Question Classification using HDAG Kernel

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

Abstract

This paper proposes a machine learning based question classification method using a kernel function, *Hierarchical Directed Acyclic Graph (HDAG) Kernel*. The HDAG Kernel directly accepts structured natural language data, such as several levels of chunks and their relations, and computes the value of the kernel function at a practical cost and time while reflecting all of these structures. We examine the proposed method in a question classification experiment using 5011 Japanese questions that are labeled by 150 question types. The results demonstrate that our proposed method improves the performance of question classification over that by conventional methods such as *bag-of-words* and their combinations.

1 Introduction

Open-domain Question Answering (ODQA) involves the extraction of correct answer(s) to a given free-form factual question from a large collection of texts. ODQA has been actively studied all over the world since the start of the Question Answering Track at TREC-8 in 1999.

The definition of ODQA tasks at the TREC QA-Track has been revised and extended year after year. At first, ODQA followed the *Passage Retrieval* method as used at TREC-8. That is, the ODQA task was to answer a question in the form of strings of 50 bytes or 250 bytes excerpted from a large set of news wires. Recently, however, the ODQA task is considered to be a task of extracting *exact answers* to a question. For instance, if a QA system is given the question “When was Queen Victoria born?”, it should answer “1832”.

Typically, QA systems have the following components for achieving ODQA:

Question analysis analyzes a given question and determines the question type and keywords.

Text retrieval finds the top N paragraphs or documents that match the result of the question analysis component.

Answer candidate extraction extracts answer candidates of the given question from the documents retrieved by the text retrieval component, based on the results of the question types.

Answer selection selects the most plausible answer(s) to the given question from among the answer candidates extracted by the answer candidate extraction component.

One of the most important processes of those listed above is identifying the target of intention in a given question to determine the type of sought-after answer. This process of determining the question type for a given question is usually called *question classification*. Without a question type, that is, the result of question classification, it would be much more difficult or even nearly infeasible to select correct answers from among the possible answer candidates, which would necessarily be all of the noun phrases or named entities in the texts. Question classification provides the benefit of a powerful restriction that reduces to a practical number of the answer candidates that should be evaluated in the answer selection process.

This work develops a machine learning approach to question classification (Harabagiu et al., 2000; Hermjakob, 2001; Li and Roth, 2002). We use the *Hierarchical Directed Acyclic Graph (HDAG) Kernel* (Suzuki et al., 2003), which is suited to handle structured natural language data. It can handle structures within texts as the features of texts without converting the structures to the explicit representation of numerical feature vectors. This framework is useful for question classification because the works of (Li and Roth, 2002; Suzuki et al., 2002a) showed that richer information, such as structural and semantical information inside a given question, improves the question classification performance over using the information of just simple key terms.

In Section 2, we present the question classification problem. In Section 3, we explain our proposed method for question classification. Finally, in Section 4, we describe our experiment and results.

2 Question Classification

Question classification is defined as a task that maps a given question to more than one of k question types (classes).

In the general concept of QA systems, the result of question classification is used in a downstream process, answer selection, to select a correct answer from among the large number of answer candidates that are extracted from the source documents. The result of the question classification, that is, the labels of the question types, can reduce the number of answer candidates. Therefore, we no longer have to evaluate every noun phrase in the source documents to see whether it provides a correct answer to a given question. Evaluating only answer candidates that match the results of question classification is an efficient method of obtaining correct answers. Thus, question classification is an important process of a QA system. Better performance in question classification will lead to better total performance of the QA system.

2.1 Question Types: Classes of Questions

Numerous question taxonomies have been defined, but unfortunately, no standard exists.

In the case of the TREC QA-Track, most systems have their own question taxonomy, and these are reconstructed year by year. For example, (Ittycheriah et al., 2001) defined 31 original question types in two levels of hierarchical structure. (Harabagiu et al., 2000) also defined a large hierarchical question taxonomy, and (Hovy et al., 2001) defined 141 question types of a hierarchical question taxonomy.

