Classifying Questions in Question Answering System Using Finite State Machines with a Simple Learning Approach

Mohammad Moinul Hoque
University of Evora, Evora, Portugal
moincse@yahoo.com

Teresa Goncalves
University of Evora, Evora, Portugal
tcg@uevora.pt

Paulo Quaresma
L2F/INESC-ID & University of Évora, Portugal
pq@uevora.pt

Abstract

Question Classification plays a significant part in Question Answering system. In order to obtain a classifier, we present in this paper[1] a pragmatic approach that utilizes simple sentence structures observed and learned from the question sentence patterns, trains a set of Finite State Machines (FSM) based on keywords appearing in the sentences and uses the trained FSMs to classify various questions to their relevant classes. Although, questions can be placed using various syntactic structures and keywords, we have carefully observed that this variation is within a small finite limit and can be traced down using a limited number of FSMs and a simple semantic understanding instead of using complex semantic analysis. WordNet semantic meaning of various keywords to extend the FSMs capability to accept a wide variety of wording used in the questions. Various kinds of questions written in English language and belonging to diverse classes from the Conference and Labs of the Evaluation Forum's Question Answering track are used for the training purpose and a separate set of questions from the same track is used for analyzing the FSMs competence to map the questions to one of the recognizable classes. With the use of learning strategies and application of simple voting functions along with training the weights for the keywords appearing in the questions, we have managed to achieve a classification accuracy as high as 94%. The system was trained by placing questions in various orders to see if the system built up from those orders have any subtle impact on the accuracy rate. The usability of this approach lies in its simplicity and yet it performs well to cope up with various sentence patterns.

1 Introduction

Classifying a question to its appropriate class in an important subtask and plays a substantial role in the Question Answering (QA) systems. It can provide some useful clues for identifying potential answers in large collections of texts. The goal of this current work is to develop a classifier using Finite State Machines (FSM) to classify a set of questions into their relevant classes. Various techniques have already been tried by the community either to classify a question to its relevant class or to a finer subclass of a specific class. Results of the error analysis acquired from an open domain QA system demonstrates that more or less 36.4% of the errors were generated due to the wrong classification of questions (Moldovan et al., 2003). So, this issue can be highlighted as a subject of interest and has arisen the aim of developing more accurate question classifiers (Zhang and W. Sun Lee, 2003). Usually the answers generated from the classified questions have to be exact in nature and the size of the answer has to be within a restricted size (Peters et al., 2002; Voorhees, 2001) which greatly emphasizes the need of an accurate question classifier. Techniques involving Support Vector Machines (Dell Zhang and Wee Sun Lee, 2003; K. Hacioglu and W. Ward, 2003) showed a good accuracy rate of over 96% in classifying questions to their finer classes instead of diverse super classes. Li and Roth (2002) investigated a variety of feature combinations using their Sparse Network of Winnows algorithm (A. Carlson et al., 1999). The Decision Tree algorithm (Mitchell, 2002) was also used for question classification with fair amount of accuracy rate. It is a method for approximating discrete valued target function where the learned function is presented in a tree which classifies instances. Naïve Bayes (Mitchell, 2002) method was also used in the question classification task with limited accuracy rate of around 79.2%. In another work (Fan Bu et al., 2010), where a function-based question classifi-

[1] This work was supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013

27th Pacific Asia Conference on Language, Information, and Computation pages 409－414

cation technique is proposed, the authors of that paper claimed to have achieved as high as 86% precision levels for some classes of questions. Some attempts have been made to develop a language independent question classifier (Thamar Solorio et al., 2004) with not a mentionable success rate.

