SVM Answer Selection for Open-Domain Question Answering

Jun Suzuki, Yutaka Sasaki, and Eisaku Maeda
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
{jun, sasaki, maeda}@cslab.kecl.ntt.co.jp

Abstract

This paper presents an answer selection method based on *Support Vector Machines (SVM)* for Open-Domain Question Answering (QA). Selecting and ranking plausible answers from a large number of candidates in documents is one of the most critical parts of QA systems. It is extremely difficult to find good evaluation functions or rules for the answer selection. To overcome this issue, we apply SVM to answer selection. We evaluate the performance measured by mean reciprocal rank (MRR) and the correct ratio of answer ranked first. The results show that the proposed SVM-based method offers a statistically significant increase in performance compared to other machine learning methods such as decision tree learning (C4.5) boosting with decision tree learning (C5.0), and the maximum entropy method.

1 Introduction

Question Answering (QA) involves the extraction of answers to a question from large-scale documents. For instance, if a QA system is given the question "When was Queen Victoria born?", it should answer "1832". Question Answering has been studied intensely all over the world since the start of the Question Answering Track at TREC-8 (1999).

The definition of QA tasks at the TREC QA-Track has been revised and extended year after year. At first, QA research focused on the Passage Retrieval method as used at TREC-8. That is, the QA task was to answer a question in the form of strings of 50 bytes or 250 bytes excerpted from a large set of news wire articles. Recently, however, the QA task is considered to be to extract *exact answers* to a question.

Typically, question answering systems use the following components:

Question analysis analyzes a given question sentence and determines the question type and keywords. In addition, some systems find the *question focus* of a given question.

Text retrieval finds the top N paragraphs (or documents) that match the output of question analysis, such as keywords and question types.

Answer candidate extraction extracts answer candidates from the relevant documents retrieved by the text retrieval component.

Answer selection selects answers to the question from among the answer candidates based on the result of question analysis.

We have studied a Japanese QA system SAIQA (Sasaki et al., 2001), which has above four components and an article summarization module to provide a justification of the answer. Now we are developing a trainable QA system SAIQA-II based on *Support Vector Machines (SVM)* technique (Sasaki et al., 2002; Hirao et al., 2002; Isozaki and Kazawa, 2002).

This paper focuses on the answer selection part and proposes SVM based answer selection method. In answer selection, selecting and ranking plausible answers from a large number of candidates is the key to success. It is, however, very difficult to find good evaluation functions or rules that work well in all fields because there are many system parameters that must be carefully tuned in order to achieve good answer selection.

Our solution is to apply SVM to determine the best answer selection function. The SVM has achieved high performance in the fields of Natural Language Processing, such as chunking(Kudo and Matsumoto, 2001) and text cat-

egorization(Joachims, 1998; Taira and Haruno, 1999).

We utilize a QA test collection of Japanese question sentences whose answers are named entities (exact answers), such as dates and person names. While Japanese is an agglutinative language such as Chinese and Thai, the exact definition of a named entity has been already discussed and defined in IREX[1].

Before we present the SVM approach, we define the question answering task addressed in this paper.

Basically, our question answering system follows the definition set by the TREC QA-Track. In addition, we adopt some additional conditions to evaluate the answer selection part.

1. Answers to questions are named entities.
2. The answer exists in at least one of the documents in the set.

In the following sections, we will show how to apply SVM to answer selection and its performance. It is compared against the baseline method and other machine learning approaches (decision tree learning, boosting with decision tree learning and maximum entropy method).

2 Support Vector Machines (SVM)

2.1 Key Ideas of SVM

This section briefly introduces the machine learning methodology of Support Vector Machines (Vapnik, 1995; Cortes and Vapnik, 1995).

SVM offer the following advantages over conventional statistical learning algorithms (i.e., decision tree learning, maximum entropy method):

1. high generalization performance even with feature vectors of high dimension, and
2. the ability to manage *kernel functions* that map input data to higher dimensional space without increasing computational complexity.

