Latent Semantic Analysis for Question Classification with Neural Networks

Babak Loni, Seyedeh Halleh Khoshnevis, Pascal Wiggers

Department of Media and Knowledge Engineering, Delft University of Technology
PO Box 5031, 2600 GA Delft, Netherlands

b.loni@student.tudelft.nl
s.khoshnevis@student.tudelft.nl
p.wiggers@tudelft.nl

***Abstract*—An important component of question answering systems is question classification. The task of question classification is to predict the entity type of the answer of a natural language question. Question classification is typically done using machine learning techniques. Most approaches use features based on word unigrams which leads to large feature space. In this work we applied Latent Semantic Analysis (LSA) technique to reduce the large feature space of questions to a much smaller and efficient feature space. We used two different classifiers: Back-Propagation Neural Networks (BPNN) and Support Vector Machines (SVM). We found that applying LSA on question classification can not only make the question classification more time efficient, but it also improves the classification accuracy by removing the redundant features. Furthermore, we discovered that when the original feature space is compact and efficient, its reduced space performs better than a large feature space with a rich set of features. In addition, we found that in the reduced feature space, BPNN performs better than SVMs which are widely used in question classification. Our result on the well known UIUC dataset is competitive with the state-of-the-art in this field, even though we used much smaller feature spaces.**

I. Introduction

Question Answering (QA) provides an alternative for search engines. In many cases the user prefers to get a precise and concise piece of information instead of a list of documents, in response to a natural language question. QA systems aim to do so. Question classification (QC) is a crucial component of QA systems which maps a question to a predefined category that specifies the *entity* type of the expected answer. Our focus is on *fact-based* questions for which the answer is one or a few words. For example for the question "Who was President of Costa Rica in 1994?" the role of question classification is to map this question to the category "human" since its answer is a named entity of type "human".

Determining the class of a question not only specifies the search strategy, but can also reduced the search space to a much smaller space since the answering system only needs to search for those entities which match with question class [1].

Question classification is typically done using machine learning approaches. A problem of learning-based QC systems is the high dimensionality of the feature space which typically is due to n-grams over all words in vocabulary. In this work we applied LSA [2], a successful feature reduction technique, for the QC task. LSA has been successfully applied on text and document classification [3], [4], [5]. We used LSA to reduce the high dimensional feature space of questions to a smaller dimension. We extracted different sets of features and tested our system with different combinations of these feature sets with two different classifiers in both original space and reduced space. The results show that LSA on question classification leads to a more reliable classifier with a more efficient set of features and better accuracy.

This paper is organized as follows: in section II we introduce the classifiers we used in this work. We explain the features that we extract from the questions in section III. Section IV describes our method to reduce feature spaces with LSA. Our experiments and results are described in section V. We discuss related work on this task in section VI and finally we draw a conclusion in section VII.

II. Classifiers

Question classification has been studied by using different type of classifiers. Most of the successful studies on this task uses support vector machines [6], [7], [8], [9]. SVMs are very successful on high dimensional data since they are timely efficient especially when the feature vectors are sparse, but they still suffer from the redundant features. Question classification has also been done by Maximum Entropy models [7], [10], Sparse Network of Winnows (SNOW) [11] and language modeling [12].

In this work we adopted SVMs as well as back-propagation neural networks. Training a neural network with high dimensional vectors such as questions, demands very large networks which make them very costly to train. However, by applying LSA feature reduction technique, we can train smaller yet efficient networks in a reasonable time which makes them suitable to be used for question classification. To our knowledge this is the first work which uses neural networks for question classification. In this section we briefly describe the classifiers we used.

A. Support Vector Machines

SVM is a linear discriminant model which tries to find a hyperplane with maximum margin for separating the classes.

They are fast classifiers for high dimensional data [13]. To be able to linearly separate data, the feature space usually is mapped to a higher dimensional space. The mapping is done with a so-called *kernel function*. The most widely used kernel in question classification is the linear kernel. In this work we used linear kernel since it showed better performance compare to other type of kernels. We adopted LIBSVM [14], a library for support vector machines, to implement our experiments.

