Learning Unsupervised SVM Classifier for Answer Selection in Web Question Answering

Youzheng Wu, Ruiqiang Zhang, Xinhui Hu, and Hideki Kashioka
National Institute of Information and Communications Technology (NICT),
ATR Spoken Language Communication Research Labs.
2-2-2 Hikaridai "Keihanna Science City" Kyoto 619-0288 Japan
{Youzheng.wu,Ruiqiang.zhang,Xinhui.hu,Hideki.kashioka}@atr.jp

Abstract

Previous machine learning techniques for answer selection in question answering (QA) have required question-answer training pairs. It has been too expensive and labor-intensive, however, to collect these training pairs. This paper presents a novel unsupervised support vector machine (U-SVM) classifier for answer selection, which is independent of language and does not require hand-tagged training pairs. The key ideas are the following: 1. unsupervised learning of training data for the classifier by clustering web search results; and 2. selecting the correct answer from the candidates by classifying the question. The comparative experiments demonstrate that the proposed approach significantly outperforms the retrieval-based model (Retrieval-M), the supervised SVM classifier (S-SVM), and the pattern-based model (Pattern-M) for answer selection. Moreover, the cross-model comparison showed that the performance ranking of these models was: U-SVM > Pattern-M > S-SVM > Retrieval-M.

1 Introduction

The purpose of answer selection in QA is to select the exact answer to the question from the extracted candidate answers. In recent years, many supervised machine learning techniques for answer selection in open-domain question answering have been investigated in some pioneering studies [Ittycheriah et al. 2001; Ng et al. 2001; Suzuki et al. 2002; Sasaki, et al. 2005; and Echihabi et al. 2003]. Compared with retrieval-based [Yang et al. 2003], pattern-based [Ravichandran et al. 2002 and Soubbotin et al. 2002], and deep NLP-based [Moldovan et al. 2002, Hovy et al. 2001; and Pasca et al. 2001] answer selection, machine learning techniques are more effective in constructing QA components from scratch. These techniques suffer, however, from the problem of requiring an adequate number of hand-tagged question-answer training pairs. It is too expensive and labor intensive to collect such training pairs for supervised machine learning techniques.

To tackle this knowledge acquisition bottleneck, this paper presents an unsupervised SVM classifier for answer selection, which is independent of language and question type, and avoids the need for hand-tagged question-answer pairs. The key ideas are as follows:

1. Regarding answer selection as a kind of classification task and adopting an SVM classifier;

2. Applying unsupervised learning of pseudo-training data for the SVM classifier by clustering web search results;

3. Training the SVM classifier by using three types of features extracted from the pseudo-training data; and

4. Selecting the correct answer from the candidate answers by classifying the question. Note that this means classifying a question into one of the clusters learned by clustering web search results. Therefore, our classifying the question

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 33–41, Prague, June 2007.

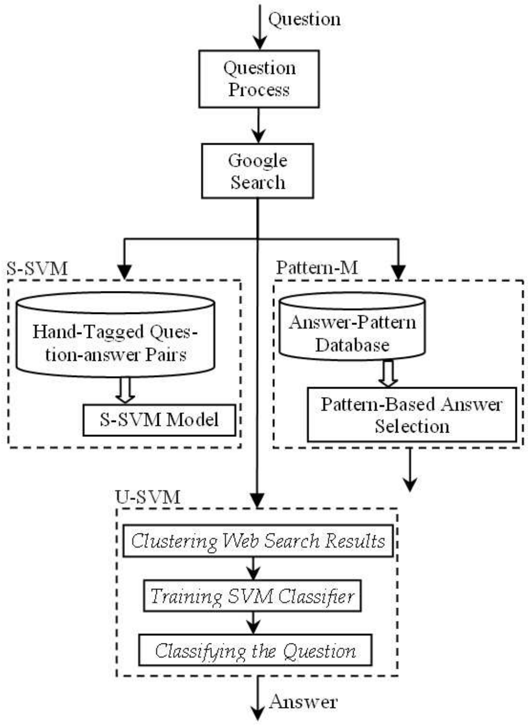

Figure 1: Web Question Answering Architecture

is different from conventional question classification (QC) [Li et al. 2002] that determines the answer type of the question.

