Towards Domain Independent Why Text Segment Classification Based on Bag of Function Words

Katsuyuki Tanaka, Tetsuya Takiguchi, and Yasuo Ariki

Kobe University
1-1 Rokkodai, Nada, Kobe 657-8501, Japan
katsutanaka@puppy.kobe-u.ac.jp,{takigu,ariki}@kobe-u.ac.jp

Abstract. Increased attention has been focused on question answering (QA) technology as next generation search since it improves the usability of information acquisition from web. However, not much research has been conducted on **"**non-factoid-QA**",** especially on *Why Question Answering* (*Why-QA*). In this paper, we introduce a machine learning approach to automatically construct a classifier with function words as features to perform *Why Text Segments Classification* (*WTS* classification) by using SVM. It is a process of detecting text segments describing *"reasons-causes"* and is a subtask of *Why-QA* mainly related to an answer extraction part. We argue that function words are a strong discriminator for *WTS* classification. Furthermore, since function words appear in almost all text segments regardless of the domain of the topic, it also enables construction of a domain independent classifier. The experimental results showed significant improvement over state-of-the-art results in terms of accuracy of *WTS* classification as well as domain independent capability.

Keywords: Non-Factoid QA, Classification, Machine Learning.

1 Introduction

The recent progression of internet technology has increased with the number of internet users. Trends such as the development of online knowledge bases like Wikipedia and community portal sites such as Yahoo!Answers have emerged, and the diversity of information now available on the internet has increased. That has lead to the dramatic growth of information availability on the internet, and as such it has become increasingly difficult for users to acquire the information that they really need. It requires changes to the way of obtaining information, from simple knowledge acquisition to complicated or deeper knowledge acquisition.

Increased attention has been focused on the question answering (QA) technology as next generation search. This is because QA systems return a list of exact answers as search results while most of the commercial information retrievers, such as Google, return a list of documents. Returning the list of exact answers reduces the labour intensive filtering process to obtain information since it does not need to look into each document to find chunks of information from the lists of retrieved documents, which are often very large.

M. Thielscher and D. Zhang (Eds.): AI 2012, LNCS 7691, pp. 469–480, 2012.

A significant amount of literature on QA has reported on **"**factoid-QA", which deals with a question asking for a fact that can be answered by few words (*what is the height of Mt. Everest?*), and achieved high performance in terms of the answer acquisition [3, 8, 9]. However, not much research has been conducted on **"**non-factoid-QA**"**, which requires more complicated question answering mechanisms to obtain the answers (*what is non-factoid question answering?* or *why is the sky blue?*). Especially *Why Question Answering* (*Why-QA*) is not a very active area of research on non-factoid-QA field. *Why-QA* is a process of finding answers describing *"reasons-causes"* (*why-answer*) for a question asking the *"reasons-causes"* for some facts (*why-question*). The main obstacles of under-developed *Why-QA* techniques are that it becomes increasingly difficult to obtain *why-answers* since it requires deeper understanding of text content than that of factoid-QA.

Among the several recent works explored on *Why-QA* methods, the most popular method is a rule based method (*RB* method) [6, 16, 18]. *RB* method detects *why-answer* by referring to a manually predefined list of keyword cues or patterns based on *"reasons-causes"* characteristics, which are called 'rule dictionary'. However, the rule dictionary construction is laborious and the performance of *RB* method is not very stable in terms of *why-answer* extraction accuracy.

As a subtask of *Why-QA*, Tanaka [4] developed a machine learning approach to detect a group of sentences, a text segments (*TS*), describing *"reasons-causes"* based on "bag-of-words (*BOW*)" representation. Even though *BOW* effective representation of text to deal with topic classification, since the vocabulary size of nouns is very large and they carry domain dependent information, they increase the computation of building a classifier while decreasing the domain independency of the classifier. Moreover, most of the nouns are not effective discriminators to detect *TS* describing *"reasons-causes"* hence *BOW* may not be an optimal representation to apply in such a task.