Within all of these taxonomies, question types are defined from the viewpoint of the target intention of the given questions, and they have hierarchical structures, even though these question taxonomies are defined by different researchers. This because the purpose of question classification is to reduce the large number of answer candidates by restricting the target intention via question types. Moreover, it is very useful to handle question taxonomy constructed in a hierarchical structure in the downstream processes. Thus, question types should be the target intention and constructed in a hierarchical structure.

2.2 Properties

Question classification is quite similar to *Text Categorization*, which is one of the major tasks in *Natural Language Processing (NLP)*. These tasks require classification of the given text to certain defined classes. In general, in the case of text categorization, the given text is one document, such as a newspaper article, and the classes are the topics of the articles. In the case of question classification, a given text is one short question sentence, and the classes are the target answers corresponding to the intention of the given question.

However, question classification requires much more complicated features than text categorization, as shown by (Li and Roth, 2002). They proved that question classification needs richer information than simple key terms (bag-of-words), which usually give us high performance in text classification. Moreover, the previous work of (Suzuki et al., 2002a) showed that the sequential patterns constructed by different levels of attributes, such as words, part-of-speech (POS) and semantical information, improve the performance of question classification. The experiments in these previous works indicated that the structural and semantical features inside questions have the potential to improve the performance of question classification. In other words, high-performance question classification requires us to extract the structural and semantical features from the given question.

2.3 Learning and Classification Task

This paper focuses on the machine learning approach to question classification. The machine learning approach has several advantages over manual methods.

First, the construction of a manual classifier for questions is a tedious task that requires the analysis of a large number of questions. Moreover, mapping questions into question types requires the use of lexical items and, therefore, an explicit representation of the mapping may be very large. On the other hand, machine learning approaches only need to define features. Finally, the classifier can be more flexibly reconstructed than a manual one because it can be trained on a new taxonomy in a very short time.

As the machine learning algorithm, we chose the *Support Vector Machines (SVMs)* (Cortes and Vapnik, 1995) because the work of (Joachims, 1998; Taira and Haruno, 1999) reported state-of-the-art performance in text categorization as long as question classification is a similar process to text categorization.

3 HDAG Kernel

Recently, the design of kernel functions has become a hot topic in the research field of machine learning. A specific kernel can drastically increase the performance of specific tasks. Moreover, a specific kernel can handle new feature spaces that are difficult to manage directly with conventional methods.

The HDAG Kernel is a new kernel function that is designed to easily handle structured natural language data. According to the discussion in the previous section, richer information such as structural and semantical information is required for high-performance question classification.

We think that the HDAG Kernel is suitable for improving the performance of question classification: The HDAG Kernel can handle various linguistic structures within texts, such as chunks and their relations, as the features of the text without converting such structures to numerical feature vectors explicitly.

3.1 Feature Space

Figure 1 shows examples of the structures within questions that are handled by the HDAG kernel.

As shown in Figure 1, the HDAG kernel accepts several levels of chunks and their relations inside the text. The nodes represent several levels of chunks including words, and directed links represent their relations. Suppose $\{p_i|p_i \in P\}$ and $\{q_i|q_i \in Q\}$ represent each node. Some nodes have a graph inside themselves, which are called "non-terminal nodes". Each node can have more than one attribute, such as words, part-of-speech tags, semantic information like WordNet (Fellbaum, 1998), and class names of the named entity. Moreover, nodes are allowed to not have any attribute, in other words, we do not have to assign attributes to all nodes.

The "attribute sequence" is a sequence of attributes extracted from the node in sub-paths of HDAGs. One type of attribute sequence becomes one element in the feature vector. The framework of the HDAG Kernel allows node skips during the extraction of attribute sequences, and its cost is based the decay factor $\lambda (0 < \lambda \leq 1)$, since HDAG Kernel deals with not only the exact matching of the sub-structures between HDAGs but also the approximate structure matching of them.

Explicit representation of feature vectors in the HDAG kernel can be written as $\phi(G) = (\phi_1(G), \ldots, \phi_N(G))$, where ϕ represents the explicit feature mapping from the HDAG to the feature vector and N represents the number of all possible types of attribute sequences extracted to the HDAGs. The value of $\phi_i(G)$ is the number of occurrences of the i'th attribute sequence in the HDAG G, weighted according to the node skip.

