Query Classification System Based on Snippet Summary Similarities for NTCIR-10 1CLICK-2 Task

Tatsuya Tojima
Nagaoka University of Technology
Nagaoka, Japan
tojima@stn.nagaokaut.ac.jp

Takashi Yukawa
Nagaoka University of Technology
Nagaoka, Japan
yukawa@vos.nagaokaut.ac.jp

ABSTRACT

A query classification system for NTCIR-10 1CLICK-2 is described in this paper. The system classifies queries in Japanese and English into eight predefined classes by using support vector machines (SVMs) for classification. Feature vectors are created based on snippet similarities instead of snippet word frequency. These vectors, which have fewer dimensions than those made from raw words, reduce the number of parameters of SVMs. Therefore, the system achieves more generalization and reduces computing resources. Two methods for calculating document similarity, cosine similarity and Jaccard index, were compared. Additionally, two snippet sources, Bing search results given by the task organizer and Yahoo! Japan Web search results, were compared. Other methods that add query string information to snippet information for the feature vectors were compared with the above methods. Our system achieved 0.89 accuracy in the English task by cosine similarity and the Yahoo! Japan Web search results, and 0.86 in the Japanese task by cosine similarity and the Bing search results.

Team Name

NUTKS

Subtask/Language

Japanese and English Query Classification Subtask

External Resources Used

Yahoo! Japan Web search

Keywords

query classification, dimension reduction, intent, web search, mobile

1. INTRODUCTION

The purpose of 1CLICK for Web searches is to satisfy users with summarized simple text after clicking the SEARCH button. This text is suitable for small screens, e.g., cellular phones, and tablet devices.

The first 1CLICK task started as a pilot task in the NTCIR-9 INTENT task [11]. NTCIR-10 1CLICK-2 has two tasks: a Main task and a Query Classification subtask. Both tasks are in English and Japanese. In the Main task, a query is given, and the output is summarized simple text. The task details can be found in the NTCIR-10 overview paper [9]. The Query Classification subtask is the first step in the Main task: a query is given and the output is a query type.

In the NTCIR-9 1CLICK task, all participants used support vector machines (SVMs) for query classifications. The TTOKU team [5] previously used the one-vs.-rest approach for a multilabel classification problem. They used the frequency of bigrams retrieved in Web pages as features of the classifier and created training datasets by crawling query-relevant URLs provided as training datasets. The rate of true positives out of the total queries was 0.28. CELEBRITY, DEFINITION, and QA were sometimes mislabeled as LOCATION. In addition, this method needs many raw Web pages as resources. It was perceived importance of reducing obstructive resources. The KUIDL team [8] used these features: “Has Wikipedia article”, “Frequency of POS”, “Query unigram”, “Sentence pattern”, “Number of documents containing expanded query”, “Has travel service”, “Number of search result”, and “Terms in search results”. The total number of features was 185. The training data, including sample queries, were created manually. The number of queries was 400, in which each class contained 100 sample queries given by the organizer. The system accuracy was 0.93. This system requires much processing time because it uses very high-dimensional feature vectors. The MSRA team [10] used a different approach based on two types of features. The first features were from query strings: “Query length”, “Appearance of Particles”, “Appearance of clue words”, and “Character type combinations”. The other features were from the Web search results: “Content words in the snippets” and “URL Hosts”. This classifier showed a 0.91 accuracy. The KUIDL team and the MSRA team use snippets for some features. Thus, their work suggests that snippets have much information for classification. However, because their systems strongly depend on the features of the Japanese language and snippets, they are not suitable for multilingual support.

In other work, Shima et al. [12] proposed Latent Semantic Indexing (LSI) for reducing the feature vector elements of SVMs. This method improved classification performance with considerable compact representation. However, LSI strongly depends on initial documents and needs recalculation when adding new documents. Thus, it, too, is not suitable for Web searches.

Nagaoka University Technology, Knowledge System laboratory (NUTKS) developed a classification system that requires less resources and takes advantage of the information

in snippets. The system has less dependency on language and more flexibility to new words. NUTKS participated in only the Query Classification subtask.

The remainder of this paper is structured as follows. Section 2 describes the details of the Query Classification subtask, and Section 3 describes our system and the methods used. Section 4 shows the evaluation result. Sections 6 presents the conclusions.