The objective of our research is to introduce the methodology of automatically building a highly discriminative classifier to detect *TS* indicating *"reasons-causes"* regardless of the domain of *TS*. The classifier is constructed by making use of function words, which are usually ignored by most of the *BOW* based information retrieval research, as bases of feature space for machine learning. We call such *TS* describing *"reasons-causes"* as *"Why Text Segment (WTS)"* and *TS* that is not *WTS* as "*NotWhy Text Segment (NWTS)"*. We define the process of detecting *WTS* as *WTS* classification and the classifier to perform such a classification as *WTS* classifier. *WTS* classification is a subtask of *Why-QA* mainly related to Answer Extraction part of QA system. Here, *TS* could be some answers on an online forum or community portal or chunks of sentences extracted from any web page. Domain means a group of words or terms share the same concept such as sports, science, finance, and so forth.

As an example of *WTS* classification, consider the following three *TS* extracted from Wikipedia[1] and one of its reference links[2] related to the topic of the sky. It is clear that 1 is an explanation of "sky" while 2 and 3 state the reason why the sky is blue or yellow (red).

---

[1] `http://en.wikipedia.org/wiki/Sky`

[2] `http://math.ucr.edu/home/baez/physics/General/BlueSky/blue_sky.html`

1. "The sky is the part of the atmosphere or of outer space visible from the surface of any astronomical object."
2. "The light from the sky is a result of the scattering of sunlight, which results in a light blue colour perceived by the human eye. On a sunny day Rayleigh Scattering gives the sky a blue gradient - dark in the zenith, light near the horizon."
3. "When the air is clear the sunset will appear yellow, because the light from the sun has passed a long distance through air and some of the blue light has been scattered away. If the air is polluted with small particles, natural or otherwise, the sunset will be more red.

Applying *WTS* classification on these *TS* means to classify 2 and 3 as *WTS* and 1 as *NWTS*.

Like *BOW*, we call bag representation of function words *"bag-of-function-words (BOFW)"*, therefore, we refer to our proposed method as *BOFW* method. In this paper, *WTS* classification is considered as a binary classification into binary classes, *WTS* and *NWTS* as binary classes and we use SVM [15] to build the classifier.

Our research is similar to existing literature [5, 13, 14] in terms of utilizing function words, but our approach differs in the way of utilizing the function words. Our method only uses morpheme based function word as a unit for a machine learning feature whereas [5] use structured clause with function words as a feature and [13], [14] use more than function words. Moreover, we propose machine learning frame work as classification to discriminate *WTS* rather than ranker learning to re-rank *TS* according to *WTS* [5]. The proposed *BOFW* method does not require laboriously labelled training data [5], or deep language analyses [21] to choose features.

Even though *BOFW* provides limited contribution in terms of topic classification (*BOW* is more useful for topic classification), the focus of this research is not about classification based on topic, but it is classification of *WTS*. Consequently, the essence of the proposed *BOFW* methodology is that despite its simplicity, it provides a strong discriminative power for *WTS* classification regardless of domain of *TS*. Hence it could provide a simple yet effective way of boosting the performance of QA system to build finer *Why-QA* technology. Moreover, even though this research is conducted on Japanese, it is adaptable to different languages by simply changing the *BOFW* definitions as we have proved in [11]. In this paper, we describe the details of *BOFW* methods and its set up in Japanese. In addition we discuss the domain independent issues of *WTS* classification based on *BOFW* method which has not stated in [11]. In summary, the main contributions of this paper are as follows:

- **Classification performance issue:** We experimentally show that *BOFW* method boosts the performance of *WTS* classification, yielding performance of prior works.
- **Domain independent issue:** *WTS* classifier with *BOFW* method provides stable classification performance regardless of the domain of *TS* than any prior works.

The rest of this paper is organised as follows. Section 2 describes the related work done on *Why-QA*. Section 3 states our proposed *BOFW* approaches to construct *WTS* classifier. Finally, Section 4 discusses the experiment.