Table 1 shows a example of attribute sequences that are extracted from the example question in Figure 1. The symbol $*$ in the sub-path column shows that more than one node skip occurred there. The parentheses "( )" in the attribute sequence column represents the boundaries of a node. For example, attribute sequence "purchased-(NNP-Bush)" is ex-

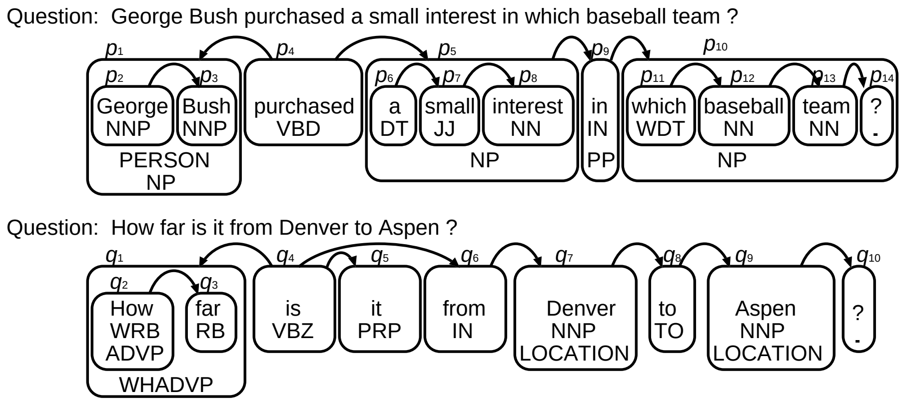

Figure 1: Example of text structure handled by HDAG (From the questions in TREC-10)

Table 1: Examples of attribute sequences, elements of feature vectors, extracted from the example question in Figure 1

sub-path	attribute sequence: element	value
p_1	PERSON	1
p_2	George	1
p_2	NNP	1
p_2-p_3	George-Bush	1
p_2-p_3	NNP-Bush	1
p_2-p_3	George-NNP	1
p_2-p_3	NNP-NNP	1
$\vdots$	$\vdots$	$\vdots$
p_4-p_1	purchased-(NNP-Bush)	1
p_4-p_1	purchased-(PERSON)	1
p_4-p_1	purchased-(Bush)	λ
$\vdots$	$\vdots$	$\vdots$
p_4-*-p_{10}	purchased-(NP)	λ^4
$\vdots$	$\vdots$	$\vdots$
p_4-p_5-*-p_{10}	VBD-(a-small-interest)-(which-baseball-team)	λ^2
p_4-p_5-*-p_{10}	purchased-(NP)-(which-team)	λ^3

tracted from sub-path "p_4-p_1", and "NNP-Bush" is in the node p_1.

The return value of the HDAG Kernel can be defined as:

$$K(x,y) = \langle \phi(x) \cdot \phi(y) \rangle = \sum_{i=1}^{N} \phi_i(x) \cdot \phi_i(y), \qquad (1)$$

where input objects x and y are the objects represented in HDAG G_1 and G_2, respectively. According to this formula, the HDAG Kernel calculates the inner product of the common attribute sequences weighted according to their node skips and the occurrence between the two HDAGs, G_1 and G_2.

3.2 Efficient Calculation Method

In general, the dimension of the feature space N in equation (1) becomes very high or even infinity. It might thus be computationally infeasible to generate feature vector $\phi(G)$ explicitly.

To solve this problem, we focus on the framework of the kernel functions defined for a discrete structure, Convolution Kernels (Haussler, 1999). One of the most remarkable properties of this kernel methodology is that it can calculate kernel functions by the "inner products between pairs of objects" while it retains the original representation of objects. This means that we do not have to map input objects to the numerical feature vectors by explicitly representing them, as long as an efficient calculation for the inner products between a pair of texts is defined.

However, Convolution Kernels are abstract concepts. The *Tree Kernel* (Collins and Duffy, 2001) and *String Subsequence Kernel (SSK)* (Lodhi et al., 2002) are examples of instances in the Convolution Kernels developed in the NLP field.