B. Back-Propagation Neural Networks

Back-propagation neural networks are multi-layer feed forward neural networks which are trained with the back-propagation learning rule [15]. They consist of an input layer, an output layer and one or more hidden layers. Each neuron has a forward connection to all neurons in the subsequent layer and the importance of connections are reflected by weight parameters. The input of each neuron is weighted sum of its input signals and the output is calculated as a function of input signals and an optional threshold parameter.

1) Training the BBNN: Consider we are given a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ such that $\mathbf{x}_i = (x_{i1}, ..., x_{id})$ is a d-dimensional input vector and y_i is its corresponding class label which takes one of the values from the set of labels $C = \{c_1, ..., c_m\}$. To build a network based on our training set, the number of input neurons should be equal to d, the number of features, and the number of output neurons should be set to m, the number of classes. The number of hidden layers and neurons in each layer should be learned or specified in advance. Figure 1 depicts the structure of a network with one hidden layer in which the number of hidden neurons are equal to the number of output neurons. For an input vector $\mathbf{x}_i = (x_{i1}, ..., x_{id})$, the input of each neuron in input layer is fed with exactly one feature of $\mathbf{x}_i$. The network generates m outputs. The class label of $\mathbf{x}_i$ is determined by a max rule:

$$c = \arg\max_{k=1}^{m} Y_k(\mathbf{x}_i), \tag{1}$$

where $Y_k(\mathbf{x}_i)$ is the output value generated by neuron k in output layer and c is the class number.

According to the defined notations, for a given input vector $\mathbf{x}_i$, the input of a hidden node j is defined as the following weighted sum:

$$\phi_j = \sum_{k=1}^{d} w_{kj} x_{ik}, \tag{2}$$

where w_{kj} indicates the weight in the link between input neuron k and hidden neuron j. The output of a hidden unit is calculated as follow:

$$\psi_j = f(\phi_j + \theta_j), \tag{3}$$

such that θ_j is the threshold parameter of hidden unit j and f is a non-linear transformation which is referred as the activation function. Our experimental results show that the sigmoid activation function performs better than other types of functions. The sigmoid function is defined as:

$$f(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})} \tag{4}$$

The input and output of the output layer are also calculated using (2) and (3). The back-propagation learning rule, initializes weight and threshold parameters randomly and iteratively updates these, using a gradient descent method such that the error on the training set converges to a small value. We used Neroph[1], a Java framework for neural networks, to implement our classifier.

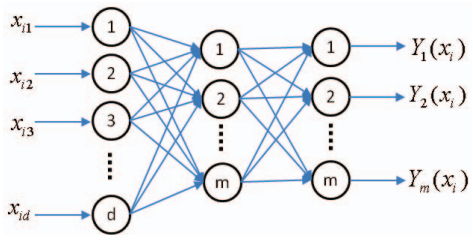

Fig. 1. The structure of network we used in this work.

III. Features

In question classification, a question is represented using a *vector space model*, i.e., the question is a vector which is described by the words inside it. Therefore, a question $\mathbf{x}$ is represented by vector $\mathbf{x} = (x_1, ..., x_d)$ in which x_i is the frequency of term i in $\mathbf{x}$ and d is the total number of terms. This representation is also referred as *bag-of-words* or *unigrams* which is the simplest type of features that can be extracted from a question and is the most widely used feature space in document classification [16]

However, more advanced features can be extracted from semantic and syntactical structure of questions and expand the features space [11], [7]. We extracted 6 more features namely: *bigrams*, *word-shapes*, *wh-words*, *head-words*, *related-words* and *hypernynms*. These features are added to the bag-of-words, i.e., the feature is viewed as a new term and its value set to one if the feature exists in the question.

