Enhanced semantic expansion for question classification

Ali Harb, Michel Beigbeder, Kristine Lund, Jean-Jacques Girardot

Enhanced semantic expansion for question classification

Ali Harb* and Michel Beigbeder

École National Supérieure des Mines de Saint-Étienne,
Laboratory for Information Sciences and Technology (LIST),
42023, Saint Etienne, France
E-mail: harb@emse.fr
E-mail: mbeig@emse.fr
*Corresponding author

Kristine Lund

ICAR, CNRS,
University of Lyon,
ENS-LSH, 15 Parvis René Descartes,
BP 7000, 69342, Lyon, France
E-mail: kristine.lund@univ-lyon2.fr

Jean-Jacques Girardot

École National Supérieure des Mines de Saint-Étienne,
Laboratory for Information Sciences and Technology (LIST),
42023, Saint Etienne, France
E-mail: girardot@emse.fr

Abstract: Most question and answering systems are based on three research themes: question classification and analysis, document retrieval and answer extraction. The performance in every stage affects the final result. To respond correctly to a question given a large collection of textual data is not an easy task. There is a need to perceive and recognise the question at a level that permits to detect some constraints that the question imposes on possible answers. The classification of questions appears as an important task because it deduces the type of expected answers. The purpose is to provide additional information to reduce the gap between answer and question. A method to improve the performance of question classification focusing on linguistic analysis and statistical approaches is presented. This work also proposes two methods of questions expansion. Various questions representation, term weighting and diverse machine learning algorithms are studied. Experiments conducted on actual data are presented. Of interest is the improvement in the precision on the classification of questions.

Keywords: classification; feature selection; semantic expansion; mutual information; MI; machine learning; text mining.

Reference to this paper should be made as follows: Harb, A., Beigbeder, M., Lund, K. and Girardot, J-J. (2011) 'Enhanced semantic expansion for question classification', *Int. J. Internet Technology and Secured Transactions*, Vol. 3, No. 2, pp.134–148.

Biographical notes: A. Harb is a PhD student majoring in Computer Science at École des mines of Saint-Étienne. He received his Masters in Computer Science from Montpellier II University (France) in 2008. His thesis is supervised by Professor Jean-Jacuqes Girardot and Assistant Professor Kristien Lund. He is working on a model for information retrieval within interactive traces of collaborative activities.

M. Beigbeder defended his PhD in 1988 in the image synthesis domain. Then, he worked on distributed and networked application paradigms. He has been working in the information retrieval domain since 1995. He is interested in different subjects from global architecture of distributed information retrieval systems, taking into account their scalability to high precision models of information retrieval.

Kristine Lund is a CNRS Research Engineer in Human and Social Sciences and is currently Vice Director of the Language Sciences Laboratory ICAR at the University of Lyon. In 2003, she obtained her PhD in Cognitive Science from the University of Grenoble, in which she developed an analytical model for explanation during face-to-face-human interaction. Her research interests include the multimodal co-construction of complex knowledge in goal-oriented computer-mediated human interaction. She recently co-directed PhD work on trace analysis tool for interaction analysts (Tatiana), the four main functionalities of which are synchronisation, transformation, analysis and visualisation of interaction data.

Jean-Jacques Girardot is a Professor and the Head of the Information Retrieval Laboratory, at the École des Mines de Saint-Étienne. He received his PhD in Computer Sciences in 1976, and a thesis in mathematics in 1989. He has been involved in the domain of programming languages, networks and information retrieval. He presently works on the analysis of corpus of interaction. He had been working during the '90s at the IBM Watson Research Center in Yorktown Heights, NY, USA. He has published a large number of papers in journals, conferences and workshops.

1 Introduction

With the ongoing expansion of the web, the number of documents becomes increasingly both significant and considerable. During an information search, the user is faced with numerous documents returned by search engines, many of which are not relevant. Question answering (*QA*) systems are viewed as a way to rapidly deliver information, particularly in response to specific questions. Classifying the questions to anticipate the type of answer is a very important step in an effective QA system. This useful task appreciably reduces the search space to identify the correct answer and improves the quality of service.