2. QUERY CLASSIFICATION SUBTASK

2.1 Input

In the NTCIR-10 1CLICK-2 task, the query types given are fine-grained compared with those given in the NTCIR-9 1CLICK task. Queries are classified into eight types: ARTIST (10), ACTOR (10), POLITICIAN (10), ATHLETE (10), FACILITY (15), GEO (15), DEFINITION (15), and QA(15). The numbers of queries are shown in parentheses after each type. A total of 100 queries are given for each task in Japanese and English. The given query format is as follows:

<query ID>[Tab]<query string>

2.2 Output

The Query Classification subtask must predict the query type for a given query. This is a multiclass query classification problem. Each line in the Query Classification run files consists of the following format:

<query ID>[Tab]<query type>

where <query type> is one of eight types predicted by the system.

3. SNIPPET-BASED QUERY CLASSIFICATION SYSTEM

3.1 System Overview

In NTCIR-9, KUIDL and MSRA used snippets for manually creating features with predefined characteristic Japanese words. However, we decided that snippets have more information. Since snippets contain the essentials of related Web pages for each query, a system using words in a snippet for a feature vector is expected to improve its performance compared with that using all words in related Web pages as the feature vector. However, the variations of words in snippets are still large and the dimension of a feature vector becomes high if each word corresponds to each axis in vector space. High-dimensional feature vectors cause overfitting and loss of generalization. To solve this problem, reduction of parameters is required.

Therefore, the advantage of our system is dimension reduction. A high-dimensional word vector of query snippets is replaced by a similarity vector that has only eight parameters. In addition, dimension reduction has a beneficial side effect. Lower-dimensional vectors reduce computer resources, and this improves SVM optimization. As a result, this method provides easier recalculation. This is suitable for Web searches in which new words appear.

Our system consists of three parts: feature extractor, search engine, and classifier. Six classification systems (snippet sources, similarity methods, and other features) were compared in each language. The system overview is shown in Fig. 1, and details of the system are given in the following subsections.

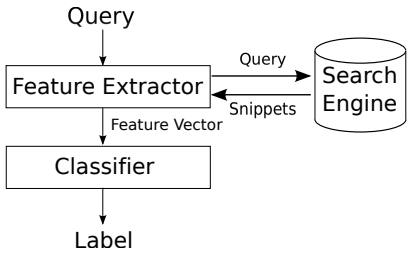

Figure 1: System overview

3.2 Vector Space Model

Snippets and query information must be given in a format that is easy to use in a computer. In the vector space model for information retrieval, a vector represents each item or document in a collection [1]. The document vector is defined as $\overrightarrow{d_n}$, $\overrightarrow{D_j}$, and the word value w_i in the formulas (1), (2), and (3). In (3),w_i shows whether a word exists in each document.

$$\overrightarrow{d_n} = \{w_1, w_2, w_3, ..., w_i\} \quad (1)$$

$$\overrightarrow{D_j} = \{d_1, d_2, d_3, ..., d_n\} \quad (2)$$

$$w_i = \begin{cases} 1 & \text{(Exists)} \\ 0 & \text{(Does not exist)} \end{cases} \quad (3)$$

3.3 Search Engine

3.3.1 Data source

The organizer provided a document collection in each language for the Main task. The document collection has 500 top-ranked documents returned by the Bing search engine for each query. This collection was constructed on July 4, 2012. Each document has a page title, summary (snippet: generated by the search engine), URL, and document rank.

In addition, we created a document collection from a Yahoo! Japan Web search[1]. These document collections were constructed on Oct. 23, 2012. Each document has the same structure as those in the Bing result. We used these document collections for the Query Classification subtask.

3.3.2 Snippets

The 100 top-ranked snippets were used for each query. Each snippet was separated into each word. We used TreeTagger[2] in English and MeCab[3] in Japanese for separating. We used only the part-of-speech (POS) tag words listed in Table 1.

[1]Yahoo! Japan Web search API: http://developer.yahoo.co.jp/webapi/search/

[2]TreeTagger: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[3]MeCab: http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

Table 1: POS tags used for systems

Language	POS-tags
Japanese	Noun (Meishi), Verb (Doushi), Adjective (Keiyoushi), Adnominal (Rentaishi), Auxiliary verb (Jodoushi)
English	CC, FW, JJ, JJR, JJS, NN, NNS, NP, NPS, PP, PP$, RB, RBR, RBS, RP, VB, VBN, VBP, VBZ, WDT, WP, WP$

3.4 Feature Extractor

Feature vectors are created according to the following formula (4) and Fig. 2. First, document vectors $\overrightarrow{d_n}$ are created from each query snippet and $\overrightarrow{D_j}$, which includes the same label of $\overrightarrow{d_n}$. Second, the similarity is calculated between document vectors with the query document vector. Finally, all similarities are joined as a vector by formula (4).