The HDAG Kernel also use this framework: we can learn and classify without creating explicit numerical feature vectors like equation (1). The efficient calculation of inner products between HDAGs, the return value of HDAG Kernel, was defined in a recursive formulation (Suzuki et al., 2003). This recursive formulation for HDAG Kernel can be rewritten as "for loops" by using the dynamic programming technique. Finally, the HDAG Kernel can be calculated in $O(|P||Q|)$ time.

Table 2: Distribution of 5011 questions over question type hierarchy

d	question type	#	d	question type	#	d	question type	#
0	!TOP	4963	3	LINE	24	3	!POSITION_TITLE	97
1	NAME	3190	4	*RAILROAD	3	2	*LANGUAGE	8
2	PERSON	824	4	!ROAD	11	2	*RELIGION	6
3	*LASTNAME	0	4	*WATERWAY	0	1	NATURAL_OBJECT	96
3	*MALE_FIRSTNAME	1	4	*TUNNEL	1	2	ANIMAL	18
3	*FEMALE_FIRSTNAME	2	4	*BRIDGE	1	2	VEGETABLE	15
2	ORGANIZATION	733	3	*PARK	2	2	MINERAL	54
3	COMPANY	119	3	*MONUMENT	3	1	COLOR	10
3	*COMPANY_GROUP	0	2	PRODUCT	468	1	TIME_TOP	779
3	*MILITARY	4	3	VEHICLE	37	2	TIMEX	652
3	INSTITUTE	26	4	*CAR	8	3	TIME	50
3	*MARKET	0	4	*TRAIN	2	3	DATE	594
3	POLITICAL_ORGANIZATION	103	4	*AIRCRAFT	5	3	*ERA	5
4	GOVERNMENT	38	4	*SPACESHIP	8	2	PERIODX	125
4	POLITICAL_PARTY	43	4	!SHIP	12	3	*TIME_PERIOD	9
4	PUBLIC_INSTITUTION	19	3	DRUG	15	3	*DATE_PERIOD	9
3	GROUP	96	3	*WEAPON	4	3	*WEEK_PERIOD	4
4	!SPORTS_TEAM	20	3	*STOCK	0	3	*MONTH_PERIOD	6
3	*ETHNIC_GROUP	4	3	*CURRENCY	8	3	!YEAR_PERIOD	41
3	*NATIONALITY	4	3	AWARD	11	1	NUMEX	882
2	LOCATION	752	3	*THEORY	1	2	MONEY	187
3	GPE	265	3	RULE	66	2	*STOCK_INDEX	0
4	CITY	77	3	*SERVICE	2	2	*POINT	9
4	*COUNTY	1	3	*CHARCTER	4	2	PERCENT	94
4	PROVINCE	47	3	METHOD_SYSTEM	33	2	MULTIPLICATION	10
4	COUNTRY	116	3	ACTION_MOVEMENT	21	2	FREQUENCY	27
3	REGION	23	3	*PLAN	1	2	*RANK	8
3	GEOLOGICAL_REGION	22	3	*ACADEMIC	5	2	AGE	58
4	*LANDFORM	9	3	*CATEGORY	0	2	MEASUREMENT	133
4	*WATER_FORM	7	3	SPORTS	11	3	PHYSICAL_EXTENT	53
4	*SEA	3	3	OFFENCE	10	3	SPACE	18
3	*ASTRAL_BODY	5	3	ART	78	3	VOLUME	14
4	*STAR	2	4	*PICTURE	2	3	WEIGHT	22
4	*PLANET	2	4	*BROADCAST_PROGRAM	6	3	*SPEED	9
3	ADDRESS	59	4	MOVIE	15	3	*INTENSITY	0
4	POSTAL_ADDRESS	24	4	*SHOW	4	3	*TEMPERATURE	7
4	PHONE_NUMBER	22	4	MUSIC	13	3	*CALORIE	1
4	*EMAIL	4	3	PRINTING	31	3	*SEISMIC_INTENSITY	2
4	*URL	8	4	!BOOK	10	2	COUNTX	326
2	FACILITY	147	4	*NEWSPAPER	7	3	N_PERSON	162
3	GOE	99	4	*MAGAZINE	4	3	N_ORGANIZATION	49
4	SCHOOL	27	2	DISEASE	44	3	N_LOCATION	27
4	*MUSEUM	3	2	EVENT	99	4	*N_COUNTRY	9
4	*AMUSEMENT_PARK	4	3	*GAMES	8	3	*N_FACILITY	6
4	WORSHIP_PLACE	10	3	!CONFERENCE	17	3	N_PRODUCT	47
4	STATION_TOP	12	3	*PHENOMENA	6	3	*N_EVENT	8
5	*AIRPORT	6	3	*WAR	3	3	*N_ANIMAL	7
5	*STATION	3	3	*NATURAL_DISASTER	5	3	*N_VEGETABLE	0
5	*PORT	3	3	*CRIME	6	3	*N_MINERAL	0
5	*CAR_STOP	0	2	TITLE	97	0	*OTHER	48