The explanation of SVM starts with a set of l training data $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)$ where $\mathbf{x}_i (\in \mathbf{R}^n)$ is an n-dimensional vector and $y_i (\in \{+1, -1\})$ is the class label of i-th data.

[1]Information Retrieval and Extraction Exercise, http://cs.nyu.edu/cs/projects/proteus/irex/

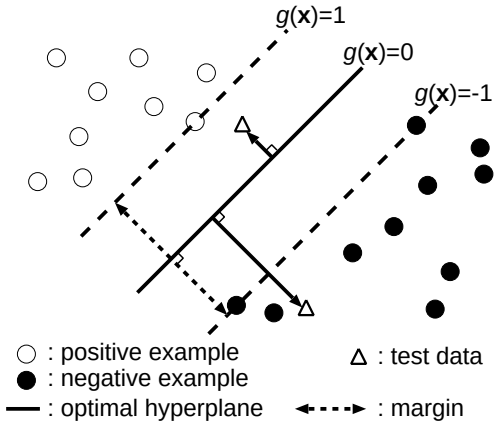

Figure 1: Support Vector Machines

The optimal hyper-plane $\mathbf{w} \cdot \mathbf{x} + b = 0 (\mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R})$ separates the training data into two classes. The basic idea of SVM is to maximize the margin between the positive and negative examples. Figure 1 shows training examples linearly separated into two classes.

In general, it is not necessary to separate training examples into each class. Variable $\eta_i (\geq 0)$ is introduced for misclassification errors. This optimization problem is defined as follows:

$$\min_{\mathbf{w},b,\eta} \quad : \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\eta_i \tag{1}$$

$$\text{s.t.} \quad : \quad y_i[(\mathbf{w} \cdot \mathbf{x}) + b] \geq 1. \tag{2}$$

The first term in equation (1) specifies the size of the margin and the second term represents the cost of the misclassification.

The decision function $f(\mathbf{x})$ can be written as:

$$f(\mathbf{x}) = \text{sgn}(g(\mathbf{x})) \tag{3}$$

$$g(\mathbf{x}) = \sum_{i}^{l}\alpha_i y_i(\mathbf{x}_i \cdot \mathbf{x}) + b. \tag{4}$$

We calculate the *kernel function* defined as $\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = K(\mathbf{u}, \mathbf{v})$ for a non-liner SVM classifier. Using a Kernel function, we can rewrite equation (4) as:

$$g(\mathbf{x}) = \sum_{i}^{l}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{5}$$

2.2 Application to QA Tasks

From the viewpoint of machine learning, answer selection is defined as the task of training and classifying *answer candidates* into positives (correct answers) and negatives (incorrect answers) for a given question.

To apply SVM, we have to prepare a set of training examples that contain feature vectors $\mathbf{x}_i$ of i-th answer candidates. For each question, a QA system analyzes the question, retrieves documents related to the question, then lists the answer candidates. The system parameters that were computed in the process are recorded and used to create feature vectors for each answer candidate.

We used the $g(\mathbf{x})$ in equation (5) to rank the answer candidates. $g(\mathbf{x})$ represents the *distance* of $\mathbf{x}$ from the optimal hyper-plane normalized by the margin (Figure 1).

3 Feature Vectors

3.1 Preparation for Feature Extraction

As with typical QA systems, question analysis, text retrieval and answer candidate extraction are performed before the feature extraction. The following three steps are performed:

(1) For each question, the question analysis module analyzes the question and obtains keywords (KW), question types (QT), a question focus (QF), numerical units (QU) such as "litter" or "piece", and auxiliary terms (AT) which is quoted term and a sequences of *katakana* words in Japanese[2].

(2) The text retrieval module collects the documents that contain at least one keyword or named entity.

(3) All named entities that match the question types are extracted from the retrieved documents.

After these steps, the features for the answer selection are extracted using the results of question analysis (1) and the extracted answer candidates (3).