Traditionally, many QA systems use manually constructed rules (Kosseim and Yousefi, 2007; Plamondon et al., 2003; Kangavari et al., 2008; Saxena et al., 2007) to typify the questions, which is not very efficient for maintenance and upgrading. Recently, with the growing popularity of statistical approaches, machine learning was applied to detect the categories of questions (Harb et al., 2009; Krishnan et al., 2005; Hacioglu and Ward, 2003; Zhang and Lee, 2003; Li and Roth, 2006; Fu et al., 2009). The advantage is

that machine learning algorithms can recognise among discriminating features, and rely on the learning process to efficiently cope with the features. The bag-of-words (BOW) representation is frequently used in the classification tasks using machine learning. However, where the questions are short, we need to combine many features in question representation and to add relevant information to let classifiers achieve higher precision.

In this work, several possible semantic information sources will be described that differ in their granularity and method of acquisition. Then, these new enhanced sources will fill-out the semantic sense of the questions. For instance, this integration should help to obtain the grammatical category of terms (noun, verb, adjective, etc.), the semantic categories of nouns (e.g., person, location), or synonyms, hypernyms, hyponyms of nouns. The actual sense of terms must be retained and for this we will use their context.

The paper is organised as follows: Section 2 describes the principal techniques used in question classification. In Section 3, we present a study on the different features used in question representation. Section 4 details the experiments conducted on various features with three learning algorithms. Our approach is summarised in Section 5, where we will describe the general expansion method. Section 6 presents the experimental results conducted on actual data.

2 Related work

Question classification approaches can be divided into two main groups: one composed of manually constructed sets of rules and the other based on machine learning. In the former hand written grammar rules are used to parse a question and to extract significant patterns Kosseim and Yousefi (2007). The QUANTUM system, Plamondon et al. (2003) define a set of 40 rules which properly classifies 88% of the 492 questions collected from TREC-10. The systems described in Kangavari et al. (2008) and Saxena et al. (2007) use a set of rules based on the determinant words (e.g., *who*, *where*, ...). Thus, for each determinant word they build a special category question type (e.g., *who* questions are classified as requiring answer type *person*). Those manual rules are difficult and time consuming to construct. Their coverage is limited, because it is almost impossible to anticipate all the question categories. Thus, this influences the effectiveness of the entire QA system. During evolution of the taxonomy or when a new taxonomy is adopted, many previously prepared rules have to be scrapped, modified or completely rewritten.

In the latter group that uses machine learning expert knowledge is replaced by a learning corpus containing labelled questions. Using this corpus, a classifier is trained in a supervised mode. Possible choices of classifiers include but are not limited to: neural networks (NN), naive Bayes (NB), decision trees (DT), support vector machines (SVM) and K nearest-neighbours method (KNN). Reconstruction of a learned classifier is more flexible than of a manually constructed system because it can be trained on a new taxonomy in a short time.

Zhang and Lee (2003) compare a SVM-based classification system to other machine learning approaches (KNN, NB, and DT). All these classifiers use the BOW model and are trained on the same learning corpus. Recently, Li and Roth (2002) used the SNoW learning architecture Khardon et al. (1999) for question classification. They constructed the UIUC question classification corpus. In this work, they used part-of-speech tags, parsing, head chunks (first noun in a question) and named entities. They achieved 78.8% accuracy.

In recent years, numerous question taxonomies have been defined, however there is not a single standard used by all systems. Most of the participants in the TREC-10 campaign implement their own question taxonomy. Moreover, their taxonomy is frequently redefined from year to year. Usually, the systems use a taxonomy consisting of less than 20 question categories. However, as demonstrated by several QA systems, employing detailed taxonomy consisting of fine-grained categories is beneficial (Zhang and Lee, 2003; Li and Roth, 2002). More recently, the taxonomy and corpus described in Li and Roth (2006) has become the most frequently used in current research.

Hacioglu and Ward (2003) describe a system using a support vector machine with word bigrams which obtains a precision of 80.2%. Most recently, Krishnan et al. (2005) used N-grams (N = 1 or 2) and integrated all the hypernyms of words and achieved 86.2% on the same corpus of questions with the same taxonomy. In the work of Fu et al. (2009), they used a classifier based on SVM combined with a question semantic similarity. Harb et al. (2009) describe a comparison between the performances of different classifiers. They extract discriminant words from question. Furthermore, they expand question by hypernyms. They achieved an accuracy of 80.9%. Later, in Li and Roth (2006), they used more semantic information sources including named entities, WordNet and class specific related works. Using these, they were able to achieve the best accuracy 86.3%.