This work[1] focuses on the questions posed only in English language and uses questions from the Question Answering (QA) track of the Conference and Labs of the Evaluation Forum (CLEF) (QA4MRE, 2013). It classifies the questions into 5 major classes namely Factoid (FA), Definition (DE), Reason/Purpose (RP), Procedure (PR) and Opinion (OP) Class. CLEF QA track have some diverse types of questions and we are required to fit each of the questions into any of the above mentioned classes. Factoid class of questions are mainly fact oriented questions, asking for the name of a person, a location, some numerical quantity, the day on which something happened such as 'What percentage of people in Bangladesh relies on medical insurance for health care?', 'What is the price of an air-conditioning system?' etc. Definition questions such as 'What/Who is XYZ?' asks for the meaning of something or important information about someone or an organization. 'What is avian influenza?', 'Define SME', 'What is the meaning of Bluetooth signal?' are some examples of the definition class questions. Reason/Purpose questions ask for the reasons/goals for something happening. 'Why was Ziaur Karim sentenced to death?' and 'What were the objectives of the National meeting?' are the example questions of this class. Procedural questions ask for a set of actions which is an accepted way of doing something. Such as: 'How do you calculate the monthly gross salary in your office?' Opinion questions ask for the opinions, feelings, ideas about people, topics or events. An example question of this type may be like 'What did the Academic Council think about the syllabus of informatics department?' A question is either mapped to only one class or may be classified as 'other'.

The next section of the paper describes the procedure used to create the states and transitions in the FSMs involving a simple learning mechanism and the section 3 presents the data set for the experimental verification of the procedures and outcome of the experiments followed by a section covering a discussion about the future works.

2 Classification using Finite State Machines (FSM) with learning strategy

From a large number of questions derived from the gold standard set of QA track of CLEF 2008 - 2011 and observing them manually, we came to a conclusion that it is possible to classify a set a questions using a set of FSMs and the FSMs can be automatically built and adjusted according to the questions in the training set and later on can be used to classify the questions appearing in the test set. Initially we start off with some elementary states for each of the FSMs beginning with different headwords. The headwords are usually What, Why, How, When, Where etc. Questions that do not begin with a known headword can be restructured to a suitable form. For example, 'In which country was the Vasco da Gama born?' can be changed to 'What country was the Vasco da Gama born?'. Similarly, 'What does SME stand for?' can be reformatted to 'What is SME?' and so on. The initial preprocessing module performs this question restructuring step. A set of non stops words of English language are extracted from the question instances which we call a Keyword set. The preprocessing module also converts the keywords into its present tense and singular form to make sure that the keywords 'thought' and 'think' are treated similarly. It also reduces the number of keywords in the set. Each FSM is represented with a directed graph and may have more than one state for each question class. Those states are called final states. Rests of the intermediate states are called 'undefined' (UN).

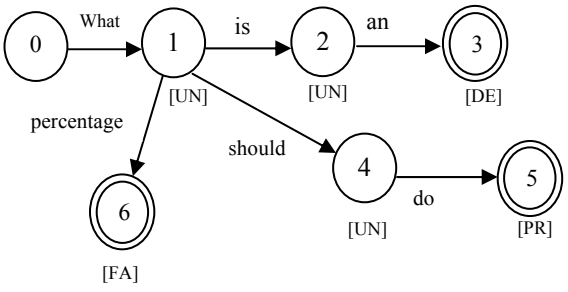

Figure 1. An FSM accepting questions Q1 and Q2.

An FSM can have many intermediate and undefined states as well as transitions between them. The inputs to a FSM are keyword tokens extracted from the question. An example FSM beginning with the headword 'What' and accepting the questions Q1: 'What is an SME?' and Q2: 'What percentage of people relies on TV for news?' is depicted in the figure 1.