3.2 Method of Feature Extraction

Tables 1 and 2 show the features used in this paper. Table 1 shows the features extracted using the *window function*, explained in Section 3.2.1. Table 2 shows the remaining features.

[2]Foreign words are expressed in katakana in Japanese.

3.2.1 Features Extracted by Window Function

The answer selection module calculates several parameters using a *window function* that we define based on the phrase (*bunsetsu*) unit, the sentence unit, and the paragraph unit (Table 1).

Let D be a document. Let w_i, b_i, s_i, and p_i be sequentially numbered words, *bunsetsu*, sentences, and paragraphs in D, respectively.

We define *window function* $W^{\Psi}_{\Delta}(k)$ with meta-variable $\Psi \in \{p, s, b, w\}$ as:

$$W^{\Psi}_{\Delta}(k) = \bigcup_{i=k-\Delta}^{k+\Delta} \Psi_i. \quad (6)$$

Here, we treat w_i as a singleton set. The window function forms a set of words of interest for subsequent processing.

Figure 2 shows an example of the window function. For example, $W^{s}_{2}(k)$ includes all words in the region from the $(i-2)$-th sentence to the $(i+2)$-th sentence.

We use $\Delta = 1, 5, \ldots, 20$ (every five) for *bunsetsu* analysis, $\Delta = 0, 1, 2, 3$ for the sentence analysis, and $\Delta = 0$ for the paragraph analysis. This is necessary due to the analysis error of sentence boundaries and parts of speech, and the difficulty of context understanding or semantic analysis in Natural Language Processing.

3.2.2 Other Extraction Functions

Some parameters are real numbers (type *real* in Tables 1 and 2 in the range of 0 to 1 after normalization. These real values are quantized into five bins. Integer values (type *int*) are also quantized into five bins. The bin widths are uniform. As a result of this operation, our feature vectors contain only boolean features.

We use the semantic categories of the semantic attribute system of "Goi-Taikei — A Japanese Lexicon" (Ikehara et al., 1997). This semantic attribute system is used for calculating the similarity of keywords (SC in Table 1). Thus, semantic relations are included in the feature vector.

4 Experimental Settings

4.1 Evaluation Method

We adopted a recent standard style used by TREC QA-Track, and call this style answer

Table 1: Feature Extraction from Answer Candidate and Question using Window Function

class	feature	type	W^c_Δ	W^s_Δ	W^p_Δ	W^s_Δ+Headline
keywords (KW)	ave. # of stem match	real	○	○	○	○
	ave. # of inflection match	real	○	○	○	○
	ave. # of Part-of-Speech match	real	○	○	○	○
	ave. # of head word match	real	○	○	○	○
	ave. # of function word match	real	○	○	○	○
	all KW match	bool		○	○	
	ranking KW match	int		○	○	
semantic category (SC) of KW	ave. # of category match	real	○	○	○	○
named entities (NE)	ave. # of type match	real	○	○	○	○
	ave. # of entity match	real	○	○	○	○
	all NE match	bool		○	○	
	ranking NE match	int		○	○	
auxiliary terms (AT)	entity match	bool		○	○	
question focus (QF)	entity match	bool		○	○	

Table 2: Feature Extraction from Answer Candidate and Question

class	feature	type
answer candidate (AC)	word length	int
	normalized position in the document	real
	matching with Part-of-Speeches	bool
	matching with attached function words	bool
numerical unit (QU)	entity	bool
question type (QT)	pair of QT and NE of AC	bool

ranking. The evaluation measure for answer ranking is the mean reciprocal rank (MRR), which is the same as that used by TREC QA-Track(1999). This score is simply the rank position of the first correct answer. If the first correct answer is ranked n, the score is $1/n$ (first = score 1, second = score 1/2, ..., fifth = score 1/5).

The evaluation method used is ten fold cross validation, divided into ten sets. Nine sets were used for training and the remaining one for testing. In addition, the ratio of each category in each set is even.