In this work, we propose to use a combination of known and new features. However, we will expand question with terms that preserve the actual sense of the original words. For this endeavour, we propose and test several types of semantical expansions and their combinations.

3 Features

In any automatic classification of text, the choice of the instance representation to be processed (in our case questions) and the operations to be applied is crucial. With the *BOW* representation the single information used is the presence or the frequency of certain words. Many researchers have chosen to use a vector depending on the Salton model (Salton and Buckley, 1988). This representation transforms each question in a vector of n weighted words. Initially, the descriptors of the text may well simply be all the unique words in the documents. It is possible to use other types of features to characterise vectors, some of which will be presented later.

Our analysis revealed that some syntactic and semantic information that frequently exist in questions and belonging to the same category do not appear in the others. So, exploiting this information will provide valuable clues to classifiers to supplement the simple BOW approach.

3.1 Syntactic features

In addition to the words themselves, the syntactic features for each question include lemma, part-of-speech tag (verb, noun, adjective, etc.), the result of syntactic analysis and in particular grammatical dependencies. Significant words in sentence (e.g., *object*, *subject*, ...) can be detected with the use of grammatical dependencies.

3.2 Semantic relationships between words

Words can be related in several different ways. These relationships, in controlled vocabularies, can be categorised into many important classes such as: equivalency and hierarchy. The primary relation in the equivalency class is that of synonymy. Specifically, synonymy describes the relationship between two words that have the same conceptual meaning (e.g., *city* and *town*). In the hierarchy class, hypernymy describes the semantic relation of being superordinate or belonging to a higher class (e.g., *flower* and *plant*). The semantic information provides context-based knowledge of word meanings to any classification system and improve results.

3.3 Named entities

This feature assigns a semantic category to some nouns in the questions. The presence of those named entity tags in questions will favour the common semantic discriminant belonging to the same question type. (e.g., *Who is the first president of France*?, the named entity tagger will get: *Who is the [Num first] [Vocation president] of [Country France]*). As we can see, we obtain additional semantic information expressed in the categories of first (Number), *president* (Vocation) and France (Country).

3.4 N-grams

Words N-grams are sequences of N consecutive words. This model is founded on the assumption that the presence of a word is only relevant to the n words before and after it. It embodies the features of word order, and therefore, it can reflect the theme of the sentence more accurately than isolated words.

3.5 Term weighting

Let n be the total number of unique features (e.g., *words*, *N-grams*, etc.) in the corpus. Each question will be represented by a vector of n elements. Each component of this vector can simply be binary or can correspond to the number of occurrences of the feature in the question. However, using frequency accords too much importance to the features that appear very often in all question categories and are hardly representative of one category in particular. Another weight known as term frequency, inverse document frequency (TF.IDF). Salton and Buckley (1988) measures the importance of the words according to their frequency in the question ($tf(t, d)$) weighted by frequency of their occurrence in the entire corpus $\left(idf(t) = \log \frac{|S|}{df(t)}\right)$.

$$tf \cdot idf(t,d) = tf(t,d) \cdot idf(t) \tag{1}$$

$|S|$ is the number of documents in the corpus and $df(t)$ is the number of documents containing t. This of a question. Conversely, a term that appears in many questions will have a low weight.

3.5.1 Training set

We used a training set of 10,343 questions, which is a collection from the following: 5,500 *UIUC DATA* Li and Roth (2002), 1,343 from (*TREC10*, *TREC9*, *TREC8*), 200 from *QA@CLEF2006*, 2,249 from *CRL-QA* and 1,011 from *NTCIR-QAC1*.

3.5.2 Taxonomy

Taxonomy proposed in Li and Roth (2002) was chosen because it has a large coverage of question types and it represents a natural semantic classification for question types. It contains six coarse and 50 fine grained classes, shown in Table 1. We manually annotated the questions of our training set according to this taxonomy with both coarse classes and fine classes.