2.1 Learning new states and transitions in the FSM using keywords

FSMs continue to build up the states and transitions as it encounters more new question instances. Each of the questions in the training set is tokenized removing a few English stop words and the keywords are then isolated from each of the questions to form a keyword structure (KS). Every keyword in the KS has a weight in context with the other keywords appearing in the question. In order to calculate the relevant weights of the n number of keywords, a Keyword Frequency Matrix (KFM) of n *x* n dimension is created first and the frequency of every keyword appearing before and after of every other ones is stored in the matrix. This KFM is prebuilt from all the question instances of the training set. Table 1 shows a dummy KFM with some sample frequency values.

before \ after	*What*	*The*	*Is*	*Meaning*	*Think*
What	5	80	90	16	8
The	1	5	120	16	8
Is	2	71	6	12	6
Meaning	0	0	1	0	0
Think	0	3	2	0	0

Table 1. A sample 5x5 Keyword frequency matrix (KFM)

When we are using a sentence to build or train up the FSMs, a subset of the KFM is created using only those keywords which are appearing in the question sentence and the weights of each keyword Z in the question sentence is calculated in context with other keywords appearing in that sentence using the formula followed. The formula sums up all the frequency values where the keyword Z appears after each of the other keywords in the sentence and subtracts from it the sum of the frequency values where Z appears before each of the other keywords in that question sentence. There are many keywords in various question sentences which appear more frequently than some other rarely used keywords. In order to make sure that such keywords do not receive highest weights all the time compared to the other significant but less frequently used keywords, we divide the weight value with the sum of the frequency value of Z where Z appears after each of the other keywords in the question sentence. This process normalizes the weight value of a keyword within the range 0.0 to 1. In case of a negative weight, the weight is set to 0.0.

Probable Weight (Z) =

$$\frac{[\sum_{j \in row} KFM(j)(index\ of\ Z\ in\ column) - \sum_{j \in col} KFM(index\ of\ Z\ in\ row)(j)]}{\sum_{j \in row} KFM(j)(index\ of\ Z\ in\ column)}$$

Finally, the keyword structure (KS) is built from the question sentence and it comprises of the keywords along with their weights. An FSM is selected based on the headword appearing in the question sentence and it is built using the *Algorithm1*.

The *Algorithm1* detects the keyword boundary from the Keyword Structure (KS) which is the position of the keyword having the highest weight value. If there are multiple keywords having the same highest weights, the position of the first keyword with the highest weight value is marked as the keyword boundary position. The FSM does not take any keyword as input beyond this boundary position to build up on its own. When creating a transition to a state for an input keyword, synonyms of the keyword if there are any are also derived with the help of WordNet (George A. Miller, 1995; Christiane Fellbaum, 1998) and are added as inputs to that transition to extend the machines capability significantly.

Major steps of Algorithm1:

For *every keyword* K_i in the KS of a question sentence
  Mark the keyword boundary which is the first position of the highest weighted keyword
End for
For *every keyword* K_i within the keyword boundary position in the KS
  Try to go through the FSM states using K_i as input to the FSM starting from the state S_0
    If a valid state S_j can be reached using a transition path with K_i as input
      Continue to repeat the above step with the next K_i from the state S_j
    Else
      If for the input K_i , no transition path can be found from state S_f
        If K_i and K_{i-1} were same
          Create a loop transition in that state S_f
        Else
          Create a new state and add a transition from the current state S_f to that new state
          Set the input of the transition as K_i and also the synsets(K_i) using WORDNET
        End if
      End if

If K_i appears at the keyword boundary
Set the class of the state S_f according to the already labeled class of the question.
Else
Set the class of the state S_f to 'UNDEFINED'
End if
End for

2.2 Voting function for a state

Different ordering of the similar kind of question sentences belonging to different classes can mislead the development of an FSM with wrong classification states. For example, the question, 'What is the aim of the raid spectrum policy?' may be classified as a factoid question in one training set where as there may be 5 more questions of similar pattern that are classified as Reason/Purpose question in another training set. In this case, we propose a simple voting algorithm approach. In the voting process, every question in the training set which terminates at a final state with a keyword appearing at the keyword boundary will vote for the class of the state class in the questions labeled class. The class of that state of the FSM will finally be determined according to the class that gets the maximum vote. Voting function ensures that an FSM does not label one of its final states to a wrong class because of the different ordering of the questions appearing in the training set.

