Question Classification by Weighted Combination of Lexical, Syntactic and Semantic Features

Babak Loni, Gijs van Tulder, Pascal Wiggers, David M.J. Tax, and Marco Loog

Delft University of Technology, Pattern Recognition Laboratory,
P.O. Box 5031, 2600 GA Delft, The Netherlands
{b.loni,G.vanTulder}@student.tudelft.nl,
{P.Wiggers,D.M.J.Tax,M.Loog}@tudelft.nl

Abstract. We developed a learning-based question classifier for question answering systems. A question classifier tries to predict the entity type of the possible answers to a given question written in natural language. We extracted several lexical, syntactic and semantic features and examined their usefulness for question classification. Furthermore we developed a weighting approach to combine features based on their importance. Our result on the well-known TREC questions dataset is competitive with the state-of-the-art on this task.

1 Introduction

One of the most crucial tasks in Question Answering (QA) systems is question classification. A question classifier predicts the *entity type* of a possible (factual) answer for a given question. For example, if the system is asked "What is the capital of the Netherlands?", the question classifier should assign to this question the label *city*, since the expected answer is a named entity of type *city*.

Determining the class of a question is quite useful for the process of answering the question. Knowing that the question is of a particular type, the search space for possible answers can be narrowed down to a much smaller space. Furthermore, the question class can be used to rank the candidate answers [5,12].

In this work, we developed a feature-driven learning-based question classifier that is competitive with state-of-the-art question classification approaches. We extracted known and new lexical, syntactic and semantic features and compared the classification accuracies that can be obtained with these sets. Furthermore, we investigated whether combining feature sets can improve classification accuracy. For this we introduce a weighted combination approach that takes into account the importance of the features.

This paper is organized as follows. We start with a discussion of related work in section 2. In section 3 we discuss our motivation for choosing support vector machines (SVMs) as our classifier. In section 4 we explain the features that we extracted and their individual classification accuracies. We introduce our approach to combine features in section 5. We end with a conclusion.

2 Related Work

Different approaches to question classification have been proposed. Some early studies build question classifiers based on matching with hand-crafted rules [15]. However,

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 243–250, 2011.

these approaches do not generalize well to new domains and do not scale easily. Most recent studies build question classifiers based on machine learning approaches [6,16,9,10] that use features extracted from the question.

The accuracy of most of the studies, including this work, is usually measured on a well-known *taxonomy* of question classes proposed by Li and Roth [9]. This taxonomy has two layers consisting of 6 coarse grained and 50 fine grained classes (Table 1). A dataset of almost 6000 labeled question has been created based on this taxonomy[1] [9]. This dataset which is usually referred to as the TREC (Text REtrieval Conference) dataset, is divided in a training set of 5500 questions and a test set of 500 questions. The accuracy of a question classifier is defined as the number of correctly classified questions divided by total number of questions.

Table 1. 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

Li and Roth [9] obtained an accuracy of 84.0% on the fine grained classes using a SNOW (Sparse Network of Winnows) architecture. Using different semantic features, [10] obtained an accuracy of 89.3% for the fine grained classed. [6] reached accuracies of 89.0% using a Maximum Entropy model and 89.2% using SVMs with linear kernels on the fined grained classes, while they obtained an accuracy of 93.6% on the coarse grained classes. Zhang et. al. [16] proposed a syntactic tree kernel for SVM-based question classification. They obtained an accuracy of 90.0% on the coarse grained classes. Pen. et. al. [11] reach an accuracy of 94.0% on the course grained classes with a semantic tree kernel for SVM classifiers. [13] combined rule-based and learning based approaches. They used matched rules as features for a SVM classifier. They reach an accuracy of 90.0% on the fine grained and an accuracy of 94.2% on the coarse grained classes. To our knowledge this is the highest accuracy achieved on the TREC dataset.