Table 1 Question classification taxonomy

Coarse classes	*Fine classes*
Abbreviation	Abbreviation, expression
Description	Definition, description, manner, reason
Entity	Animal, body, colour, creative, currency disease, event, food, instrument, lang letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
Human	Description, group, individual, title
Location	City, country, mountain, other, state
Numeric	Code, count, date, distance, money order, other, period, percentage, speed, temp, volume size, weight

3.5.3 Result of experimentation

For the first experiments, the representation of the questions is simply accomplished by filtering the stop words by using a list of stop word http://www.lsi.upc.es/padro/lists.html. In our approach, the words which are common and therefore not useful are filtered by using a list of 'stop words', modified to keep the words judged pertinent for our application (e.g., who, where are discriminating words in questions, and thus are not deleted). We applied the three learning algorithms. Their results are presented in Table 2 in the 'stop word' row.

In the second series of experiments, we used treetagger (Schmid, 1994) to obtain the part-of-speech tag, and then a filtering step using the list of stop words. The results are in the POS row of Table 2, and they show an improvement over the previous ones.

In the third series of experiments we used word N-grams. The results are presented in the last three rows of Table 2. For N = 1, we only consider unigram, and thus the method is the same as the simply filtering one 'stop word'. Moreover, the results obtained with N-grams with N greater than 4 were worse. This is why hereafter we will only refer to N-grams with N between 2 and 4. The best results are obtained with N = 2, and then the efficiency decreases as N increase. The best results are obtained with the SVM classifier.

With these experiments we have demonstrated that the use of either N-grams or part-of-speech tags or filtering improves the result. So, we then tried to combine those features. We successively applied treetagger, then filtering of stop words and N-grams with $N \in \{2, 4\}$.

Table 3 displays the results of the three classification learning algorithms on the same training set of questions. In general, we note that the results deteriorate when N = 3 or 4. The best results are obtained with N = 2 and with the SVM algorithm compared to K-NN and naive Bayes. The precision of classification improved from 68.2% to 73.5% with SVM and weighting with TF.IDF.

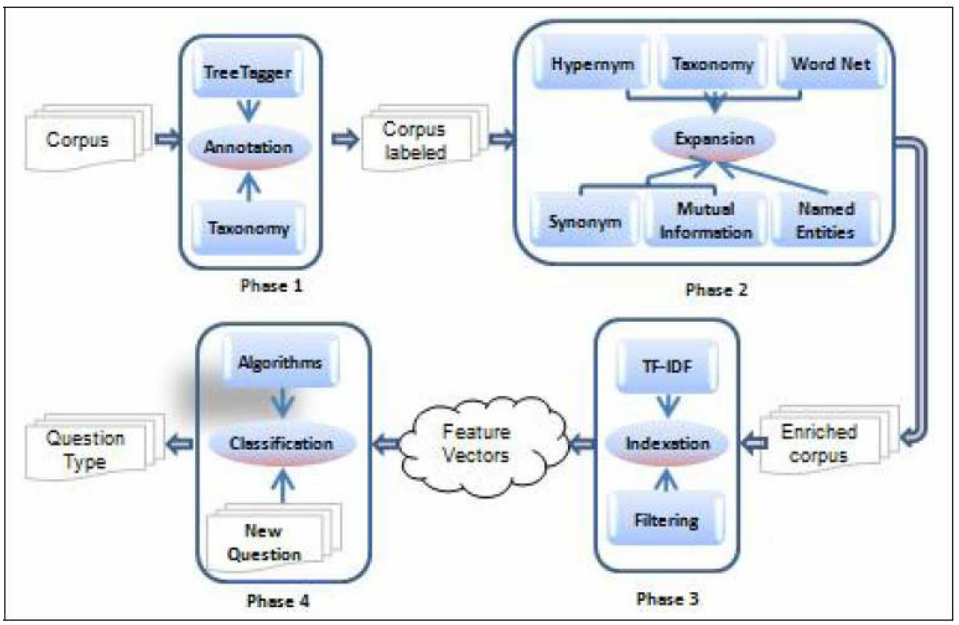
Table 2 Features classification results

Algorithms	*K-NN*		*SVM*		*NB*	
Weighting	*Frq.*	*tf · idf*	*Frq.*	*tf · idf*	*Frq.*	*tf · idf*
Filtration	52.5	55.7	57	58.6	56.7	57.2
Lemma	60.9	62.1	62,8	63.9	62.2	62.4
2 grams	65.8	66.9	67.7	68.2	67.9	68.1
3 grams	62.1	63.8	64.3	66.2	65	64.2
4 grams	54.4	56.7	55.8	56	53.1	54.9

Table 3 Combination classification results with lemmatisation, filtering, weighting and N-grams

