degrees from Wuhan University, China in 1999 and 2002, respectively, received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, China in 2006. He worked as a researcher in SLC Group, ATR, Japan from Jun. 2006 to Mar. 2009. He is currently a researcher of NiCT, Japan. His research interests are related to natural language processing, question answering, and information extraction.

**Hideki Kashioka** received his Ph.D. in Information Science from Osaka University in 1993. From 1993, he works for ATR Spoken Language Translation Research Laboratories. He is currently the head of department of Spoken Language Research in ATR Spoken Language Communication Research Laboratories. He is also the research manager of Spoken Language Communication Group at Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology. He is also the visiting associate professor of the graduate school of Information Science at the Nara Institute of Science and Technology from 1999. He is a member of ANLP, JCSS, JSAI and IPSJ.

**Satoshi Nakamura** received his B.S. in Electronic Engineering from the Kyoto Institute of Technology in 1981 and his Ph.D. in Information Science from Kyoto University in 1992. From 1981 to 1993, he worked for Sharp's Central Research Laboratory in Nara. From 1986 to 1989, he worked for ATR Interpreting Telephony Research Laboratories. From 1994 to 2000, he was the associate professor of the graduate school of Information Science at the Nara Institute of Science and Technology. In 1996, he was a visiting research professor of the CAIP center at Rutgers University in New Jersey. He is currently the director of the MASTAR project at Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology. He also serves as an honorary professor of University Karlsruhe, Germany since 2004. He received the Awaya Award from the Acoustical Society of Japan in 1992, the Interaction 2001 Best Paper Award in 2001, Yamashita Research Award from the Information Processing Society of Japan in 2001, Telecom System Award, AAMT Nagao Award and Docomo Mobile Science Award in 2007. He served as an associate editor for the Journal of the IEICE ED from 2000 to 2002, a member of the Speech Technical Committee of the IEEE Signal Processing Society in 2001–2004, a general chair of International Workshop of Spoken Language Translation (IWSLT2006) and Oriental Cocosda 2008, and a technical chair IEEE ASRU2007 and INTERSPEEC 2010. He is a member of IEEE, IPSJ and ASJ.