A. Bigrams

If any two consecutive words in a question is considered as a feature, the resulting feature space is called *bigrams*. It is an special case of *n-grams* in which any n-consecutive words are considered as a single feature. Bigram features however, are very high dimensional since all two consecutive terms in our dataset should be considered as features, of which most are redundant and do not show up in the data. We found that considering only the first two words of a question as bigram features, performs as good as all bigrams while the size of feature space is much smaller. For example consider the question "How many people in the world speak French?". The

[1] http://neuroph.sourceforge.net/

only meaning bigram in this question is "How-many" while the rest is not useful. That is also true in the questions in which the wh-word is one word, because the combination of wh-word and the immediate word next to it, is an informative feature in most cases. For example most of the questions which starts with "what is/are" are asking for a definition. In the rest of the paper, we call our limited bigrams feature space as *limited bigrams*.

B. Wh-words

We considered wh-word of a question as a separate lexical feature. Similar to [7] we adopted 8 type of wh-words namely *what*, *which*, *when*, *where*, *who*, *how*, *why* and *rest*. For example the wh-word of the question "When did CNN begin broadcasting?" is *when*.

C. Word Shapes

A small yet effective feature is the word shape. It describes the appearance of single words in a question. Huang et al. [7], introduce 5 categories for word shapes: *all digit*, *lower case*, *upper case*, *mixed* and *other*. For example for the question "When did CNN begin broadcasting?", the word "When" has a *mixed* word shape, "CNN" has *upper case* shape and "begin" has a *lower case* shape. We identify the shape of all the words in a question and add them to the feature vector.

D. Headwords

A head word is usually defined as the most informative word in a question or a word that specifies the object that the question seeks [7]. Identifying the headword correctly can significantly improve the classification accuracy since it is the most informative word in the question. For example for the question "What is the oldest city in Canada?" the headword is "city". The word "city" in this question can highly contribute the classifier to classify this question as "location".

A question's headword is extracted based on the syntactical structure of the question. Since headword extraction is not the focus of this work, we suggest the readers to read [17] for a detail explanation of headword extraction techniques.

E. Hypernyms

WordNet [18] is a lexical database of English words which provides a lexical hierarchy that associates a word with higher level semantic concepts namely *hypernyms*. For example a hypernym of the word "city" is "municipality" of which the hypernym is "urban area" and so on. As hypernyms allow one to abstract over specific words, they can be useful features for question classification.

We used the MIT Java interface to WordNet [19], to extract the hypernyms of a word. The hypernyms can be extracted for any word in the question which has an entry in the WordNet database. However since adding the hypernyms of all the words can introduce noisy information, we only add the hypernyms of the question's headword to the feature vector. Furthermore, we experimentally found that the best results are obtained when the maximum dept of hypernyms are set to 6, i.e., in the hypernyms hierarchy, we go up maximally to 6 levels. Furthermore, a word may have different senses, each of which has different hierarchy. For example the word "capital" can either be interpreted as "large alphabetic character" or "a seat of government". Each sense has its own hypernyms and the true sense should be identified based on the sentence it appears in. To identify the true sense of a word in a question we adopted Lesk's *word sense disambiguation algorithm* [20] which predicts the true sense of a word based on the context it appears.

Now the take the example "What is the capital of the Netherlands?". The headword of this question is "capital" and the true sense is sense 3 in WordNet. This sense has the following hypernyms: {capital, seat, center, area, region, location, object, physical-entity, entity}. The first 6 words are considered as features and are added to the feature vector.

F. Related Words

Another semantic feature that we implemented is related words which is based on the idea of Li et al. [11]. They defined groups of words, each represented by a category name. If a word in the question exists in one or more groups, its corresponding categories will be added to the feature vector. For example if any of the words {birthday, birthdate, day, decade, hour, week, month, year} exists in a question, then its category name, *date*, will be added to the feature vector.

To expand the feature vector with related words, still we can choose to only consider the head word or all words in question. Our experimental results show that considering the whole question gives better results.

IV. Feature Reduction with LSA

Latent semantic analysis [2] is a feature reduction technique which maps the features space to a reduced space using *singular value decomposition* (SVD). It is widely used in text classification [3], [4], [5].