Algorithms	*K-NN*		*SVM*		*NB*	
Weighting	*Frq.*	*tf · idf*	*Frq.*	*tf · idf*	*Frq.*	*tf · idf*
1 gram	60.9	62.1	62.8	63.9	62.2	62.4
2 grams	71.3	71.2	72.5	*73.5*	71.9	72.3
3 grams	64.7	66.3	65.9	66.7	65.3	65.7
4 grams	55.9	57.1	56.7	57.2	56.3	56.6

Figure 1 The main process of Sacseq approaches (see online version for colours)



3.6 Discussion

Up until now, we have not used any semantics for classification. However, questions that should be classified in one category can also use different vocabulary either synonyms (e.g., *birth* and *cradle*) or hypernyms (e.g., *city* and *town*).

In what follows, we will present our approach based on a combination of the various features discussed in this section, and the expansion of the questions with hypernyms, synonyms, named entities for addressing the semantics of words.

4 Approach Sacseq

The aim of this section is to present the Sacseq approach (Cf. Figure 4). The general process is composed of four principal phases.

Phase 1 Corpus preprocessing

Label all the questions of the corpus according to the taxonomy. Thereafter, treetagger is used to annotate the words with part-of-speech tag, and their lemma. This information is relevant for the selection of a specific word category.

Phase 2 Question expansion

1 Projection and hypernym

Regarding the vocabulary diversity and the shortness of questions, many of semantically correlated words are treated as different (e.g., *city* and *town* are semantically correlated). We also try to expand questions with more general terms, as they unify the sense of the nouns. The notion of projection refers to a matching between a given word and the term representing a taxonomy concept.

The idea is to enrich the questions with synonyms or hypernyms of the nouns belonging to the question. For this purpose, we use WordNet. For each noun, the hypernym feature of WordNet provides a set of generic words at different levels, starting with the most specific and going to those who represent a broader meaning of the noun. While searching for hypernyms of noun words, we have to preserve the original semantics of the words. The different steps of 'projection and hypernym' expansion are the following:

- extract nouns from question
- for each noun, project on the 50 concepts of the taxonomy and their instances, if this noun belongs to an instance of a concept, the question will be expanded by this concept
- if it is identical to a concept, no changes are made
- otherwise, for each remaining noun collect the set of hypernyms

- preserve the order assigned by wordNet to reflect the hypernym level of abstraction, then seek the first of these hypernyms that projects into the instances of our concepts or the concepts themselves.

2 Synonym expansion

This step proposes a new methodology to expand the questions with synomyms. Again, WordNet is the resource for finding synonyms of words. To maintain a semantic enrichment, we study the semantic correlation between a word and its set of synonyms. One of the most commonly used measures for finding how two words are correlated is cubic mutual information (MI) (Downey et al., 2007). This measure depends on a context *c*. Given this context it is based on three frequencies: *nb*(*x*, *c*) the number of co-occurrences of *x* and *c*, *nb*(*y*, *c*) the number of co-occurrences of *y* and *c*, *nb*(*x*, *y*, *c*) the number of co-occurrences of *x*, *y* and *c*. The measure then is computed with $AcroDef_{MI^3}$ formula (Roche and Prince, 2007):

$$AcroDef_{MI^3}(x, y, c) = \frac{nb\left((x, y) \wedge c\right)^3}{nb(x \wedge c) \cdot nb(y \wedge c)} \tag{2}$$

We keep only the synonyms that were judged semantically close by the threshold of the highest similarity among the group of synonyms, then, we expand questions with the relevant synonyms.

To implement this measure, firstly we must identify the context *c*. We use a syntactic analyser (de Marneffe and Manning, 2008) to extract syntactical relations from the questions. Those relations define the (*grammatical dependencies rules*) among words of questions which are used later as the context of addressed words. To evaluate the three frequencies we post queries to Google. We then only retain synonyms for which the value of 2 is greater than a threshold, so that they are semantically correlated to the original term.

The following example illustrates the information available when applying the synonym expansion.

Question What is the capital of the French Republic?

The *grammatical dependencies* are: attr (is, what), det (capital, the), nsubj (is, capital), det (republic, the), amod (republic, French) and prep of (capital, republic).