2 Related Work

Shibusawa [16] proposed a method to extract the location of a sentence with *"reasons-causes"* as *why-answer* with respect to the contiguous "fact sentence", which defines a sentence including all keywords on a question. Shibusawa defines four possible locations of the *why-answer*, 1) pre Case, 2) and 3) post Case, and 4) within Case regard with the "fact sentence". A location of *why-answer* is determined by appearance patterns of rules with respect to "fact sentence" by refering to manually constructed rule dictionary. Rule dictionary contains the list of rules such as the reason keywords ("kara", "karakoso", etc) and reference terms (such as "ikano", "tsugi", etc). This is a typical *RB* method as it defines such rules manually from manually extracted terms which characterises *"reasons-causes"*. The problem with a *RB* method is that it is a very troublesome and labour-intensive task to produce a rule dictionary. Besides, it may not be possible to find all rules to define *"reasons-causes"* characteristics. Hence the lack of rule coverage causes an inaccuracy of *why-answers* detection from corpus.

Instead of building a rule dictionary manually, Higashinaka and Isozaki [5] proposed automatic rule extraction methods to overcome the rule coverage problem in *RB* method by exploiting Japanese EDR dictionary. EDR dictionary is a collection of Japanese sentences, in which terms or phrases are labelled with its semantic role to represent semantic relations. Labelling is maintained by linguists manually. Higashinaka extracted sentences labelled with "reason" from EDR and decomposed all the sentences into clauses. Then, all content words in each clause were replaced by "*" to form a clause structured with function words (we call it a "structured clause with function words"). Higashinaka then trained SVM ranker based on these structured clauses with function words as well as manually extracted *"reasons-causes"* terms. They considered the *Why-QA* as ranking problem to boost *WTS* in higher ranking. Although the EDR approach may be able to construct why-type rules efficiently, the accessibility of such a commercial dictionary is not easy and costly. Besides, this method requires training dataset labelled with ranking according with *WTS*, which is not easy to obtain.

Intensive studies on *Why-QA* have been undertaken by Verberne [19, 20, 21]. Verberne [21] regarded *Why-QA* as re-ranking of *TS* retrieved by Wumps Search Engine. Verberne carried out deep natural language analysis on English sentence structure and utilised syntactically analysed information of *TS* to re-rank in the order of *WTS*. However, since it requires heavy language analysis and deep natural language processing skill, it is not easily adaptable to different languages.

Tanaka [4] proposed a machine learning [1, 2] method based on *BOW* to perform *WTS* classification. However, this method has a domain dependency problem on the produced classifier. This is because *BOW* representations of *TS* include noun information, and nouns are very domain specific information. Therefore, trained classifier has a bias towards the training data and it is not suitable for a domain independent classification. It also requires a re-collecting of training data and re-training to build classifier for different task and it is a troublesome for a large data set.

Brill [17] proposed a statistical translation technique to answer various types of non-factoid questions. Brill produced language model based on large sets of parallel corpus. However, the performance of the model is highly dependent on the quality of the word translation probabilities. It requires a large amount of semantically similar yet lexically different data for training the model to capture better correlations of words and such corpus is not easily available. Moreover, most of non-factoid type QA research is not specialised in *Why-QA* [14, 17] and some of them suffer from accuracy of the results on *Why-QA*. This may be due to the adaptation of factoid-QA framework into non-factoid-QA even though the representation of answers is different between factoid-QA and non-factoid-QA.

3 Domain Independent Why Text Segment Classifier Based on Bag of Function Words

In order to build *WTS* classifier with SVM, it is necessary to collect labelled data, define representations of *TS*, choose appropriate bases for features space, and build the classifier by learning patterns from training data. These are important processes not only to construct a classifier with good performance, but also to decide the task of classification. In this paper, we introduce the method of domain independent *WTS* classifier construction by clarifying these points.

In this paper, the task is clearly *WTS* classification that discriminates *WTS* or *NWTS* of *TS* input. We consider *WTS* classification as binary classification problem with *WTS/NWTS* as its classes. We used Yahoo!Answer to automatically collect and label datasets for training and testing. We trained *WTS* classifiers with SVM with function words as features. Figure1 describes the overview of *BOFW* method.

3.1 Collecting Data

Manual data labelling [4, 5, 16] is laborious and causes a problem of collecting large number of data. To overcome such shortcomings, we introduce the automatic as well as systematic way of collecting and labelling of data using Yahoo!Answer (known as Yahoo!Chiebukuro[3] in Japanese). To retain the quality and reliability of answers, we only used answers from best-answer-corpus and we used each answer as *TS*. The processes of collecting *TS* from Yahoo!Answer corpus are as follows.