The proposed approach is fully unsupervised and starts only from a user question. It does not require richly annotated corpora or any deep linguistic tools. To the best of our knowledge, no research on this kind of study we discuss here has been reported. Figure 1 illustrates the architecture of our web QA approach. The S-SVM and Pattern-M models are included for comparison.

Because the focus of this paper just evaluates the answer selection part, our approach requires knowledge of the answer type to the question in order to find candidate answers, and that the answer must be a NE for convenience in candidate extraction. Experiments using Chinese versions of the TREC 2004 and 2005 test data sets show that our approach significantly outperforms the S-SVM for answer selection, with a top_1 score improvement of more than 20%. Results obtained with the test data set in [Wu et al. 2004] show that the U-SVM increases the $top\_1/mrr\_5/top\_5$ scores by 5.95%/6.06%/8.68% as compared with the Pattern-M. Moreover, our cross-model comparison demonstrates that the performance ranking of all models considered is: U-SVM > Pattern-M > S-SVM > Retrieval-M.

2 Comparison among Models

Related researches on answer selection in QA can be classified into four categories. The retrieval-based model [Yang et al. 2003] selects a correct answer from the candidates according to the distance between a candidate and all question keywords. This model does not work, however, if the question and the answer-bearing sentences do not match on the surface. The pattern-based model [Ravichandran et al. 2002 and Soubbotin et al. 2002] first classifies the question into predefined categories, and then extracts the exact answer by using answer patterns learned off-line. Although 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. Furthermore, this model is not suitable for all types of questions. The deep NLP-based model [Moldovan et al. 2002; Hovy et al. 2001; and Pasca et al. 2001] usually parses the user question and an answer-bearing sentence into a semantic representation, and then semantically matches them to find the answer. This model has performed very well at TREC workshops, but it heavily depends on high-performance NLP tools, which are time consuming and labor intensive for many languages. Finally, the machine learning-based model has also been investigated. current models of this type are based on supervised approaches [Ittycheriah et al. 2001; Ng et al. 2001; Suzuki et al. 2002; and Sasaki et al. 2005] that are heavily dependent on hand-tagged question-answer training pairs, which not readily available.

In response to this situation, this paper presents the U-SVM for answer selection in open-domain web question answering system. Our U-SVM has the following advantages over supervised machine learning techniques. First, the U-SVM classifies questions into a question-dependent set of clusters, and the answer is the name of a question cluster. In contrast, most previous models have classified candidates into positive and negative. Second, the U-SVM automatically learns the unique question-dependent clusters and the pseudo-training for each

Table 1: Comparison of Various Machine Learning Techniques

System	Model	Key Idea	Training Data
[Ittycheriah et al. 2001]	ME Classifier	Classifying candidates into positive and negative	5,000 English Q-A pairs
[Suzuki et al. 2002]	SVM Classifier	Classifying candidates into positive and negative	1358 Japanese Q-A pairs
[Echihabi et al. 2003]	N-C Model	Selecting correct answer by aligning question with sentences	90,000 English Q-A pairs
[Sasaki et al. 2005]	ME Classifier	Classifying words in sentences into answer and non-answer words	2,000 Japanese Q-A pairs
Our U-SVM Model	SVM Classifier	Classifying question into a set of question-dependent clusters	No Q-A pairs

question. This differs from the supervised techniques, in which a large number of hand-tagged training pairs are shared by all of the test questions. In addition, supervised techniques independently process the answer-bearing sentences, so the answers to the questions may not always be extractable because of algorithmic limitations. On the other hand, the U-SVM can use the interdependence between answer-bearing sentences to select the answer to a question.

Table 1 compares the key idea and training data used in the U-SVM with those used in the supervised machine learning techniques. Here, ME means the maximum entropy model, and N-C means the noisy-channel model.