To apply SVD to question classification, we define the feature-by-question matrix $\mathbf{Q}$ in which the rows represent the features and the columns represent questions. That is, if our feature space has d dimensions and the total number of training samples is n, then $\mathbf{Q}$ would be a $d \times n$ matrix in which $\mathbf{Q}_{i,j}$ represents the frequency (weight) of feature f_i in question $\mathbf{x}_j$. SVD decomposes $\mathbf{Q}$ into tree matrices: $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices whose columns are eigenvectors of $\mathbf{Q}\mathbf{Q}^T$ and $\mathbf{Q}^T\mathbf{Q}$ respectively and $\mathbf{\Sigma}$ is a diagonal matrix containing the eigenvalues of $\mathbf{Q}\mathbf{Q}^T$ in the diagonal which are sorted in descending order. To reduce the feature space to k dimensions, we define matrix $\mathbf{U}_k$ to be a $d \times k$ matrix containing the first k column of $\mathbf{U}$. We now defined the reduced matrix as follows:

$$\mathbf{R} = \mathbf{Q}^T\mathbf{U}_k \tag{5}$$

where $\mathbf{R}$ is the $n \times k$ reduced matrix, in which each row corresponds to a question which is described by k features. This technique is very similar to *principle component analysis*.

The reduced space is called *latent semantic space* and matrix $\mathbf{U}$ is used to transform a vector to this space.

Once we train our classifiers with the reduced questions, for a given independent question $\mathbf{x}$, it first transforms to the reduced space as follows:

$$\hat{\mathbf{x}} = \mathbf{x}^T \mathbf{U}_k \tag{6}$$

where $\hat{\mathbf{x}}$ is a $1 \times k$ vector in the reduced space. This vector then is fed to our classifier, and the output is generated.

V. Experiments

A. The Dataset

The set of question categories (classes) which questions are mapped to is referred to *question taxonomy*. Most of the recent work on question classification use the taxonomy proposed by [21] since the authors published a valuable set of 6000 labeled questions. This dataset consists of two separate set of 5500 and 500 questions respectively in which the first is used as training set and the second is used as an independent test set. This dataset[2] which was first published by the University of Illinois Urbana-Champaign (UIUC) usually referred as the UIUC dataset.

We also used this dataset to evaluate our work. Table I lists the UNUC taxonomy. This taxonomy consists of 6 coarse-grained classes and 50 fine-grained classes.

B. Setup Experiment

We performed our experiment in two different scenarios: either to apply the LSA feature reduction technique or not using it. In the first scenario, the system first extracts different types of features and then combine them to make a richer feature space. After that, we apply LSA on the combined feature space to reduce the features space. The reduced space is used to train and test our classifier. Figure 2 illustrates the architecture of our system when we use the LSA technique. The second scenario is similar to the first, but lacks the feature reduction step, that is the classifier is trained and tested with the original features. Since training a neural network in the second scenario is quite time-consuming we only applied the second scenario to the SVM classifier.

[2]http://cogcomp.cs.illinois.edu/Data/QA/QC/

TABLE I
THE COARSE AND FINE GRAINED QUESTION CLASSES

Coarse	Fine
ABBR	abbreviation, expansion
DESC	definition, description, manner, reason
ENTY	animal, body, color, creation, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUM	description, group, individual, title
LOC	city, country, mountain, other, state
NUM	code, count, date, distance, money, order, other, percent, percent, period, speed, temperature, size, weight

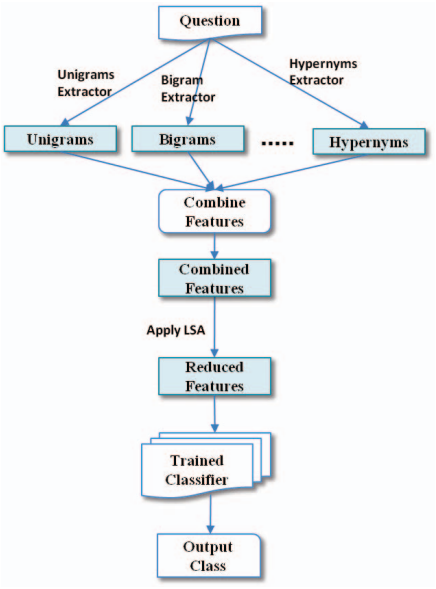

Fig. 2. The overall architecture of our question classifier system.