4.2 Data Set

The following experiments used the QA test collection constructed in (Sasaki et al., 2001) as the data sets.

The style of questions is almost the same as the TREC QA-Track style, except all questions are written in Japanese.

All questions used in the evaluation have at least one correct answer in the retrieved documents (some questions had no answers because of erroneous named entities analysis or text retrieval). This allows us to evaluate just the Answer Selection parts that we focus on.

The number of questions for each question type used is shown in Table 3. The category

Table 3: The Number of Questions in Each Category

Question Type	# of Questions
PERSON	283
ORGANIZATION	264
LOCATION	244
DATE	311
others(25 categories)	256
Total	1358

"others" includes 25 categories such as TIME, MONEY, PERCENT, and AGE.

We also evaluated certain categories, PERSON, ORGANIZATION, LOCATION and DATE, considering only questions in the target category.

The average number of answer candidates for each question was 145.12; the number of correct answer candidates was one or a few. The feature vector consisted of 3081 features.

4.3 Comparison Methods

To estimate the performance of our answer selection method, we compared it against one baseline method and three machine learning methods as described below.

(1) Baseline (BL)

Ranking score $RS = \sum_i^N \frac{1}{D(AC,KW_i)}$ is cal-

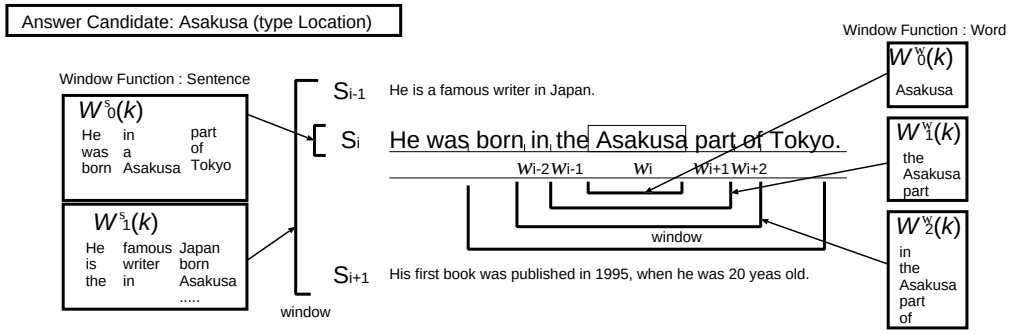

Figure 2: Example of Window Function

culated from the sum of the distance between the answer candidate and each keyword.

(2) Decision Tree Learning (C4.5)
The learning algorithm is C4.5 (Quinlan, 1993) and the default values for learning parameters were used. The evidence value was calculated for the purpose of ranking.

(3) Boosting with Decision Tree Learning (C5.0)
The learning algorithm is C5.0 with 100-rounds boosting (Freund and Shapire, 1996). The evidence value was calculated for the ranking as well as C4.5.

(4) Maximum Entropy Method (Berger et al., 1996) (ME)
The ranking scores were equated to the probability of answer correctness.

(5) Support Vector Machines (SVM)
We selected the 2-degree polynomial kernel because of its excellent performance in preliminary experiments that considered $1, 2, 3, 4, 5$-degree polynomial kernels and $\gamma = 0.0001, 0.001, 0.1, 1$ RBF kernels.

The same question analysis module and text retrieval module were used in all methods; the machine learning methods used the same feature vectors.

5 Results

Figure 3 shows the results of answer selection. The error bars represent the standard deviation between each fold.

The results of evaluating all questions and every category show that SVM answer selection offers the best performance. The results of the baseline method indicate that category PERSON probably includes easier questions since questions against this category can be found by using just keywords.

We performed a statistical significance test based on Tukey's multiple comparison method on the machine learning methods. The significance test results are shown in Table 4. The asterisk $(*)$ represents a 5% significant difference between methods, while $(**)$ represents a 1% significant difference.

Our approach became statistically different (superior) from the other methods in most of comparisons.