4 Experiment

4.1 Data Set

We used three different QA data sets together to evaluate the performance of our proposed method. One is the 1011 questions of NTCIR-QAC1 [1], which were gathered from 'dry-run', 'formal-run' and 'additional-run.' The second is the 2000 questions described in (Suzuki et al., 2002b). The last one is the 2000 questions of CRL-QA data[2]. These three QA data sets are written in Japanese.

These data were labeled with the 150 question types that are defined in the CRL-QA data, along with one additional question type, "OTHER". Table 2 shows all of the question types we used in this experiment, where d represents the depth of the hierarchy and # represents the number of questions of each question type, including the number of questions in "child question types".

While considering question classification as a learning and classification problem, we decided not to use question types that do not have enough questions (more than ten questions), indicated by an asterisk (*) in front of the name of the question type, because classifier learning is very difficult with very few data. In addition, after the above operations, if only one question type belongs to one parent question type, we also deleted it, which is indicated by an exclamation mark (!). Ultimately, we evaluated 68 question types.

4.2 Comparison Methods

We compared the HDAG Kernel (HDAG) to a baseline method that is sometimes referred to as the *bag-*

[1] http://www.nlp.cs.ritsumei.ac.jp/qac/

[2] http://www.cs.nyu.edu/~sekine/PROJECT/CRLQA/

of-words kernel, a bag-of-words (BOW) with a polynomial kernel (d1: first degree polynomial kernel, d2: second degree polynomial kernel).

HDAG and BOW differ in how they consider the structures of a given question. BOW only considers attributes independently (d1) or combinatorially (d2) in a given question. On the other hand, HDAG can consider the structures (relations) of the attributes in a given question.

We selected SVM for the learning and classification algorithm. Additionally, we evaluated the performance using SNoW[3] to compare our method to indirectly the SNoW-based question classifier (Li and Roth, 2002). Note that BOW was used as features for SNoW.

Finally, we compared the performances of HDAG-SVM, BOW(d2)-SVM, BOW(d1)-SVM, and BOW-SNoW. The parameters of each comparison method were set as follows: The decay factor λ was 0.5 for HDAG, and the soft-margin C of all SVM was set to 1. For SNoW, we used $\alpha = 1.35, \beta = 0.8$, and $r = 3$. These parameters were selected based on preliminary experiments.

4.3 Decision Model

Since the SVM is a two-class classification method, we have to make a decision model to determine the question type of a given question that is adapted for question classification, which is a multi-class hierarchical classification problem.

Figure 2 shows how we constructed the final decision model for question classification.

First, we made 68 SVM classifiers for each question type, and then we constructed "one-vs-rest models" for each node in the hierarchical question taxonomy. One of the one-vs-rest models was constructed by some of the SVM classifiers, which were the child question types of the focused node. For example, the one-vs-rest model at the node "TOP" was constructed by five SVM classifiers: "NAME", "NATURAL_OBJECT", "COLOR", "TIME_TOP" and "NUMEX". The total number of one-vs-rest models was 17.