1) Structure of Classifiers: We tested our SVM classifier with 4 types of kernel functions: linear, polynomial, radial basis and sigmoid. In all scenarios the linear kernel has significantly better performance. Furthermore, the penalty parameter of the error function [13] is set to its default value of 1.

Our BPNN classifier uses one hidden layer in which the number of hidden units are set to the number of classes (see figure 1). The reasons of choosing this architecture are both accuracy and efficiency. In the tested scenarios, having more than 1 hidden layers do not necessarily improves the performance while the network takes more time to be trained. The maximum number of iterations in the gradient descend method is set to 500 and the learning rate is set to 0.7 since with this combination of parameters in most cases the error converges to a fixed value.

C. Comparison of Feature Sets

We extracted 7 types of lexical, syntactical and semantic feature sets. Combining all these feature sets together is not necessarily the best option. We tested our SVM classifier with different combinations of features on both coarse and fine grained classes in the original feature space. Table II lists our result on the UIUC dataset using the SVM classifier. The accuracy is defined as the number of correctly classified samples divided by total number of tested samples.

The first half of table II lists combinations of features including unigrams, and the second half, lists those feature sets which do not include unigrams. As table II shows, unigram and bigram feature sets increase the size of feature space significantly. Furthermore, the best results for fine-grained classes are obtained when unigrams are used, while for the coarse grained classes unigrams are not an impressive feature set.

TABLE II
THE ACCURACY OF SVM CLASSIFIER ON UIUC DATASET BASED ON DIFFERENT COMBINATION OF FEATURES.
THE ABBREVIATION OF FEATURES ARE: **U**: UNIGRAMS, **B**: BIGRAMS, **LB**: LIMITED-BIGRAM, **WH**: WH-WORD, **WS**: WORD-SHAPES, **H**: HEADWORD, **HY**: HYPERNYMS, **R**: RELATED-WORDS

no.	Features	Dimensions	Accuracy	
			Coarse	Fine
1	U	9775	88.2	80.2
2	U+WS+H	9780	88.8	84.2
3	U+WS+H+R	9858	91.0	89.8
4	U+WS+H+R+WH	9858	91.2	89.4
5	U+WS+H+R+HY	13668	90.6	90.0
6	U+WS+H+R+LB	10876	91.4	89.4
7	U+WS+H+R+HY+LB	14708	93.0	**90.4**
8	WH+WS+H	1977	88.6	77.0
9	WH+WS+H+R	2055	89.6	87.8
10	WH+WS+H+R+LB	3072	92.4	88.2
11	WH+WS+H+R+B	32776	**93.4**	89.4

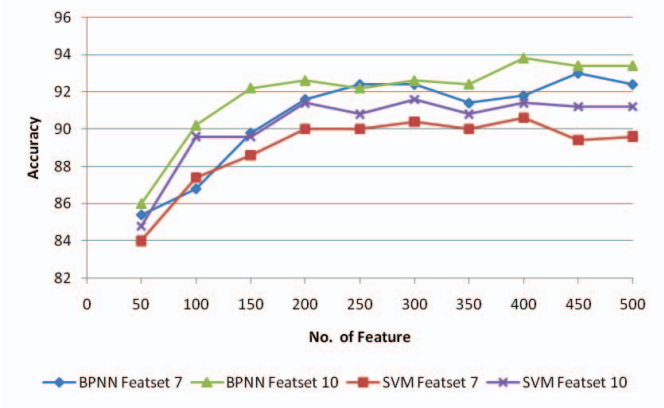

Fig. 3. Comparison of SVM and BPNN classifiers on the reduced space on two different feature spaces for coarse grained classes.

D. Comparison in the Reduced Space

The next step of our experiment is to test our classifiers in the reduced feature space. We first want to investigate the behavior of different feature sets in the reduced space and then to find out what is the best size for the reduced space. We tested the accuracy of different feature sets on the reduced space with both SVM and BPNN classifiers. Figure 3 compares the accuracy of SVM and BPNN classifiers on feature sets number 7 and 10 in table II for the coarse grained classes and figure 4 compares the accuracy of these two classifiers on the fine grained classes based on the feature set number 10. The horizontal axis in the figures are the number of features that results from the LSA reduction and the vertical axis are classification accuracies. We choose feature sets 7 and 10 for comparison since both have a good performance in the original space while the first has high dimensions and the second has lower dimensions in the original space.