After the parsing we detect that *capital* is the subject of the sentence. Based on the grammatical dependencies found (*amod* and *prep_of*), we find that *French Republic* is the context of *capital*. When collecting the set of synonyms of capital with WordNet we find: *seat of government*, *city*, *principal*, *assets*, and *wealth*. The original term *capital* in the context *French Republic*:

1 $AcroDef_{MI^3}$ (*capital*, *seat of government*, *French Republic*) = 3.57×10^{-1}

2 $AcroDef_{MI^3}$ (*capital*, *city*, *French Republic*) = 2.24×10^{-2}

3 $AcroDef_{MI^3}$ (*capital*, *assets*, *French Republic*) = 1.16×10^{-4}

4 $AcroDef_{MI^3}$ (*capital*, *wealth*, *French Republic*) = 1.0078×10^{-4}

5 $AcroDef_{MI^3}$ (*capital*, *principal*, *French Republic*) = 3.097×10^{-6}

The list above illustrates the $AcroDef_{MI^3}$ values for all these synonyms with the original term *capital* in the context *French Republic*. As we can see, regarding the context, the first two synonyms *seat of government* and *city* are the most appropriate and they have the largest values. Thus, we keep just these two synonyms to expand the question.

3 Named entities

After the two steps of question expansion by *projection* and *hypernyms* and *synonyms*, we use IdentFinder (Bikel et al., 1999) to assign a semantic category to some nouns in the questions. IdentFinder is able to tag 7 types of named entities person, description, location, profession, money, number and date.

Phase 3 Vectorisation

In this phase, first the questions of the corpus most be filtered using the list of stop words. Then, all the N-grams are extracted. Each N-gram is considered as a dimension of the vector space. Each question is then converted into a vector where the number of occurrences is weighted with TF-IDF.

Phase 4 Learning and classification

In this phase, 10-fold cross-validation is employed. Firstly, the classifier model is learned using the training corpus constituted of 90% of the corpus. Again, the three learning algorithms are SVM, KNN and NB. The classifier model is built by combining a sequence of two classifiers. The first classifies questions into the 6 coarse classes and the second into the 50 fine classes. Each uses the same learning algorithm. Then the classifier model is used to assign a class to each new question of the 10% rest of the corpus.

Algorithme of Sacseq. The principle of Algorithm 1 is as follows:

Algorithm 1 Sacseq

Input: The learning Corpus of questions Q,
Taxonomy T, WordNet, $AcroDef_{MI^3}$,
Threshold β

Output: Space vector V;

```
1  begin
2      for each q in Q do
3          q_L = TreeTagger(q);
4          Q_L = Q_L ∪ q_L;
5      for each q_L in Q do
6          S_GD = Dependencies(q_L);
7          for each Noun in q_L do
8              S = ϕ;
```

```
9          if Project(Noun, T) == True then
10             Expand(Noun, q_L);
```

Algorithm 1 Sacseq (continued)

```
11         else
12             S = WordNet_Hyp(Noun);
13             for each s in S do
14                 if Project(s, T) == True then
15                     Expand(s, q_L);
16                     break;
17         S_s = φ;
18         S_s = WordNet_Syn(Noun);
19         for each s in S_s do
20             AcroDef_MI³ (s, Noun, S_GD)
21         Sort(S_S)
22         Filtre(S_s, β)
23         Expand(S_s, q_L);
24     Q_{L_F} = Filter_{St−W}(Q_L);
25     Q_A = Part(90%, Q_{L_F});
26     for each q in Q_A do
27         N – grammes(Q_A);
28     tf · idf(Q_A);
29     Vectorise(Q_A);
30     return V;
31 end
```

For each question, we apply *TreeTagger* for stemming and to get the grammatical category of the words. For all the nouns in each question of the new corpus, we use *Project* for project these nouns at the instance of the taxonomy. If the projection fails, we use $Wordnet_{Hyp}$ to search for hypernyms. Then, again we project the hypernym on instances of the taxonomy. In the step of synonyms, Wordnet_{Syn} extract the synonym for each noun, then we calculate the correlation force between the noun and this set of synonyms using $AcroDef_{MI^3}$ (s, *Noun*, S_{GD}), followed with *Filtre*(S_s, β) to retain the relevant ones. At this step, we apply the filter to delete non relevant word using the list of

stop words. The application of *N-grams* followed by TF-IDF will give us the space vector *V*, which will then be used as the training set to learn the algorithm.