Finally, the decision model was constructed by setting one-vs-rest models in the hierarchical question taxonomy to determine the most plausible question type of a given question.

[3] http://l2r.cs.uiuc.edu/~cogcomp/cc-software.html

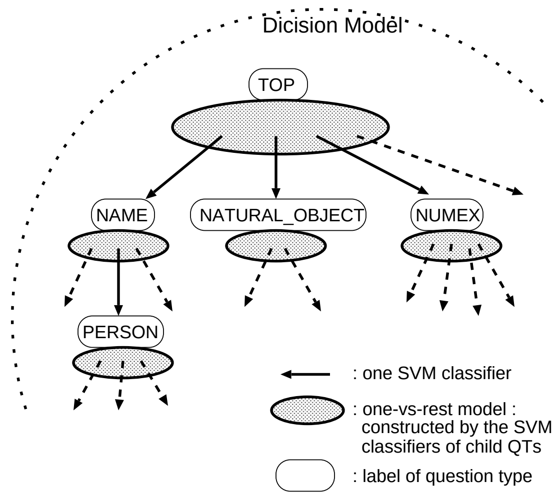

Figure 2: Hierarchical classifier constructed by SVM classifiers

4.4 Features

We set four feature sets for each comparison method.

1. words only (W)
2. words and named entities (W+N)
3. words and semantic information (W+S)
4. words, named entities and semantic information (W+N+S)

The words were analyzed in basic form, and the semantic information was obtained from the "Goitaikei" (Ikehara et al., 1997), which is similar to WordNet in English. Words, chunks and their relations in the texts were analyzed by CaboCha (Kudo and Matsumoto, 2002), and named entities were analyzed by the SVM-based NE tagger (Isozaki and Kazawa, 2002).

Note that even when using the same feature sets, method of how to construct feature spaces are entirely different between HDAG and BOW.

4.5 Evaluation Method

We evaluated the 5011 questions by using five-fold cross-validation and used the following two approaches to evaluate the performance.

Table 3: Results of question classification experiment by five-fold cross-validation

Macc	W	W+N	W+S	W+N+S
HDAG-SVM	0.862	0.871	0.877	**0.882**
BOW(d2)-SVM	0.841	0.847	0.847	0.856
BOW(d1)-SVM	0.839	0.843	0.837	0.851
BOW-SNoW	0.760	0.774	0.800	0.808

Qacc	W	W+N	W+S	W+N+S
HDAG-SVM	0.730	0.736	0.742	**0.749**
BOW(d2)-SVM	0.678	0.691	0.686	0.704
BOW(d1)-SVM	0.679	0.686	0.671	0.694
BOW-SNoW	0.562	0.573	0.614	0.626

1. Average accuracy of each one-vs-rest model (Macc)
   This measure evaluates the performance of each one-vs-rest model independently. If a one-vs-rest model classifies a given question correctly, it scores a 1, otherwise, it scores a 0.

2. Average accuracy of each given question (Qacc)
   This measure evaluates the total performance of the decision model, the question classifier. If each given question is classified in a correct question type, it scores a 1, otherwise, it scores a 0.

In Qacc, classifying with a correct question type implies that all of the one-vs-rest models from the top of the hierarchy of the question taxonomy to the given question type must classify correctly.

4.6 Results

Table 3 shows the results of our question classification experiment, which is evaluated by five-fold cross-validation.

5 Discussion

First, we could increase the performance by using the information on named entities and semantic information compared to only using the words, which is the same result given in (Li and Roth, 2002). This result proved that high-performance question classification requires not only word features but also many more types of information in the question.