$$\overrightarrow{f} = \{sim(\overrightarrow{D_1}, \overrightarrow{q}), sim(\overrightarrow{D_2}, \overrightarrow{q}), ..., sim(\overrightarrow{D_8}, \overrightarrow{q})\} \quad (4)$$

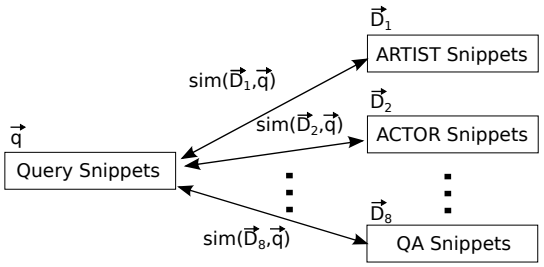

Figure 2: Calculating similarity in the feature extractor

3.4.1 Methods of calculating document similarity

The following two methods of calculating document similarity are compared to determine the most suitable for each Japanese and English task.

1. Cosine similarity

Cosine similarity [4] between $\overrightarrow{D_j}$ with $\overrightarrow{q}$ is defined by formula (5).

$$Sim(\overrightarrow{D_j}, \overrightarrow{q}) = \frac{\overrightarrow{D_j} \cdot \overrightarrow{q}}{|\overrightarrow{D_j}||\overrightarrow{q}|} \quad (5)$$

$\overrightarrow{q}$: A query document vector (same format as $\overrightarrow{d_n}$)

2. Jaccard index

Jaccard index (Jaccard similarity coefficient) [4] between $\overrightarrow{D_j}$ with $\overrightarrow{q}$ is defined by formula (6).

$$Sim(\overrightarrow{D_j}, \overrightarrow{q}) = \frac{|\overrightarrow{D_j} \cap \overrightarrow{q}|}{|\overrightarrow{D_j} \cup \overrightarrow{q}|} \quad (6)$$

3.4.2 Other features from query strings

Methods that add some features in a query string to the document similarity are compared with the methods mentioned above. Additional features include morpheme length, frequency of POS, interrogative words, and question words.

1. Morpheme length

A value based on the query word length. If a query consists of 3 words, this value is 3. If length > 5, this value is 5.

2. Frequency of POS

Part-of-speech tag counts of each query word. The POS tags used are listed in Table 1.

3. Interrogative words

A query that includes interrogative words (Boolean). Examples: who (dare), what (nani).

4. Question words

A query that includes one of the question words at the end of the sentence in Japanese or the beginning of the sentence in English (Boolean).
Examples:
Japanese: “towa”, “ka”, “?”.
English: “do”, “can”, “should”.

3.5 Classifier

3.5.1 Support vector machine

An SVM is used for the classifier. SVMs, also called support-vector networks [3], are supervised learning models in machine learning. The SVM determines the separating hyperplanes, as shown in Fig. 3. The support vectors, marked with gray, define the margin of largest separation between two classes.

The advantage of SVMs is their ability to ensure high generalization with a small number of vectors. This characteristic is compatible with the given query information.

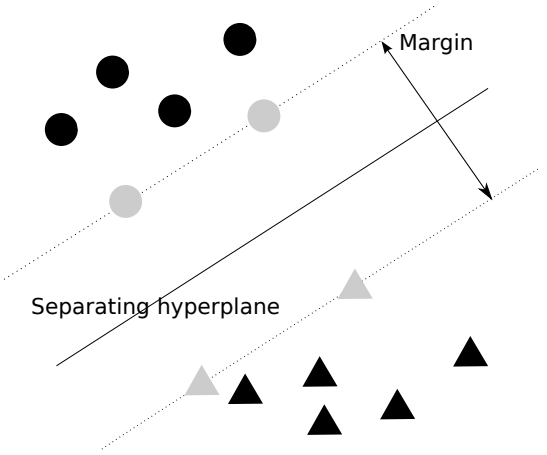

Figure 3: Strategy of SVMs

LIBSVM [2] is used in this task. LIBSVM implements the one-against-one approach [7] for multi-class classification. Hsu and Lin [6] reported that the one-against-one method is suitable for practical use.

3.5.2 *Configuring the classifier*

C-Support Vector Classification (C-SVC) and the polynomial kernel were selected for configuring the classifier. All parameters were optimized by the result created from our labeled queries. We used 99 queries for learning and only one query for testing. This 100-fold cross-validation is called leave-one-out cross-validation. The feature vectors for learning are created in the following steps.