Figure 4 shows the ratio of questions where the answer ranked first is the correct answer. SVM answer selection offers the best performance as the same as MRR results.

The performance in providing the correct answer at the first run would be seen as more important than ranking answers. In fact, the QA task is considered to be the task of providing *exact answers* instead of ranked answers.

In addition, the performance of SVM answer selection was compared to the hand-crafted ranking function of a QA system (Sasaki et al., 2001). The SVM answer selection method had a 0.446 point higher MRR value.

6 Discussion and Related Work

In the TREC QA-Track, only a few systems took the machine learning approach (Ittycheriah et al., 2001). This system demonstrates answer selection with the maximum entropy method using 168 features after feature selection, and so it shares our view of using machine learning.

Other research on adapting machine learning for answer selection was undertaken by (Ng et

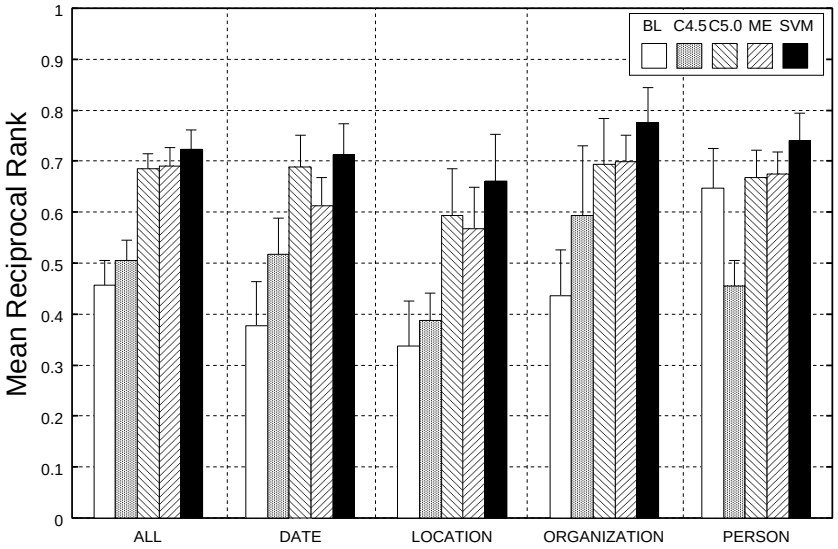

Figure 3: Answer Selection Performance Measured by Mean Reciprocal Rank (MRR)

Table 4: Significant Difference in MRR using Tukey's Multiple Comparison Method

	C5.0	ME	SVM
C4.5	**	**	**
C5.0		–	*
ME			–

(a) all

	C5.0	ME	SVM
C4.5	**	**	**
C5.0		**	–
ME			**

(b) DATE

	C5.0	ME	SVM
C4.5	**	**	**
C5.0		–	–
ME			*

(c) LOCATION

	C5.0	ME	SVM
C4.5	**	**	**
C5.0		–	*
ME			–

(d) ORGANIZATION

	C5.0	ME	SVM
C4.5	**	**	**
C5.0		–	*
ME			–

(e) PERSON

al., 2001), using four features for the feature vector and C5.0 for the learning algorithm.

In contrast to their work, this paper adopted the SVM instead of the maximum entropy method or C5.0, as well as a different method of feature extraction, which suits the SVM, that offers good performance against large scale features. In applying SVM, the strategy of feature extraction can be different from the other methods. It is a better approach to place many features in the feature vectors since answer selection is a very complicated and sensitive process.

7 Conclusion

This paper presented a question answering system based on Support Vector Machines that offers high performance in answer selection. Answer selection experiments were conducted on 1358 questions.

The experimental results showed that the proposed SVM answer selection method had statistically better performance compared to other machine learning methods such as decision tree learning (C4.5), boosting with decision tree learning (C5.0), and maximum entropy method.

Acknowledgement

We would like to thank all the members of the Knowledge Processing Research Group for valuable comments and discussions.