3 The U-SVM

The essence of the U-SVM is to regard answer selection as a kind of text categorization-like classification task, but with no training data available. In the U-SVM, the steps of "clustering web search results", "classifying the question", and "training SVM classifier" play very important roles.

3.1 Clustering Web Search Results

Web search results, such as snippets returned by Google, usually include a mixture of multiple subtopics (called clusters in this paper) related to the user question. To group the web search results into clusters, we assume that the candidate answer in each Google snippet can represent the "signature" of its cluster. In other words, the Google snippets containing the same candidate are regarded as aligned snippets, and thus belong to the same cluster. Web search results are clustered in two phases.

- A first-stage Google search (FGS) is applied to extract n candidate answers $\{c_1, c_2, \ldots, c_n\}$ from the top m Google snippets $\{s_1, s_2, \ldots, s_m\}$ by a NER tool [Wu et al. 2005]. Those snippets containing the candidates $\{c_i\}$ and at least one question keyword $\{q_i\}$ are retained. Finally, the retained snippets $\{s_1, s_2, \ldots, s_m\}$ are clustered into n clusters $\{C_1, C_2, \ldots, C_n\}$ by clustering web search results, that is,

  If a snippet includes L different candidates, the snippet belongs to L different clusters.
  If the candidates of different snippets are the same, these snippets belong to the same clusters.

  Consequently, the number of clusters $\{C_i\}$ is fully determined by the number of candidates $\{c_i\}$, and the cluster name of a cluster C_i is the candidate answer c_i. Up to this point, we have obtained clusters and sample snippets for each cluster that will be used as training data for the SVM classifier. Because this training data is learned automatically, rather than hand-tagged, we call it pseudo-training data.

- A second-stage Google search (SGS) is applied to resolve data sparseness in the pseudo-training samples learned through the FGS. The FGS data may have very few training snippets in some clusters, so more snippets must be collected. Note that this step just learns new

Google snippets into the clusters learned by the FGS, but does not add new clusters.

For each candidate answer c_i:
Combine the original query $q = \{q_i\}$ and the candidate c_i to form a new query $q\prime = \{q, c_i\}$.
Submit $q\prime$ to Google and download the top 50 Google snippets.
Retain the snippets containing the candidate c_i and at least one keyword q_i.
Group the retained snippets into n clusters to form the new pseudo-training data.
End

Here, we give an example illustrating the principle of clustering web search results in the FGS. In submitting TREC 2004 test question 1.1 ”when was the first Crip gang started?” to Google (*http://www.google.com/apis*), we extract $n(= 8)$ different candidates from the top $m(= 30)$ Google snippets. The Google snippets containing the same candidates are aligned snippets, and thus the 12 retained snippets are grouped into 8 clusters, as listed in Table 2. This table roughly indicates that the snippets with the same candidate answers contain the same sub-meanings, so these snippets are considered as aligned snippets. For example, all Google snippets that contain the candidate answer *1969* express the time of establishment of ”the first Crip gang”.

In summary, the U-SVM uses the result of ”clustering web search results” as the pseudo-training data of the SVM classifier, and then classifies user question into one of the clusters for answer selection. On the one hand, the clusters and their names are based on candidate answers to question; on the other hand, candidates are dependent on question. Therefore, the clusters are question-dependent.

3.2 Classifying Question

Using the pseudo-training data obtained by clustering web search results to train the SVM classifier, we classify user questions into a set of question-dependent clusters and assume that the correct answer is the name of the question cluster that is assigned by the trained U-SVM classifier. For the above example, if the U-SVM classifier, trained on the pseudo-training data listed in Table 2, classifies the above test question into a cluster whose name is *1969*, then the cluster name *1969* is the answer to the question.

This paper selects LIBSVM toolkit[1] to implement the SVM classifier. The kernel is the radical basis function with the parameter $\gamma = 0.001$ in the experiments.

3.3 Feature Extraction

To classify the question into a question-dependent set of clusters, the U-SVM classifier extracts three types of features.

- A similarity-based feature set (SBFS) is extracted from the Google snippets. The SBFS attempts to capture the word overlap between a question and a snippet. The possible values range from 0 to 1.