Firstly, *why-questions* were collected from question-corpus. We defined *why-questions* as the questions containing keywords with typical question style of seeking for *why-answers*. We choose "naze (why…)", "doushite (why…)" and "no riyuu ha nani (what is the reason…)" as such keywords. Subsequently, we collected an answer paired with each *why-question* as *why-answer* from best-answer-corpus. The group of collected *why-answers* is called *WTS* dataset.

Similarly, we defined questions with keywords, such as "no houhou ha nani (what is the methods of….)" and "no chigai ha (what is the difference between…)", which

[3] http://chiebukuro.yahoo.co.jp/

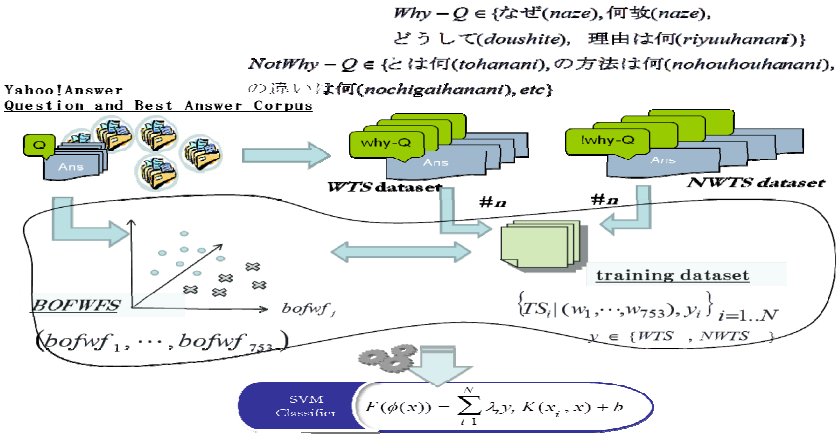

Fig. 1. Overview of *BOFW* method

are not typical question styles of seeking for *why-answers* as *notwhy-questions*. The answers paired with the *notwhy-questions* were collected as *notwhy-answers* and they are called *NWTS* dataset.

Training datasets for machine learning and test datasets for evaluation are produced by randomly selected *n TS* from *WTS* and *NWTS* datasets with no overlapping.

3.2 Bag of Function Words Method

We propose the methodology to build domain independent *WTS* classifier. A domain independent means that *WTS* classifier has the capacity of recognising any incoming *TS* regardless of its domain. In order to do so, it is important to design features representing *TS* as well as the bases of feature space properly so that it is possible for learners to capture patterns of training data to produce accurate *WTS* classifier. We leverage function words to achieve our goal.

In this paper, we define a morpheme of a function word with its part of speech as a *function-word-feature (fw-feature),* extracting such features from *TS* is called *fw-feature extraction*. In this paper, we define the process of the feature extraction as a breaking *TS* into some form of units and a unit as a feature. Furthermore, we define a group of unique units as a bag and a bag representation of such *fw-feature* is called *BOFW*. It is clear that a unique representation of *fw-feature* is *BOFW* hence we call such features as *BOFW-feature*. All the *fw-feature* extractions were conducted by using morphological analysis tool ChaSen [12] in this paper. If a unit is a *word-feature* = "any word with a part of speech" then the group of unique *word-features* is a well knowledge *BOW* representation. Figure2 lists the type of part of speech used for *BOFW* together with an example of *fw-features extraction* and in the form of *BOFW* from the sentence "tesuto na node, gakkou ni i tta. (I went to school because there is test.)".