3 Choosing the Classifier

The choice of classifier is an important decision in our system. Since in the question classification problem the questions are represented in a very high dimensional feature space, we decided to choose Support Vector Machines [14] as our classifier. SVMs are

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

shown to have good performance on high dimensional data and generally outperform other classifiers, such as Nearest Neighbor, Naive Bayes, Decision Trees, SNoW and Maximum Entropy on question classification [16,6,5]. We decided to rely on simple linear kernels for the SVMs together with rich features, rather than on other, more complex kernels. All systems were implemented with LIBSVM [1], a library for support vector machines.

4 Features in Question Classification

We used three different types of features: lexical, syntactic and semantic features. We introduce each type of feature and show classification results than can be achieved for every feature type.

4.1 Lexical Features

Lexical features of a question are generally based on the *context words* of the question, i.e., the words that appear in a question. In the *unigram* or *bag-of-words* approach each word in the vocabulary is treated as a feature. For each question the value of every word feature is set to the frequency count of that word in the question. This can lead to a high dimensional feature space, but that can be dealt with by using sparse representations that only store non-zero entries. Unigram features are a special case of n-gram features, that treat every sequence of n consecutive words in the question as a feature.

To obtain insight in the influence of lexical features on question classification, we trained our classifier with different types of lexical features. The classification accuracy is listed in Table 2. It shows that, most likely due to data sparseness, unigrams are better features than bigrams.

In recent studies Huang et. al [5,6] considered question *wh-words* as a separate feature. They selected 8 types of wh-words, namely *what*, *which*, *when*, *where*, *who*, *how*, *why* and *rest*. For example the wh-word feature of the question "What is the longest river in the world?" is *what*. We extracted wh-word features from a question with the same approach as [5].

The *word shape* is a word-level feature that refers to the type of characters used in a word. This can be useful to identify for example numerical values and names. Inspired by [6], we introduced four categories for word shapes: *all digit*, *lower case*, *upper case* and *other*. Not surprisingly, Word shapes alone is not a good feature set for question classification (Table 2), but, as will be shown in the next section, combined with other features they can improve classification accuracy.

Table 2. The accuracy of SVM classifier based on different lexical features for coarse and fine grained classes

Feature Space	unigram	bigram	wh-words	word-shapes
Coarse	88.2	86.8	45.6	35.5
Fine	80.4	75.2	46.8	30.8

4.2 Syntactic Features

A different class of features can be extracted from the syntactic structure of the question. We extracted two syntactical features namely *Tagged Unigrams* and *Head Words*.

Tagged Unigrams: We introduce a new feature namely *tagged unigram* which are unigrams augmented with their part-of-speech (POS) tags. They allow the model to differentiate between words that have the same lexical form (e.g. the verb 'to record' vs. the noun 'record'). We used the Stanford implementation of a Maximum Entropy POS-tagger [7] to tag the questions. Following is a sample question from the TREC dataset augmented with its POS-tags:

Who_WP was_VBD The_DT Pride_NNP of_IN the_DT Yankees_NNPS ?_

Similar to the bag-of-words approach, the bags are now made up of augmented words. Most likely due to data sparseness, tagged unigrams do not necessarily have a better accuracy than unigrams (Table 3). but when combined with other features, they sometimes show better performance compared to unigrams.

Head Words: For question classification the head word is usually defined as the "single word that specifies the object that the question seeks" [6]. For example for the question "What is the oldest city in the United States ?", the head word is "city". The head word is usually the most informative word in the question and correctly identifying it can significantly improve the classification accuracy.

Extracting the head word of a question is quite a challenging problem. Similar to [6,13], we extracted the head word based on the syntactical structure of the question. To obtain the head word, we first parse the question using the the Stanford parser [7] and then extract the head word based on the parse found. For head word propagation we adapted the rules defined by [2] — for the propagation of syntactic heads — to prefer noun heads over verb heads, as for question answering the subjects and objects of a sentence are usually more informative than its verbs.

Table 3 lists the accuracy of two syntactic features that we used in this work.