SBFS Features
percentage of matched keywords (KWs)
percentage of mismatched KWs
percentage of matched bi-grams of KWs
percentage of matched thesauruses
normalized distance between candidate and KWs

  To compute the matched thesaurus feature, we adopt TONGYICICILIN [2] in the experiments.

- A Boolean match-based feature set (BMFS) is also extracted from the Google snippets. The BMFS attempts to capture the specific keyword Boolean matches between a question and a snippet. The possible values are true or false.

BMFS Features
person names are matched or not
location names are matched or not
organization names are matched or not
time words are matched or not
number words are matched or not
root verb is matched or not
candidate has or does not have bi-gram in snippet matching bi-gram in question
candidate has or does not have desired named entity type

- A window-based word feature set (WWFS) is a set of words consisting of the words

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[2] A Chinese Thesaurus Lexicon

Table 2: Clustering Web Search Results

Cluster Name	Google Snippet
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 ...

preceding $\{w_{i-5}, \ldots, w_{i-1}\}$ and following $\{w_{i+1}, \ldots, w_{i+5}\}$ the candidate answer. The WWFS features can be regarded as a kind of relevant snippets-based question keywords expansion. By extracting the WWFS feature set, the feature space in the U-SVM becomes question dependent, which may be more suitable for classifying the question. The number of classification features in the S-SVM must be fixed, however, because all questions share the same training data. This is one difference between the U-SVM and the supervised SVM classifier for answer selection. Each word feature in the WWFS is weighted using its ISF value.

$$ISF(w_j, C_i) = \frac{N(w_j, C_i) + 0.5}{N(w_j) + 0.5} \quad (1)$$

where $N(w_j)$ is the total number of the snippets containing word feature w_j, and $N(w_j, C_i)$ is the number of snippets in cluster C_i containing word feature w_j.

When constructing question vector, we assume that the question is an ideal question that contains all the extracted WWFS words. Therefore, the values of the WWFS word features in question vector are 1. Similarly, the values of the SBFS and BMFS features in question vector are also estimated by self-similarity calculation.

4 Experiments

4.1 Data Sets

For the experiments, no English named entity recognition (NER) tool is in our hand at the time of the experiments; therefore, we validate the U-SVM in terms of Chinese web QA using three test data sets, which will be published with this paper[3]. Although the U-SVM is independent of the question types, for convenience in candidate extraction, only those questions whose answers are named entities are selected. The three test data sets are CTREC04, CTREC05 and CTEST05. 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. CTEST05 is a set of 178 Chinese questions found in [Wu et al. 2004] that are similar to TREC testing questions except that they are written in Chinese. Figure 2 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.

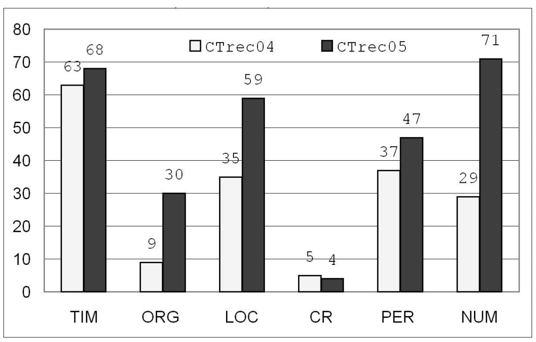

Figure 2: Statistics of CTEST05

To collect the question-answer 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 pairs as the training data of S-SVM called CTRAINDATA.

4.2 Evaluation Method

In the experiments, the top $m(=50)$ Google snippets are adopted to extract candidates by using a Chinese NER tool [Wu et al. 2005]. The number of the candidates extracted from the top $m(=50)$ snippets, n, is adaptive for different questions but it does not exceed 30. The results are evaluated in terms of two scores, top_n and mrr_5. Here, top_n is the rate at which at least one correct answer is included in the top n answers, while mrr_5 is the average reciprocal rank $(1/n)$ of the highest rank $n(n \leq 5)$ of a correct answer to each question.

[3] Currently no public testing question set for simplified Chinese QA is available.