Clearly, we use *BOFW-features* as bases of machine learning feature space. In a machine learning scheme, the choice of bases are important for a learner to capture

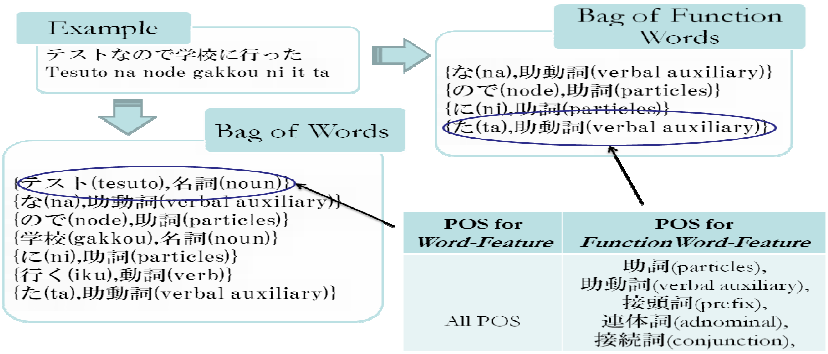

Fig. 2. Example of *function-word-features, BOFW-features, word-features*, and *BOW-features*

appropriate patterns to build an accurate classifier. We constructed such feature space by *BOFW-feature*s extracted from all answers in best-answer-corpus on Yaoo!Answer. We collected 753 *BOFW-feature*s as bases $\{bofwf_j \in \Re^{753}\}$. We randomly collected *n WTS* and *NWTS* from *WTS* dataset and *NWTS* dataset respectively, and for each drawn *TS*, we created a 753 dimensional vector with a weight w_j as an element of $bofwf_j$. In this paper, we used *tf-idf* to calculate a weight w_j for $bofwf_j$. The calculations of *tf-idf* were made based on the frequency of $bofwf_j$ in best-answer-corpus as well as its document frequency (precisely, *TS* frequency). We used $N=2n$ vector quantised *TS* as training dataset $\{TS_i \mid tf\text{-}idf_i \ , \ y_i\}_{i=1..N}$ with $tf\text{-}idf \in \Re^{753}$, $y \in$ (*WTS*, *NWTS*). Figure.1 describes these processes. The intention of this weighting scheme is to normalise terms in *TS* so that all *TS* have same weighting scheme because the length of *TS* may varies.

The rationale behind the *BOFW* method to construct *WTS classifier* is that 1) It is an effective discriminator to perform *WTS* classification and 2) because the function words appear almost in all text so that it is possible to construct domain independent *WTS classifier.*

4 Experiments

In this paper, the main purpose of our research is to carry out experiments on *WTS* classification and its evaluation. It is clear that filtering process for *WTS* detection is an important part of *Why-QA* process. The extracting the actual position of the answer and fully functional implementations of *Why-QA* are beyond our scope therefore they are left as our future works.

4.1 Experimental Setup

As an evaluation, we compare the performance of *WTS* classifier produced by *BOFW* method with 5 different baselines: *RB* method [16], *EDR+RB* and *EDR* methods based on [5] and *BOW* and *BOW-BOFW* methods based on [4]. It is difficult to reconstruct all the baselines since some of them require more than simply labelled data

[5, 16], different evaluation method [5], or different experimental setup. Therefore, we adapt baseline methods into machine learning classification framework by defining features mainly based on their rule dictionary so that we can conduct the evaluation smoothly and fairly.

Baseline.1 (*RB method*) : Among the rules in rule dictionary on Table.1 of [16], we defined each rule under "reference terms with reason-cause" and "reason-cause terms" as *rule-based-feature (RB-feature)*. We collected 83 such *RB-features* and all of them are used as bases of feature space. *RB-feature* is a binary feature indicating 1 if it exists in *TS* and 0 otherwise and. *RB-features* are typical terms indicate *"reasons-causes"* such as "dakara (because)", "riyuu (reason)" and "genninn (cause)" etc.