4.1 Experiments

In this section, the results of the different experiments we conducted to validate our methodology are presented. We will particularly look at the following point:

- What are the consequences of the choice of the features on the quality of classification?

In the first experiments, classification is performed only with the integration of *Projection* and *Hypernym*, whose aim is to evaluate the improvement brought on by this method. We limit the calculation of frequencies to TF-IDF and we only use bi-grams. Results are presented in Table 4.

Table 4 Classification results using projection and hypernym

Algorithmes	*K-NN*	*SVM*	*NB*
Projection and Hypernym	78.6	80.9	80.1

With the *projection* and *hypernym* method, we find that the percentage of correctly classified questions was improved with the three classification algorithms, and significantly with SVM which was improved by 7.4% (from 73.5% to *80.9*% (Table 4 row *projection and hypernym*).

In the next experiments, we expanded questions with *synonyms*. Table 5 displays the classification results. The precision again is improved with the three learning algorithms, and especially with SVM which was improved by 5.1% (from 73.5% to *78.6*%).

Table 5 Classification results using synonyms

Algorithmes	*K-NN*	*SVM*	*NB*
Synonym	78.1	*78.6*	77.9

Table 6 displays results when expanding questions by named entities. Again, the result with all algorithms is improved, especially with SVM by *3.1*% (from 73.5% to *76.6*%). The first two methods of expansion described above perform better than named entities.

Table 6 Classification results using named entity

Algorithmes	*K-NN*	*SVM*	*NB*
Named entity	76.5	*76.6*	76.3

In the final series of experiments, we applied all the semantic steps of our approaches Sacseq. Table 7 displays the classification results. This demonstrates the usefulness of our expansion method and features combination. Precision is improved for all the learning algorithms, especially with SVM increasing by *13.2*% from 73.5% to *86.7*%.

Table 7 Classification results using the Sacseq approaches

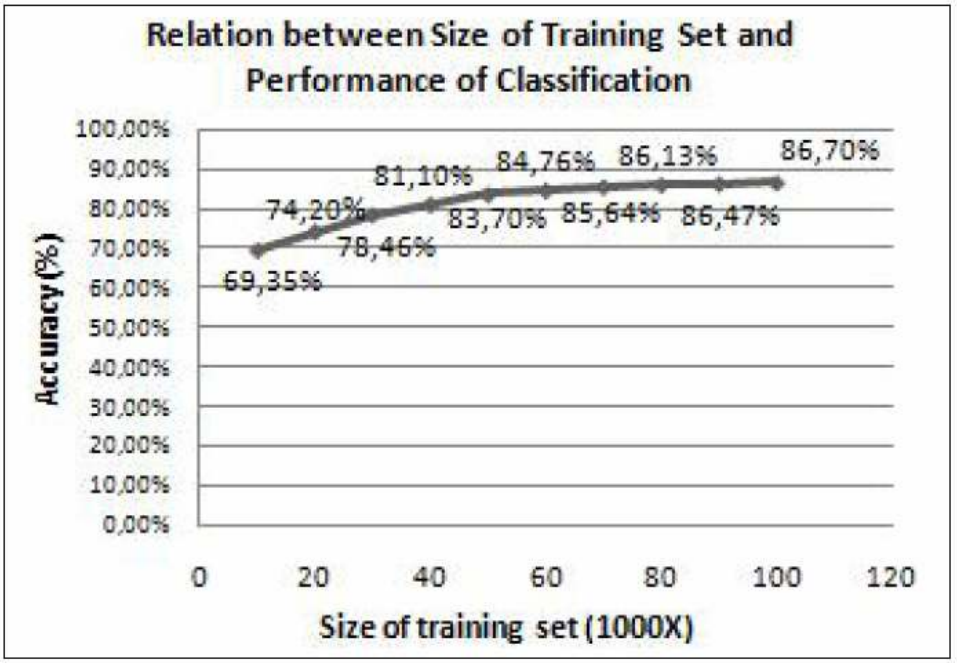
Algorithmes	*K-NN*	*SVM*	*NB*
Sacseq	84.9	*86.7*	85.2

4.2 Experiments related to training sets size vs. classification performance

In this experiment, we want to know how many questions are required to produce a stable and robust training set.