4.3 U-SVM vs. Retrieval-M

The Retrieval-M selects the candidate with the shortest distances to all question keywords as the correct answer. In this experiment, the Retrieval-M is implemented based on the snippets returned by Google, while the U-SVM is based on the SGS data, the SBFS and BMFS feature. Table 3 summarizes the comparative performance.

Table 3: Comparison of Retrieval-M and U-SVM

		Retrieval-M	U-SVM
CTREC04	top_1	27.84%	53.61%
	mrr_5	43.67%	66.25%
	top_5	71.13%	88.66%
CTREC05	top_1	34.00%	50.00%
	mrr_5	48.20%	62.38%
	top_5	71.33%	82.67%

The table shows that the U-SVM greatly improves the performance of the Retrieval-M: the top_1 improvements for CTREC04 and CTREC05 are about 25.8% and 16.0%, respectively. This experiment demonstrates that the assumptions used here in clustering web search results and in classifying the question are effective in many cases, and that the U-SVM benefits from these assumptions.

4.4 U-SVM vs. S-SVM

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-SVM with the SBFS and BMFS feature sets. The U-SVM results are compared with the S-SVM results for CTREC04 and CTREC05 in Tables 4 and 5, respectively. The S-SVM is trained on CTRAINDATA.

These tables show the following:

Table 4: Comparison of U-SVM and S-SVM on CTREC04

		FGS	SGS
top_1	S-SVM	30.93%	39.18%
	U-SVM	45.36%	53.61%
mrr_1	S-SVM	45.36%	53.54%
	U-SVM	57.44%	66.25%
top_5	S-SVM	71.13%	79.38%
	U-SVM	76.29%	88.66%

Table 5: Comparison of U-SVM and S-SVM on CTREC05

		FGS	SGS
top_1	S-SVM	30.00%	33.33%
	U-SVM	48.00%	50.00%
mrr_1	S-SVM	45.59%	48.67%
	U-SVM	58.01%	62.38%
top_5	S-SVM	72.00%	74.67%
	U-SVM	75.33%	82.67%

- The proposed U-SVM significantly outperforms the S-SVM for all measurements and all test data sets. For the CTREC04 test data set, the top_1 improvements for the FGS and SGS data are about 14.5% and 14.4%, respectively. For the CTREC05 test data set, the top_1 score for the FGS data increases from 30.0% to 48.0%, and the top_1 score for the SGS data increases from 33.3% to 50.0%. Note that the SBFS and BMFS features here is fewer than the features in [Ittycheriah et al. 2001; Suzuki et al. 2002], but the comparison is still effective because the models are compared in terms of the same features. In the S-SVM, all questions share the same training data, while the U-SVM uses the unique pseudo-training data for each question. This is the main reason why the U-SVM performs better than the S-SVM does.

- The SGS data is greatly helpful for both the U-SVM and the S-SVM. Compared with the FGS data, the $top\_1/mrr\_5/top\_5$ improvements for the S-SVM and the U-SVM on CTREC04 are 8.25%/8.18%/8.25% and 7.25%/8.81%/12.37%. The SGS can be regarded as a kind of query expansion. The reasons for this improvement are: the data sparseness in FGS data is partially resolved; and the use of the Web to introduce data redundancy is helpful. [Clarke et al. 2001; Magnini et al. 2002; and Dumais et al. 2002].

In the S-SVM, all of the test questions share the same hand-tagged training data, so the WWFS features cannot be easily used [Ittycheriah et al. 2002; Suzuki, et al. 2002]. Tables 6 and 7 compare the performances of the U-SVM with the (SBFS + BMFS) features, the WWFS features, and combination of three types of features for the CTREC04 and CTREC05 test data sets, respectively.