Baseline.2 (*EDR+RB method*) : In [5], 399 features (f1-f399) are used to as bases of feature space to train a SVM ranker. Among the features defined in [5], f1-f394 are rules, indicating *"reasons-causes"*, extracted from EDR, f395 is a feature with a list of manually extracted rules, f396-f398 are related to topic information, and f399 is related to question. We defined features for baseline.2 method as combination of automatically and manually extracted rules indicating *"reasons-causes"*. Since features f396-f398 and f399 are not directly related to our scope of *WTS* classification they were discarded. By referring to the method of rules extraction [5], we collected sentences labelled with "reason" from EDR, and they were transformed into a structured clause with function words as it is described in Section 2. We obtained 593 most frequently occurred such structure as *EDR-features*. Element of *EDR-features* are binary indicating the existence of the attributes in *TS* as 0/1. As for manual rules for f395, we used *RB-features* from baseline.1. We defined a binary feature representing the existence of any rules matched with *RB-features* in *TS* by 0/1. The bases of feature space for *EDR+RB* method, therefore, is 594 dimensional features with 593 binary *EDR-features* and 1 binary feature with a list of *RB-features.*

Baseline.3 (*EDR method*) : This baseline only use 593 *EDR-features* stated above.

Baseline.4 (*BOW method*) : By following the experimental setup of [4], we extracted words from all *TS* in training dataset and use *BOW* as bases of feature space. As for an element of *BOW-feature*, we obtained *tf-idf* according with the *BOW-feature's* term frequencies and *TS* frequencies.

Baseline.5 (*BOW-BOFW method*) : This feature space is formed by subtracting *BOFW-features* from *BOW-features* in *BOW* method.

We conducted two experiments to evaluate the effectiveness of *BOFW* method. The first experiment shows the effectiveness of on *BOFW* method in term of comparative accuracy of *WTS* classification against baselines. The purpose of the second experiment is to evaluate domain independent classification ability of *WTS* classifier constructed by *BOFW* method. All evaluations were done by comparing F-Score and the rate of correctly classified *TS* of proposed method and baselines. We used SMO [10], provided in data mining software Weka [7], with the first order of polynomial kernel ($K(x_i,x_j)=(x^T x+1)$) to train five *WTS classifiers*. We also performed paired-t-test to show the statistical significance of the experimental results.

4.2 Effectiveness on *WTS* Classification Accuracy

To train *WTS* classifier, we produced five datasets for each containing randomly selected 5000 *WTS* and *NWTS* from *WTS* and *NWTS* datasets collected in 3.1 respectively (they are called $D_{10k.[1..5]}$). To record F-Score and classification accuracy, we used one dataset of $D_{10k.[1..5]}$ to train *WTS* classifier and others as test dataset. We repeated this process 5 times and take the macro-averages of F-score and the rate of correctly classified *TS* for each method. Table1 shows the results of evaluations. A value of inside the bracket on each baseline shows the difference of *BOFW* method and baseline method.

It was found that it is possible to construct *WTS* classifier with F-Score=0.661 with 63.25% of *WTS* classification accuracy by using *BOFW* method. The results show that the performance of *WTS* classifier produced by *BOFW* method outperforms baseline methods (*RB, EDR+RB, EDR, BOW*) by 4.5%~16.3% on F-Score and 1.8%~5.3% on classification accuracy.

To check the statistical significance of the results, we performed paired-t-test on both F-Scores and classification accuracies. All results on paired-t-test against baselines showed significant difference of the results at the level of 0.01.

One of the reasons why the results of the *BOFW* method outweigh baseline methods is that *BOFW-features* work more effectively to form hyper-plane that separates *WTS* and *NWTS* class on SVM learning process. This can be explained by comparing the results of *BOFW* method, *BOW* method, and *BOW-BOFW* method.

As it is stated in 4.2, *BOW* method construct the classifier with both content words and function words, while *BOW-BOFW* method only use content words. Now the results on *BOW* method and *BOW-BOFW* method showed that it is possible to perform *WTS* classification with F-Score=0.617 with 60.20% of classification accuracy and F-Score=0.56 with 57.95% classification accuracy respectively. Clearly, the performance of *WTS* classification dropped significantly by discarding *BOFW-features* from *BOW-features*. This indicates the *BOFW-features* provide a significant contribution in order to form an effective decision boundary to distinguish *WTS* and *NWTS* classes. This also can be supported by the results of *BOFW* method, only using *BOFW-features* provides higher discriminative *WTS* classifier than *BOW* method and *BOW-BOFW* method.