Table 3. The accuracy of SVM classifier based on different syntactic features for coarse and fine grained classes

Feature Space	tagged unigram	Headwords
Coarse	87.4	62.2
Fine	80.6	40.6

4.3 Semantic Features

In addition to lexical and syntactic features, we extracted two features related to the semantics of the question called *related words group* and *Hypernyms*.

Related Words Group: Li and Roth [10] 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.

Hypernyms: For a given word, a hypernym is a word with a more general meaning. For example a hypernym of the word "city" is "municipality". As hypernyms allow one to abstract over specific words, they may be useful features for question classification [5]. We used WordNet [3] together with the MIT Java Wordnet Interface package [4] to extract hypernyms.

However, extracting hypernyms is not straightforward. There are four challenges that should be addressed to obtain hypernym features: 1) For which word(s) in the question should we find hypernyms? 2) For the candidate word(s), which part-of-speech should be considered? 3) The candidate word(s) augmented with their part-of-speech may have different senses in WordNet. Which sense is the sense that is used in the given question? and 4) How far should we go up through the hypernym hierarchy to obtain the optimal set of hypernyms?

To address the first issue, we choose the head word as the candidate word, since its the most informative word in a question. We found that considering (also) other words in a question can introduce noisy information in feature vector and leads to lower accuracy.

For the second issue we used the POS tags extracted for the syntactic features. To tackle the third issue we adopted Lesk's Word Sense Disambiguation (WSD) algorithm to find the right sense of the candidate word. Lesk's WSD algorithm [8] is a dictionary-based method for resolving the true sense of a word in a sentence. It looks at the descriptions of different senses of the candidate word and chooses the sense in which the description has maximum similarity with the description of the context words in the sentence.

Finally, to address the 4th challenge, i.e. the depth of hypernyms to use, we relied on the experiments of [5], in which the value of six is considered as the maximum depth of hypernyms. Table 4 lists the accuracies of the semantic features for question classification. The interesting point about the results in Table 4 is that the "Related words group" features alone have better performance than lexical features. This shows the importance of semantic features in question classification. Hypernyms did not result in a good classification accuracy. The reason may lie in the complicated sequence of tasks needed to extract the hypernyms; an incorrect decision in any task can increase the noise in the feature vector.

Table 4. The accuracy of SVM classifier based on different semantic features for coarse and fine grained classes

Feature Space	related word groups	Head hypernyms
Coarse	85.2	66.6
Fine	79.8	41.6

5 Combining Features

The three feature sets we described each take a different perspective on the question. We explored whether combining different feature sets will improve the classification accuracy. Unlike related work in which the augmented features are blindly added to the feature vector, we suggest a weighted concatenation of the various feature sets:

$$f = (w_1 f_1^{\mathrm{T}}, \ldots, w_m f_m^{\mathrm{T}})^{\mathrm{T}} \tag{1}$$

where f_i is the i^{th} feature set, w_i is its weight, m is the number of feature sets that are extracted and f is the final feature set. In total we implemented 9 types of different features, i.e, $m = 9$. If $w_i = 0$ it means that the i^{th} feature set will not be added to the final feature set. Table 5 lists the classification accuracies based on different combinations of features with equal weights (1.0) on the standard TREC test set.

Table 5. The accuracy of SVM classifier based on different combinations of feature sets, with equal weights, on coarse and fine grained classes

No.	Feature Set	Coarse	Fine
1	unigram	88.2	80.4
2	unigram + wh-words	88.2	80.4
3	unigram + head words	89.0	84.0
4	unigram + hypernyms	90.2	84.6
5	unigram + related words group	90.0	85.2
6	unigram + related words group + word shapes	89.8	86.2
7	(6) + tagged unigram	90.6	86.2
8	(6) + bigram	92.0	86.6
9	(6) + head words	90.8	86.4
10	(6) + head words + hypernyms	91.4	88.0
11	(6) + head words + hypernyms + bigram	93.2	88.0