Table 6: Performances of U-SVM for Different Features on CTREC04

	SBFS+BMFS	WWFS	Combination
top_1	53.61%	46.39%	60.82%
mrr_5	66.25%	59.19%	71.31%
top_5	88.66%	81.44%	88.66%

Table 7: Performances of U-SVM for Different Features on CTREC05

	SBFS+BMFS	WWFS	Combination
top_1	50.00%	49.33%	57.33%
mrr_5	62.38%	59.26%	65.61%
top_5	82.67%	74.00%	80.00%

These tables report that combining three types of features can improve the performance of the U-SVM. Using a combination of features with the CTREC04 test data set results in the best performances: 60.82%/71.31%/88.66% for $top\_1/mrr\_5/top\_5$. Similarly, as compared with using the (SBFS + BMFS) and WWFS features, the improvements from using a combination of features with the CTREC05 test data set are 7.33%/3.23%/-2.67% and 8.00%/6.35%/6.00%, respectively. The results also demonstrate that the (SBFS + BMFS) features are more important than the WWFS features.

These comparative experiments indicate that the U-SVM performs better than the S-SVM does, even though the U-SVM is an unsupervised technique and no hand-tagged training data is provided. The average top_1 improvements for both test data sets are both more than 20%.

4.5 U-SVM vs. Pattern-M vs. S-SVM

To compare the U-SVM with the Pattern-M and the S-SVM, we use the CTEST05 data set, shown in Figure 3. The CTEST05 includes 14 different question types, for example, Inventor_Stuff (with question like "Who invented telephone?"), Event-Day (with question like "when is World Day for Water?"), and so on. The Pattern-M uses the dependency syntactic answer patterns learned in [Wu et al. 2007] to extract the answer, and named entities are also used to filter noise from the candidates.

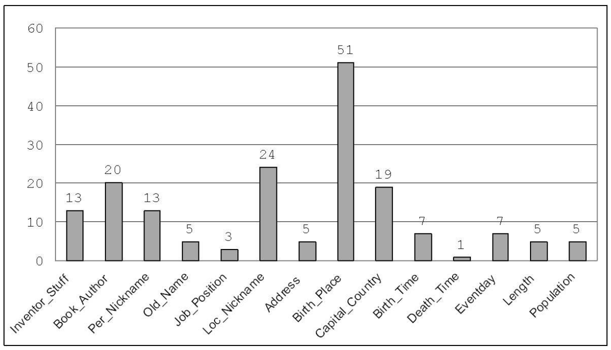

Figure 3: Statistics of CTEST05

Table 8 summarizes the performances of the U-SVM, Pattern-M, and S-SVM models on CTEST05.

Table 8: Comparison of U-SVM, Pattern-M and S-SVM on CTEST05

	S-SVM	Pattern-M	U-SVM
top_1	44.89%	53.14%	59.09%
mrr_5	56.49%	61.28%	67.34%
top_5	74.43%	73.14%	81.82%

The results in the table show that the U-SVM significantly outperforms the S-SVM and Pattern-M, while the S-SVM underperforms the Pattern-M. Compared with the Pattern-M, the U-SVM 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.
- The imperfection of Google snippets affects pattern matching, and thus adversely influences the Pattern-M model. From the cross-model comparison, we conclude that the performance ranking of these models is: U-SVM > Pattern-M > S-SVM > Retrieval-M.

5 Conclusion and Future Work

This paper presents an unsupervised machine learning technique (called the U-SVM) for answer selection that is validated in Chinese open-domain web QA. Regarding answer selection as a kind of classification task, the U-SVM automatically learns clusters and pseudo-training data for each cluster by clustering web search results. It then selects the correct answer from the candidates according to classifying the question. The contribution of this paper is that it presents an unsupervised machine learning technique for web QA that starts with only a user question. The results of our experiments with three test data sets are encouraging. As compared with the S-SVM, the top_1 performances of the U-SVM for the CTREC04 and CTREC05 data sets are significantly improved, at more than 20%. Moreover, the U-SVM performs better than the Retrieval-M and the Pattern-M.

These experiments have only validated the U-SVM on named entity types of questions that account for about 82% of all TREC2004 and 2005 FACTOID test questions. In fact, our technique is independent of question types only if the candidates can be extracted. In the future, we will explore the effectiveness of our technique for the other types of questions. The web search results clustering in the U-SVM defines that a candidate in a Google snippet can represent the "signature" of its cluster. This definition, however, is not always effective. To filter noise in the pseudo-training data, we will extract relations between the candidates and the keywords as the cluster signatures of Google snippets. Moreover, applying the U-SVM to QA systems in other languages, like English and Japanese, will also be included in our future work.