Table 4: Accuracy of each question (Qacc) evaluated at different depths of hierarchy in question taxonomy

d	# of QTs	W	W+N	W+S	W+N+S
HDAG-SVM					
1	5	0.946	0.944	0.953	0.948
2	25	0.795	0.794	0.800	0.803
3	55	0.741	0.743	0.751	0.756
4	68	0.730	0.736	0.742	0.749
BOW(d2)-SVM					
1	5	0.906	0.914	0.908	0.925
2	25	0.736	0.748	0.748	0.763
3	55	0.687	0.698	0.695	0.712
4	68	0.678	0.691	0.686	0.704
BOW(d1)-SVM					
1	5	0.906	0.918	0.905	0.917
2	25	0.736	0.752	0.730	0.752
3	55	0.688	0.697	0.678	0.701
4	68	0.679	0.686	0.671	0.694
BOW-SNoW					
1	5	0.862	0.870	0.880	0.896
2	25	0.635	0.640	0.687	0.696
3	55	0.570	0.582	0.623	0.634
4	68	0.562	0.573	0.614	0.626

Second, our proposed method showed higher performance than any method using BOW. This result indicates that the structural information in the question, which includes several levels of chunks and their relations, must provide powerful features to classify the target intention of a given question. We assume that such structural information must provide *shallow* semantic information of the text. Therefore, it is natural to improve the performance to identify the intention of the question in order to use the structural information in the manner of our proposed method.

Table 4 shows the results of Qacc at each depth of the question taxonomy. The results of depth d represent the total performance measured by Qacc, considering only the upper d levels of question types in the question taxonomy. If the depth goes lower, all results show worse performance. There are several reasons for this. One problem is the unbalanced training data, where the lower depth question types have fewer positive labeled samples (questions) as shown in table 2. Moreover, during the classification process misclassification is multiplied. Consequently, if the upper-level classifier performed misclassification, we would no longer get a correct answer, even though a lower-level classifier has the ability to classify correctly. Thus, using a machine learning approach (not only SVM) is not suitable for deep hierarchically structured class labels. We should arrange a question taxonomy that is suitable for machine learning to achieve the total performance of question classification.

The performance by using SVM is better than that by SNoW, even in handling the same feature of BOW. One advantage of using SNoW is its much faster learning and classifying speed than those of SVM. We should thus select the best approach for the purpose, depending on whether speed or accuracy is needed.

6 Conclusions

This paper presents a machine learning approach to question classification. We proposed the HDAG kernel, a new kernel function, that can easily handle structured natural language data.

Our experimental results proved that features of the structure in a given question, which can be computed by the HDAG kernel, are useful for improving the performance of question classification. This is because structures inside a text provide the semantic features of question that are required for high-performance question classification.

References

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proc. of Neural Information Processing Systems (NIPS'2001).*

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

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

S. Harabagiu, M. Pasca, and S. Maiorano. 2000. FALCON: Boosting Knowledge for Answer Engines. In *Proc. of the 9th Text Retrieval Conference (TREC-9)*. NIST.

D. Haussler. 1999. Convolution Kernels on Discrete Structures. In *Technical Report UCS-CRL-99-10*. UC Santa Cruz.

U. Hermjakob. 2001. Parsing and Question Classifications for Question Answering. In *Proc. of the Workshop on Open-Domain Question Answering at ACL-2001*. ACL.

E. H. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 2001. Toward Semantics-Based Answer Pinpointing. In *Proc. of the Human Language Technology Conference (HLT2001).*

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. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 390–396.

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

T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of European Conference on Machine Learning(ECML '98)*, pages 137–142.

T. Kudo and Y. Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proc. of the 6th Conference on Natural Language Learning (CoNLL 2002)*, pages 63–69.

X. Li and D. Roth. 2002. Learning Question Classifiers. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 556–562.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text Classification Using String Kernel. *Journal of Machine Learning Research*, 2:419–444.

J. Suzuki, Y. Sasaki, and E. Maeda. 2002a. Question type classification using statistical machine learning. In *Forum on Information Technology (FIT2002), Information Technology Letters (in Japanese)*, pages 89–90.

J. Suzuki, Y. Sasaki, and E. Maeda. 2002b. SVM Answer Selection for Open-Domain Question Answering. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 974–980.

J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda. 2003. Hierarchical directed acyclic graph kernel: Methods for natural language data. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, page to appear.

H. Taira and M. Haruno. 1999. Feature Selection in SVM Text Categorization. In *Proc. of the 16th Conference of the American Association for Artificial Intelligence (AAAI '99)*, pages 480–486.