We thus applied the Sacseq method several times. Each time we have increased by 1,000 the number of question until we get a stability on the performance of classification. Figure 2 depicts the relationship between the size of the corpus and the percentage of classification. As we can notice, above 10,000 questions we do not increase much the percentage of classification.

Figure 2 Relation between the size of training corpus and the number of learned adjectives



5 Conclusions

A novel method called Sacseq for automatically expanding terms in questions by *synonyms, projection and hypernym* while retaining the context has been proposed. Various features for question representation were examined. How they influence the performance of the classifiers was determined. The experiments were executed on training corpus. This demonstrated the usefulness of our method for ameliorating the effectiveness of the classification.

Future work may entail a broad series of projects and initiatives. Firstly, our method depends on the quality and the number of questions in the learning corpus. We would like

to study the relationship between classification performance and the size of the learning corpus. Secondly, in this paper, we focused on machine learning. We plan to extend the first step of the classification by applying a set of hand written rules that cover the six coarse categories. Thirdly, we hope to extend this work to support interactive QA, where the users can interact with the system. Finally, we propose to complete this exploratory work within a complete QA system in the context of information retrieval in a corpus of structured documents.

References

Bikel, D., Schwartz, R. and Weischedel, R. (1999) 'Algorithm that learns what's in a name', *Machine Learning*, Vol. 34, pp.211–231.

de Marneffe, M-C. and Manning, C.D. (2008) 'The Stanford typed dependencies representation', in *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Downey, D., Broadhead, M. and Etzioni, O. (2007) 'Locating complex named entities in web text', in *Proceedings of the IJCAI07*, pp.2733–2739.

Fu, J., Qu, Y. and Wang, Z. (2009) 'Two level question classification based on SVM and question semantic similarity', *International Conference on Electronic Computer Technology*, pp.366–370.

Hacioglu, K. and Ward, W. (2003) 'Question classification with support vector machines and error correcting codes', *The Association for Computational Linguistics on Human Language Technology*, Vol. 2, pp.28–30.

Harb, A., Beigbeder, M. and Girardot, J.J. (2009) 'Evaluation of question classification systems using differing features', in *Proceedings of IEEE/ACM ICTST-2009 (The 4th International Conference for Internet Technology and Secured Transactions)*.

Kangavari, M., Ghandchi, S. and Golpour, M. (2008) 'A new model for question answering systems', in *Proceedings of World Academy of Science, Engineering and Technology*, August, Vol. 32, pp.536–543.

Khardon, R., Roth, D. and Valiant, L.G. (1999) 'Relational learning for NLP using linear threshold elements', *The Conference on Artificial Intelligence*, pp.911–919.

Kosseim, L. and Yousefi, J. (2007) 'Improving the performance of question answering with semantically equivalent answer patterns', *Data and Knowledge Engineering*, Vol. 66, pp.53–67.

Krishnan, V., Das, S. and Chakrabarti, S. (2005) 'Answer type inference from questions using sequential models', *The conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp.315–322.

Li, X. and Roth, D. (2002) 'Learning question classifiers', in *Proceedings of the 19th International Conference on Computational Linguistics*, pp.556–562.

Li, X. and Roth, D. (2006) 'Learning question classifiers: the role of semantic information', *Natural Language Engineering*, Vol. 12, No. 3, pp.229–249.

Plamondon, L., Lapalme, G. and Kosseim, L. (2003) 'The QUANTUM question answering system', *Proceedings of the Eleventh Text Retrieval Conference (TREC'02)*.

Roche, M. and Prince, V. (2007) 'Acrodef: a quality measure for discriminating expansions of ambiguous acronyms', *CONTEXT*, pp.411–427.

Salton, G. and Buckley, C. (1988) 'Term-weighting approaches in automatic text retrieval', *Information Processing Management*, pp.513–523.

Saxena, A., Sambhu, G., Subramaniam, L. and Kaushik, S. (2007) 'IITD-IBMIRL system for question answering using pattern matching, semantic type and semantic category recognition', *Proceedings: The Fourteenth Text Retrieval Conference (TREC 2007)*, October, Gaithersberg, MD.

Schmid, H. (1994) *Treetagger*, TC project at the Institute for Computational Linguistics of the University of Stuttgart.

Zhang, D. and Lee, W.S. (2003) 'Question classification using support vector machines', *Proceedings of the 26th ACM SIGIR*, pp.26–32.