References

Abdessamad Echihabi, and Daniel Marcu. 2003. *A Noisy-Channel Approach to Question Answering*. In Proc. of ACL-2003, Japan.

Abraham Ittycheriah, Salim Roukos. 2002. *IBM's Statistical Question Answering System-TREC 11*. In Proc. of TREC-11, Gaithersburg, Maryland.

Bernardo Magnini, Matteo Negri, Roberto Prevete, Hristo Tanev. 2002. *Is It the Right Answer? Exploiting Web Redundancy for Answer Validation*. In Proc. of ACL-2002, Philadelphia, pp. 425 432.

Charles L. A. Clarke, Gordon V. Cormack, Thomas R. Lynam. *Exploiting Redundancy in Question Answering* In Proc. of SIGIR-2001, pp 358–365, 2001.

Christopher Pinchak, Dekang Lin. 2006. *A Probabilistic Answer Type Model*. In Proc. of EACL-2006, Trento, Italy, pp. 393-400.

Dan Moldovan, Sanda Harabagiu, Roxana Girju, et al. 2002. *LCC Tools for Question Answering*. NIST Special Publication: SP 500-251, TREC-2002.

Deepak Ravichandran, Eduard Hovy. 2002. *Learning Surface Text Patterns for a Question Answering System*. In Proc. of the 40th ACL, Philadelphia, July 2002.

Eduard Hovy, Ulf Hermjakob, Chin-Yew Lin. 2001. *The Use of External Knowledge of Factoid QA*. In Proc. of TREC 2001, Gaithersburg, MD, U.S.A., November 13-16, 2001.

Hui Yang, Tat-Seng Chua. 2003. *QUALIFIER: Question Answering by Lexical Fabric and External Resources*. In Proc. of EACL-2003, page 363-370.

Hwee T. Ng, Jennifer L. P. Kwan, and Yiyuan Xia. 2001. *Question Answering Using a Large Text Database: A Machine Learning Approach*. In Proc. of EMNLP-2001, pp66-73 (2001).

Jun Suzuki, Yutaka Sasaki, Eisaku Maeda. 2002. *SVM Answer Selection for Open-Domain Question Answering*. In Proc. of Coling-2002, pp. 974 980 (2002).

Marius Pasca. 2001. *A Relational and Logic Representation for Open-Domain Textual Question Answering*. In Proc. of ACL (Companion Volume) 2001: 37-42.

Martin M. Soubbotin, Sergei M. Soubbotin. 2002. *Use of Patterns for Detection of Likely Answer Strings: A Systematic Approach*. In Proc. of TREC-2002, Gaithersburg, Maryland, November 2002.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andre Ng. *Web Question Answering: Is More Always Better?*. In Proc. SIGIR-2002, pp 291–298, 2002.

Xin Li, and Dan Roth. 2002. *Learning Question Classification*. In Proc. of the 19th International Conference on Computational Linguistics, Taibai, 2002.

Youzheng Wu, Hideki Kashioka, Jun Zhao. 2007. *Using Clustering Approaches to Open-domain Question Answering*. In Proc. of CICLING-2007, Mexico City, Mexico, pp506 517, 2007.

Youzheng Wu, Jun Zhao and Bo Xu. 2005. *Chinese Named Entity Recognition Model Based on Multiple Features*. In Proc. of HLT/EMNLP-2005, Vancouver, Canada, pp.427-434.

Youzheng Wu, Jun Zhao, Xiangyu Duan and Bo Xu. 2004. *Building an Evaluation Platform for Chinese Question Answering Systems*. In Proc. of the First NCIRCS, China, December, 2004.

Yutaka Sasaki. 2005. *Question Answering as Question-Biased Term Extraction: A New Approach toward Multilingual QA*. In Proc. of ACL-2005, pp.215-222.