Table 1. Average F-Score and Correctly Classified Rate of *WTS* classifiers using $D_{10k.[1..5]}$

	BOFW	***RB***	***EDR+RB***	***EDR***	***BOW***	***BOW-BOFW***
F-Score	0.661	0.499 (-0.163)*	0.605 (-0.056)*	0.584 (-0.077)*	0.617 (-0.045)*	0.596 (-0.065)*
Correct Classified	63.25	60.57 (-2.68)*	61.43 (-1.82)*	59.11 (-4.14)*	60.20 (-3.05)*	57.95 (-5.30)*

*paired-t test with significance at a level of 0.01.

4.3 Capability of *BOFW* Method as Domain Independent *WTS* Classification

Yahoo!Answer provides various topics in answers and it can be considered as open domain corpus. Therefore, from the experimental results in 4.2, it is possible to say that *BOFW* method is capable of classifying any *TS* regardless of its domain. We conducted further experiment to test the effectiveness of the proposed method as a domain independent classifier. Evaluation of this experiment is conducted by creating *WTS* classifier with training dataset containing only one domain and test the classifier by test dataset not containing the domain. Since Yahoo!Answer provides various categories sharing the same topic, we define a category as a domain and created datasets with/without domain as follow.

First, we extracted *WTS* and *NWTS* only belonging to one category to form a dataset D_{cat} and *TS* not belonging to the category to form a dataset D_{nocat}. In order to obtain enough data, we only used categories provides more than 3000 *WTS* and *NWTS on WTS* datasets and *NWTS* datasets to create D_{cat} There were 4 such categories. From D_{nocat} we randomly selected 5000 *WTS* and *NWTS* each and created 5 datasets $D_{nocat[1..5]}$. In effect, we have 4pairs of D_{cat} and $D_{nocat[1..5]}$.

Evaluation method is the same as 4.2, we compared the performance of *WTS* classifier by comparing F-Score and classification accuracy of *BOFW* method with baselines. F-Score and classification accuracy were recorded by testing each $D_{nocat[1..5]}$ on *WTS* classifier build by D_{cat} as training dataset. Similarly, we recorded F-Score and classification accuracy of 5 *WTS* classifier trained by $D_{nocat.[1..5]}$ tested by D_{cat}. Table2 shows F-Score and classification accuracy macro-average of evaluation results of 4 categories (4 D_{cat} x $D_{nocat.[1..5]}$ = 20 results per each evaluation) and its total average.

Table 2. Average F-Score and Correctly Classified Rate Experimental Results of *WTS* classifiers using D_{cat} and $D_{nocat.[1..5]}$

	BOFW	***RB***	***EDR+RB***	***EDR***	***BOW***	***BOW-BOFW***
	D_{cat} classifier vs $D_{nocat.[1..5]}$ test datasets					
F-Score	0.636	0.487 (-0.149)*	0.568 (-0.068)*	0.567 (-0.069)*	0.591 (-0.045)*	0.563 (-0.072)*
Correctly Classified	61.99	60.20 (-1.79)*	59.09 (-2.90)*	57.60 (-4.38)*	58.41 (-3.58)*	55.34 (-6.65)*
	$D_{nocat.[1..5]}$ classifiers vs D_{cat} test dataset					
F-Score	0.626	0.475 (-0.151)*	0.574 (-0.053)*	0.562 (-0.063)*	0.577 (-0.048)*	0.555 (-0.070)*
Correctly Classified	61.01	58.11 (-2.90)*	59.07 (-1.94)**	58.15 (-2.86)*	58.38 (-2.63)*	56.17 (-4.84)*
	Average Results					
F-Score	0.632	0.481*	0.571*	0.565*	0.585*	0.632
Correctly Classified	61.50	59.16*	59.08*	57.88*	58.40*	61.50

*, ** paired-t test with significance at a level of 0.01 and 0.005.

The results shows that proposed method performed *WTS* classification with average F-Score=0.632 and average classification accuracy=61.50%, it was found *BOFW* method outperformed all baselines methods.

A pair-t-test showed that the results of *BOFW* method significantly differed at the level of 0.05 on the classification accuracy of *ERD+RB* method ($D_{nocat.[1..5]}$ *classifiers vs* D_{cat} *test dataset*) and the rest at the level of 0.01.