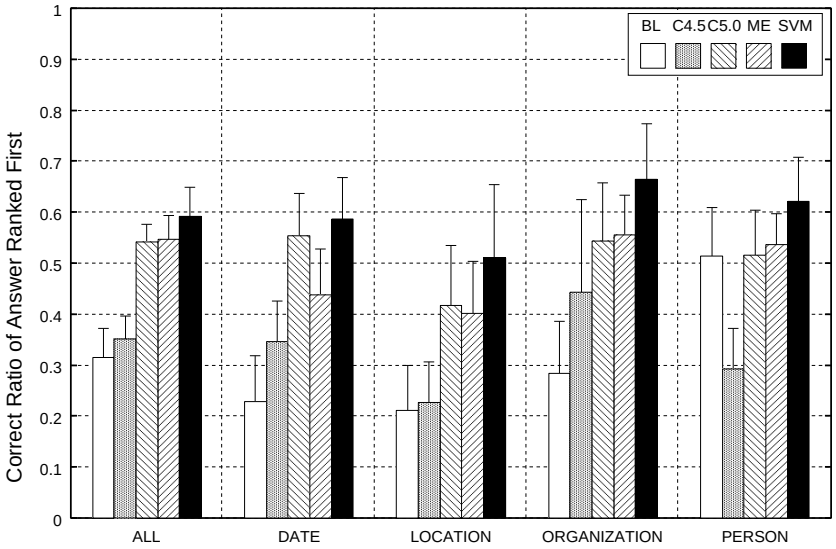

Figure 4: Answer Selection Performance Measured by Correct Ratio of Answer Ranked First

References

A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(2):39–71.

C. Cortes and V. M. Vapnik. 1995. Support Vector Networks. *Mahine Learning*, 20:273–297.

Y. Freund and R. E. Shapire. 1996. Experiments with A New Boosting Algorithm. *Proc. of ICML'96*, pages 148–156.

T. Hirao, H. Isozaki, E Maeda, and Y. Matsumoto. 2002. Extracting Important Sentences with Support Vector Machines. *Coling-2002*.

S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Oyama, and Y. Hayashi, editors. 1997. *The Semantic Attribute System,* Goi-Taikei — A Japanese Lexicon, volume 1. Iwanami Publishing. (in Japanese).

H. Isozaki and H. Kazawa. 2002. Efficient Support Vector Classifiers for Named Entity Recognition. *Coling-2002*.

A. Ittycheriah, M. Franz, and S. Roukos. 2001. IBM's Statistical Question Answering System – TREC-10. *Proc. of TREC-10*.

T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proc. of ECML'98*, pages 137–142.

T. Kudo and Y. Matsumoto. 2001. Chanking with Support Vector Machines. *Proc. of NAACL 2001*, pages 192–199.

H. T. Ng, J. L. P. Kwan, and Y. Xia. 2001. Question Answering Using a Large Text Database: A Machine Learning Approace. *Proc. of EMNLP 2001*, pages 67–73.

NIST. 1999. *Eighth Text REtrieval Conference (TREC-8)*. NIST.

J. R. Quinlan. 1993. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann.

Y. Sasaki, H. Isozaki, H. Taira, T. Hirao, H. Kazawa, J. Suzuki, K. Kokuryo, and E. Maeda. 2001. SAIQA: A Japanese QA System Based on a Large-Scale Corpus. *SIG-Notes Fundamental Infology of Information Processing Society of Japan*, pages 77–82. (in Japanese).

Y. Sasaki, K. Kokuryou, H. Isozaki, T. Hirao, J. Suzuki, and E. Maeda. 2002. SAIQA-II: A Trainable Question Answering System. *Demonstrations at Coling-2002*.

H. Taira and M. Haruno. 1999. Feature Selection in SVM Categorization. *Proc. of AAAI'99*, pages 480–486.

V. N. Vapnik. 1995. *The Nature of Statistical Learnig Theory*. Springer.