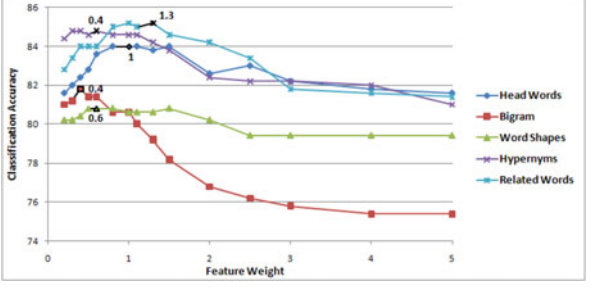

Fig. 1. Classification accuracy of unigram features combined with an other feature set as a function of the combination weight

Table 6. Confusion matrix showing the classifications of the TREC-500 for the coarse categories

		Predicted labels					
		ABBR:*	DESC:*	ENTY:*	HUM:*	LOC:*	NUM:*
True	ABBR:*	9					
	DESC:*		134	2		1	1
	ENTY:*		10	83	1		
	HUM:*		1	1	63		
	LOC:*		1	9		71	
	NUM:*		3	2			108

The best classification accuracy is obtained with the combination of the following six feature sets: unigram, related words group, word shapes, head words, hypernyms and bigrams. To optimize the weight values in equation 1, we would need an exhaustive search of all possible weight assignments. As this is time-consuming, we chose a greedy approach instead. For each feature set we searched for the optimal weight when it was combined with the unigram features only. Figure 1 illustrates the classification accuracy of different features as a function of their weight. The best weight values, which are specified by a label in Figure 1, are used as weight values when combining all feature sets. This resulted in an accuracy of 89.0% on the fine grained classes and 93.6% on the coarse grained classes. We found that some questions are easier to classify than others. While the system performed well for the categories ABBR, DESC, HUM and NUM it made many more errors for the ENTY and LOC categories (Table 6).

6 Conclusion

We developed a learning-based, feature driven question classifier which reaches an accuracy of 89.0% on the fine grained and 93.6% on the coarse grained classes of the TREC dataset, by weighted combination of different features. We succeeded to improve the classification accuracy by almost 9% by adding different features to the basic unigram (bag-of-word) features.

We introduced the concept of a weighted combination of features on question data. Adopting the weights is an important issue when the features are combined. We could further improve the accuracy of classifier by approximating the weights to their optimal combination.

References

1. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001) Software, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
2. Collins, M.: Head-Driven Statistical Models for natural Language Parsing. PhD thesis, University of Pennsylvania (1999)
3. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
4. Finlayson, M.A.: MIT Java WordNet Interface series 2 (2008)

5. Huang, Z., Thint, M., Celikyilmaz, A.: Investigation of question classifier in question answering. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), pp. 543–550 (2009)
6. Huang, Z., Thint, M., Qin, Z.: Question classification using head words and their hypernyms. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 927–936 (2008)
7. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceeding of the 41st Annual Meeting for Computational Linguistics (2003)
8. Lesk, M.: 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, pp. 24–26 (1986)
9. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational linguistics, pp. 1–7. Association for Computational Linguistics (2002)
10. Li, X., Roth, D.: Learning question classifiers: The role of semantic information. In: Proc. International Conference on Computational Linguistics (COLING), pp. 556–562 (2004)
11. Pan, Y., Tang, Y., Lin, L., Luo, Y.: Question classification with semantic tree kernel. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, pp. 837–838. ACM, New York (2008)
12. Quarteroni, S., Manandhar, S.: Designing an interactive open-domain question answering system. Nat. Lang. Eng. 15, 73–95 (2009)
13. Silva, J., Coheur, L., Mendes, A., Wichert, A.: From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review 35(2), 137–154 (2011)
14. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York (1995)
15. Voorhees, E.M.: Overview of the trec 2001 question answering track. In: Proceedings of the Tenth Text REtrieval Conference (TREC), pp. 42–51 (2001)
16. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 26–32. ACM, New York (2003)