As the figures reveals, BPNN performs better on the reduced space for coarse grained classes while for the fine grained

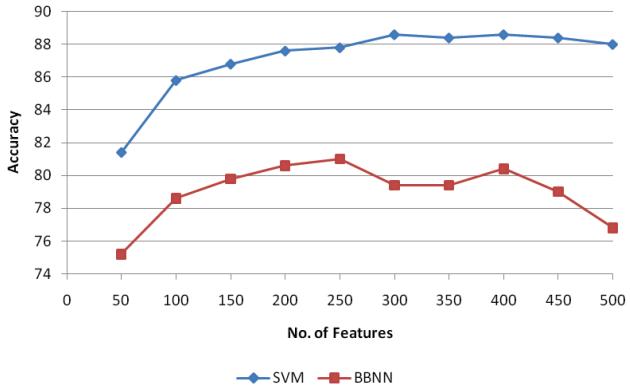

Fig. 4. Comparison of SVM and BPNN classifiers on the reduced space for fine grained classes.

TABLE III
THE ACCURACY OF SVM AND BPNN CLASSIFIER ON THE 400 DIMENSIONAL REDUCED SPACE FOR THE COARSE GRAINED CLASSES COMPARE TO THE ACCURACY OF SVM IN THE ORIGINAL SPACE.

Features	Original Space	Reduced Space	
	SVM	SVM	BPNN
U+WS+H+R+HY+LB	93.0	90.6	93.4
WH+WS+H	88.6	85.6	88.8
WH+WS+H+R	89.6	88.4	90.2
WH+WS+H+R+LB	92.4	91.4	**93.8**

classes SVM performs better in the reduced space. Furthermore, for the coarse grained classes, BPNN achieves higher accuracy in the reduced space than the SVM in the original space, while SVM performs worse compare to the original space. The most interesting result from figure 3, is that feature set 10 has higher accuracy than feature set 7 in the reduced space even though in the original space feature set 7 has a higher accuracy. The reason may be that feature set 10 has lower dimensions and describes the samples in a more compact space and therefore looses less information when it is reduced to a lower dimensional space.

The best accuracy which is obtained for coarse grained classes in the reduced space is 93.8% with 400 features using BPNN classifier. This result is not only better than the highest accuracy of SVM in the original space (93.4%), but also uses only 400 features which is much less than the dimensionality of the original space which is 32776. Table III compares accuracies of more feature sets with SVM and BPNN classifiers in the 400 dimensional reduced space for the coarse grained classes. As this table reveals, in all cases BPNN in the reduced space has a higher accuracy than the SVM classifier in the original and the reduced space.

VI. RELATED WORK

The task of question classification came into the focus when Text REtrieval Conference (TREC) began a QA track in 1999[22]. The first successful learning-based question clas-

sifier was introduced by Li et al. [21] when they first used syntactic and semantic features for question classification. They later improve their work by introducing richer semantic features. They uses SNOW architecture in their work and obtained an accuracy of 89.3% on the fine grain classes of UIUC dataset while their feature space is more than 200,000 dimensions. Later, Huang et al. [7] extracts richer set of syntactic and semantic features in rather lower dimensions and reach an accuracy of 89.2% on the fine and 93.4% on the coarse grained classes of UIUC dataset using SVM classifier. They also reported almost a similar accuracy using maximum entropy models. More recently, Silva et al. [8] developed a hybrid approach for question classification which uses both hand crafted rules and SVMs. They reported an accuracy of 90.8% on the fine and 95.0% on the coarse grained classes which is the highest accuracy reported on this dataset. In the most recent study, Loni et al. [9] combined different lexical, syntactical and semantic feature sets by a weighted approach. They obtained accuracies of 89.0% and 93.6% on the fine and coarse grained classes of UIUC dataset, respectively. A comprehensive overview of the state-of-the-art methods on question classification can be found in [23].