1. Create vectors $\overrightarrow{d_1}$ to $\overrightarrow{d_{99}}$ by the snippets of each learning query (i.e., not the test query).
2. Select one $\overrightarrow{d_n}$ from the 99 $\overrightarrow{d_n}$ vectors.
3. Create $\overrightarrow{D_j}$ from $\overrightarrow{d_n}$ vectors for each label ($j = 1 - 8$).
4. Calculate similarities between $\overrightarrow{D_j}$ with the selected $\overrightarrow{d_n}$.
5. Repeat $1 \sim 4$ for all queries (99 times).
6. If "Other features" from $\overrightarrow{q}$ are used, they are added.

The above procedure means $\overrightarrow{q}$ is always excluded from $\overrightarrow{D_j}$ in learning and testing sequences.

4. RESULT

Six runs were submitted by each language query classification subtask. The run name format is shown below.

<team>-QC-<priority>

Odd priority numbers are English runs and even priority numbers are Japanese runs. Each run consists of the combinations of methods and sources.

Three method combinations are used: only Cosine similarity feature vector (Only Cosine), only Jaccard Index feature vector (Only Jaccard), and cosine similarity feature + other features from query strings vector (Cosine + Other features). Two sources are used: Bing search results (Bing) and Yahoo! Japan Web search results (Yahoo).

Section 4.1 describes the result of the Query Classification subtask in English and Section 4.2 shows the result in the Japanese task.

4.1 English Task

Table 2 shows the results of the accuracy by each submitted run of the Query Classification subtask in English.

NUTKS-QC-3 (Only Cosine similarity & Yahoo! Web search result) has the highest score in six English runs. Its accuracy is 0.89. This Query Classification subtask has more classes than those in the NTCIR-9 subtask, so this method achieves a good score. In addition, the Yahoo! Web search result tends to give a better result than the Bing search result in the English task. Moreover, the results of Cosine + Other features were lower than Only Cosine.

Table 3 shows the result of the number of true positives (TP) and false positives (FP) in the English task for each query type. This table indicates that celebrity query types (ARTIST, ACTOR, POLITICIAN, and ATHLETE) can be classified correctly. However, it is more difficult to identify FACILITY, GEO, DEFINITION, and QA query types. Some FACILITY query types tend to be misclassified as GEO, and some GEO query types tend to be misclassified as FACILITY. Similarly, some DEFINITION query types tend to be misclassified as QA, and some QA query types misclassified as DEFINITION.

Table 2: Result of submitted English runs

Run name	Methods	Source	Accuracy
NUTKS-QC-1	Only Cosine	Bing	0.84
NUTKS-QC-3	Only Cosine	Yahoo	0.89
NUTKS-QC-5	Only Jaccard	Bing	0.79
NUTKS-QC-7	Only Jaccard	Yahoo	0.86
NUTKS-QC-9	Cosine + Other features	Bing	0.81
NUTKS-QC-11	Cosine + Other features	Yahoo	0.84

Table 3: Number of true positives (TP) and false positives (FP) in the English task

Run name (NUTKS-QC-)		1	3	5	7	9	11
ARTIST	TP	9	10	9	10	9	9
	FP	0	0	1	1	0	1
ACTOR	TP	9	9	8	8	9	8
	FP	1	0	1	0	1	1
POLITICIAN	TP	10	10	10	10	9	10
	FP	2	3	2	3	2	1
ATHLETE	TP	10	10	9	10	9	10
	FP	1	0	0	0	0	0
FACILITY	TP	12	12	10	11	9	10
	FP	2	1	4	1	5	3
GEO	TP	11	13	12	13	10	11
	FP	3	2	6	3	8	8
DEFINITION	TP	10	13	10	13	14	14
	FP	2	2	3	3	3	2
QA	TP	13	12	11	11	12	12
	FP	5	3	4	3	0	0
Accuracy		0.84	0.89	0.79	0.86	0.81	0.84

4.2 Japanese Task

Table 4 shows the results of the accuracy by each submitted run of the Query Classification subtask in Japanese.

NUTKS-QC-2 (Only Cosine similarity & Bing search result) and NUTKS-QC-10 (Cosine + Other features & Bing search result) achieve accuracy 0.86.

In contrast to the English task, the Bing search result is more suitable than the Yahoo! Japan Web search result in the Japanese task. However, the source influence is less than that in the English task. In addition, no significant difference was observed by adding other features.

Table 5 shows the result of the number of true positives (TP) and false positives (FP) in the Japanese task. This result has the same tendency as the English task. The celebrity queries are classified correctly and it is difficult to identify FACILITY, GEO, DEFINITION, and QA query types. Moreover, the tendency of misclassified types is also the same as the tendency in the English task.