5 Conclusions

In this paper, we proposed new methodologies to construct domain independent *WTS* classifiers based on function words as features. Experimental results showed that the proposed method provides higher *WTS* classification capability than previous methods. The proposed method also provides a simple way to build the *WTS* classifier hence it reduces the labour required for manually defining a rule dictionary. It also showed that the *BOFW* method provides the more stable *WTS* classification performance regardless of the domain of training dataset and test dataset. Consequently, we accomplished our aim to introduce a simple yet effective way to build a domain independent *WTS* classifier to perform accurate *WTS* classification.

In the future, we are interested in building a non-factoid based QA system by extending the *BOFW* method to develop automated non-factoid *TS* classification of answers describing *"definition"* and *"method"*. We believe that these technologies greatly contribute to developing next generation searching techniques which will improve the information retrieval on the web.

Acknowledgements. This research is supported by Yahoo!Answer. We would like to show our gratitude to NII and Yahoo! for providing QA data.

References

1. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. Machine Learning, 148–156 (1996)
2. Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Technical Report, Stanford University (1998)
3. Radev, D., Fan, W., Qi, H., Wu, H., Grewal, A.: Probabilistic question answering on the web. In: WWW, pp. 408–419 (2002)
4. Tanaka, K., Takiguchi, T., Ariki, Y.: Automatic Why Text Segment Classification and Answer Extraction by Machine Learning. IPSJ Journal 49(6), 57–64 (2008) (Japanese)
5. Higashinaka, R., Isozaki, H.: Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions. TALIP 7, 1–29 (2008)
6. Yin, L.A.: Two-Stage Approach to Retrieving Answers for How-To Questions. In: EACL 2006, pp. 63–70 (2006)
7. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

8. Kwok, C.C.T., Etzioni, O., Weld, D.S.: Scaling Question and Answering to the Web. In: WWW, pp. 150–161 (2002)
9. Lin, J., Katz, B.: Question answering from the web using knowledge annotation and knowledge mining techniques. In: CIKM, pp. 116–123 (2003)
10. Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization, pp. 185–208. MIT Press (1999)
11. Nagy, I., Tanaka, K., Ariki, Y.: Why Text Segment Classification Based on Part of Speech Feature Selection. In: Pfahringer, B., Holmes, G., Hoffmann, A. (eds.) DS 2010. LNCS, vol. 6332, pp. 87–101. Springer, Heidelberg (2010)
12. Matsumoto, Y.: Morphological Analysis System Chasen. IPSJ 41(11), 1208–1214 (2000) (Japanese)
13. Mizuno, J., Akiba, T., Fujii, A., Itou, K.: Non-factoid Question Answering Experiments at NTCIR-6: Towards Answer Type Detection for Realworld Questions. In: The Sixth NTCIR Workshop, pp. 487–492 (2007)
14. Ishioroshi, M., Sato, M., Mori, T.: Answering Any Class of Japanese Non-factoid Question by Using the Web and Example Q&A Pairs from a Social Q&A Website. In: WAIIT, pp. 59–65 (2008)
15. Cortes, C., Vapnik, V.: Support Vector Networks. Mach. Learn. 20(3), 273–297 (1995)
16. Shibusawa, U., Hayashi, T., Onai, R.: Development and Evaluation of a System for Extracting Answers of a ”Why” Type Question from the WEB. IPSJ Journal 48(3), 1512–1523 (2007) (Japanese)
17. Soricut, R., Brill, E.: Automatic Question Answering: Beyond the Factoid. In: HLT/NAACL, pp. 54–64 (2004)
18. Srihari, R., Li, W.: Information Extraction Supported Question Answering. In: TREC, pp. 185–196 (1999)
19. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.A.J.M.: What is not in the Bag of Words for Why-QA? Comput. Linguist. 36(2), 229–245 (2010)
20. Verberne, S., Boves, L., Oostdijk, N.H.J., Coppen, P.A.J.M.: Evaluating Discourse-based Extraction for Why-Question Answering. In: SIGIR, pp. 735–737 (2007)
21. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.: Using Syntactic Information for Improving Why-Question Answering. In: COLING, pp. 953–960 (2008)