VII. Conclusions and Future Directions

In this work we applied the LSA reduction technique to reduce the dimensionality of the feature space in learning-based question classification. While extracting rich syntactical and semantic features can increase the classification accuracy, they can both introduce noisy information and hurt the efficiency of classifiers. LSA tries to map the feature space to a smaller and more efficient space while it keeps the accuracy high.

Our experimental results show that in the reduced space SVMs are not as good as the original space but BPNN performs better specially when the number of classes are few. Furthermore, if we succeed to represent the questions in lower dimensions, then its reduced space performs better than a feature space with high dimensions. We compared the classification accuracy of different feature sets and found that the high dimensional feature spaces such as unigram ans bigram can be replaced by lower dimensional features such as wh-words, headwords and limited bigrams while the accuracy remains high.

Different extensions to this work can be done in future studies. Richer feature sets from syntax and semantic of questions can be extracted in future works to enhance the classification accuracy. Furthermore, by augmenting our latent semantic space with semantic information from third party sources, the features can be tuned to more informative features. In this work we obtained better performance by combining different feature sets. Combining different *classifiers* can also be done in future studies to see whether they can improve classification accuracy or not.

References

[1] Z. Huang, M. Thint, and A. Celikyilmaz, "Investigation of question classifier in question answering," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, ser. (EMNLP '09), 2009, pp. 543–550.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] B. Yu, Z.-b. Xu, and C.-h. Li, "Latent semantic analysis for text categorization using neural network," *Know.-Based Syst.*, vol. 21, pp. 900–904, December 2008.

[4] S. L. Y. Lam and D. L. Lee, "Feature reduction for neural network based text categorization," in *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications*, ser. DASFAA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 195–202.

[5] S. Zelikovitz and H. Hirsh, "Using lsi for text classification in the presence of background text," in *Proceedings of the tenth international conference on Information and knowledge management*, ser. CIKM '01. New York, NY, USA: ACM, 2001, pp. 113–118.

[6] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 26–32.

[7] Z. Huang, M. Thint, and Z. Qin, "Question classification using head words and their hypernyms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. (EMNLP '08), 2008, pp. 927–936.

[8] J. a. Silva, L. Coheur, A. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, Feb. 2011.

[9] B. Loni, G. van Tulder, P. Wiggers, D. M. J. Tax, and M. Loog, "Question classification by weighted combination of lexical, syntactical and semantic features," in *Proceedings of the 14th international conference of Text, Speech and Dialogue*, 2011.

[10] P. Blunsom, K. Kocik, and J. R. Curran, "Question classification with log-linear models," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '06. New York, NY, USA: ACM, 2006, pp. 615–616.

[11] X. Li and D. Roth, "Learning question classifiers: The role of semantic information," in *In Proc. International Conference on Computational Linguistics (COLING*, 2004, pp. 556–562.

[12] A. Merkel and D. Klakow, "Improved methods of language model based question classification," in *In Proceedings of Interspeech Conference*, 2007.

[13] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," Jul. 2008.

[14] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[15] Y. H. Hu, *Handbook of Neural Network Signal Processing*, 1st ed., J.-N. Hwang and J.-N. Hwang, Eds. Boca Raton, FL, USA: CRC Press, Inc., 2000.

[16] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, pp. 537–546, 1998.

[17] B. Loni, "Enhanced question classification with optimal combination of features," Master's thesis, August 2011.

[18] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[19] M. A. Finlayson, "MIT Java WordNet Interface series 2," 2008. [Online]. Available: http://projects.csail.mit.edu/jwi/

[20] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation*, 1986, pp. 24–26.

[21] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics*, ser. COLING '02. Association for Computational Linguistics, 2002, pp. 1–7.

[22] E. M. Voorhees and D. Harman, "Overview of the eighth text retrieval conference (trec-8)," 2000, pp. 1–24.

[23] B. Loni, "A survey of state-of-the-art methods on question classification," Delft University of Technology, Tech. Rep., June 2011.