Table 4: Result of submitted Japanese runs

Run name	Methods	Source	Accuracy
NUTKS-QC-2	Only Cosine	Bing	0.86
NUTKS-QC-4	Only Cosine	Yahoo	0.84
NUTKS-QC-6	Only Jaccard	Bing	0.84
NUTKS-QC-8	Only Jaccard	Yahoo	0.82
NUTKS-QC-10	Cosine + Other features	Bing	0.86
NUTKS-QC-12	Cosine + Other features	Yahoo	0.85

5. DISCUSSION

Table 5: Number of true positives (TP) and false positives (FP) in the Japanese task

Run name (NUTKS-QC-)		2	4	6	8	10	12
ARTIST	TP	10	9	9	8	10	9
	FP	1	3	1	2	1	2
ACTOR	TP	10	10	10	10	10	9
	FP	0	1	0	2	1	2
POLITICIAN	TP	9	10	9	10	9	10
	FP	2	0	2	0	1	0
ATHLETE	TP	10	9	10	9	9	9
	FP	0	0	0	0	0	0
FACILITY	TP	11	11	11	10	13	11
	FP	4	3	4	3	5	5
GEO	TP	12	13	12	13	12	14
	FP	2	4	2	4	1	4
DEFINITION	TP	11	11	11	11	10	10
	FP	4	4	6	6	3	0
QA	TP	13	11	12	11	13	13
	FP	1	1	1	1	2	2
Accuracy		0.86	0.84	0.84	0.82	0.86	0.85

Overall, this system could achieve good results without language dependency by the simple method (Only Cosine or Only Jaccard). The fact suggests that a similar method will apply to other languages. However, the differences of search engines for each language must be considered. The search engine difference suggests what is important in snippets. In addition, other features are not appropriate for this method, although this may be caused by insufficient SVM parameter tuning.

For details of the classification result, some search results of query strings include the same words; that is, these query strings are closely related in semantic space. Normally, it is difficult to classify the correct class. Nevertheless, most of the queries could be classified into the correct class as expected, due to the use of not only one similarity but also other similarities. It can be considered that using other similarities shifts the hyperplane separating two classes by the SVM. In any case, FACILITY and GEO results were less sufficient than the celebrity query types because they are too close in semantic space. Therefore, these types need improved separation methods. DEFINITION and QA also have the same problem. These problems may be remedied by adding preprocessing to the search result or by constructing multistage classification. These options are still under investigation.

6. CONCLUSIONS

The query classification system based on snippet summary similarity achieved accuracy 0.89 in the English subtask and 0.86 in the Japanese subtask.

This method is suitable for celebrity classifications and is likely to apply to other languages. Further studies are needed to distinguish FACILITY/GEO and DEFINITION/QA. The influence of the source on the search results also requires further investigation to identify what is important in snippets.

7. REFERENCES

[1] M. Berry, Z. Drmac, and E. Jessup. Matrices, vector spaces, and information retrieval. *SIAM review*, 41(2):335–362, 1999.

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] D. Grossman and O. Frieder. *Information retrieval: Algorithms and heuristics*, volume 15. Springer, 2004.

[5] T. S. H. T. Hajime Morita, Takuya Makino and M. Okumura. Ttoku summarization based systems at ntcir-9 1click task. *In Proceedings of NTCIR-9, to appear*, 2011.

[6] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[7] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In J. Fogelman, editor, *Neurocomputing: Algorithms, Architectures and Applications*. Springer-Verlag, 1990.

[8] K. T. Y. S. T. Y. H. O. Makoto P. Kato, Meng Zhao and K. Tanaka. Information extraction based approach for the ntcir-9 1click task. *In Proceedings of NTCIR-9, to appear*, 2011.

[9] V. P. T. S. T. Y. Makoto P. Kato, Matthew Ekstrand-Abueg and M. Iwata. Overview of ntcir-10 1click-2 task. *In Proceedings of NTCIR-10, to appear*, 2013.

[10] Y.-I. S. Naoki Orii and T. Sakai. Microsoft research asia at the ntcir-9 1click task. *In Proceedings of NTCIR-9, to appear*, 2011.

[11] T. Sakai, M. P. Kato, and Y.-I. Song. Overview of the ntcir-9 1click. *In Proceedings of NTCIR-9, to appear*, pages 180–201, 2011.

[12] K. Shima, M. Todoriki, and A. Suzuki. Svm-based feature selection of latent semantic features. *Pattern Recognition Letters*, 25(9):1051–1057, 2004.