Major steps of the Voting Algorithm:

For every FSM_i in the FSM set
For Every Question Q_i in the question set
For Every Keyword K_i in the Q_i appearing at the keyword boundary and terminating at a final state S_f in the FSM_i
Cast a vote for that state S_f in favor of the class that Q_i itself belongs to
End for
End for
Update each of the states of the FSM_i to that class which gets the maximum vote
End for

3 Experimental verification

For the experimental purpose, we took questions from CLEF question answering track for the year 2008-2011. A total number of 850 questions of various classes were selected for the evaluation purpose. Around 400 questions from various years were selected for the training purpose and a test set was created with the rest of the questions. The system was trained with a training set and a test set was used to test the capability of the system. Effectiveness of the classification was calculated in terms of precision and recall and the accuracy was calculated from the confusion matrix (Kohavi R., and F. Provost, 1998).

In order to make sure that the system does not get biased with specific question patterns, we have trained and tested the system in various ways to see if any subtle changes occur in the case of precision and recall. We also trained the system with 50% questions from one year mixed up with the 50% question from another year to cope up with the variations used in question wording and syntactic structure. We also changed the question order to see if the FSMs built from different order cause any considerable error or not. Keyword frequency matrix was trained using a dataset and it continued to update itself with the introduction of new questions from the training set. The data set and the result is presented in table 2. Throughout the training process, voting function was kept activated.

Year	No. of Questions for training	No. of Questions for testing	Accuracy
2008	50	50	96.5190%
2009	250	250	94.1777%
2010	80	90	95.1278%
2011	40	40	90.6014%
Mixed Set of questions	400	450	93.2111%

Table 2. Data set for the question classification using FSMs

Precision and Recall for each class is calculated and is shown in Table 3 followed. From the data in Table 3, we can see that most of the questions were correctly classified by the FSMs, because it could find correct patterns for the questions belonging to specific classes. Wrong classifications were made in some cases where almost similar pattern existed in questions belonging to two different classes. Fortunately, our voting function took the feedback from the questions and responded accordingly to reduce the classification error by a margin. Because of the inaccurate calculation of weights for some keywords in context with the other also played a role to the errors, though most of the time, the weight calculation function provided near correct assumption. In order to check the building procedure of the FSM during the training stage using

the training data, we were concerned about the question ordering. We have created an *n x n* question index matrix with each of the questions in the training set having an index number in that question matrix. We have randomly selected a question index and started to train the system from there on. The next question selected for the training was the question that was most similar to the previously selected one.

Question Class	2008	2009	2010	2011	Mixed
DE (Precision)	1.0	1.0	1.0	0.811	0.981
DE (Recall)	1.0	0.938	0.966	0.721	0.921
FA (Precision)	1.0	0.899	0.903	0.904	0.967
FA (Recall)	1.0	0.955	1.0	0.964	0.911
OP(Precision)	0.0	0.0	1.0	1.0	1.0
OP (Recall)	0.0	0.0	1.0	1.0	1.0
PR(Precision)	0.0	0.977	1.0	0.89	0.965
PR(Recall)	0.0	0.957	0.939	0.85	0.991
RP(Precision)	0.0	0.959	1.0	1.0	0.978
RP(Recall)	0.0	0.967	1.0	1.0	0.988

Table 3. Measure of precision and recall for every class of questions based on the data in the test set

The similarity was calculated in terms of most similar words or their synonyms appearing in two of those questions.

Question Class	Trained with most similar ones. Overall Accuracy 92.1%	Trained with most dissimilar ones Overall Accuracy 94.1%
DE (Precision)	0.942	0.962
DE (Recall)	0.943	0.943
FA (Precision)	0.913	0.921
FA(Recall)	0.891	0.890
OP(Precision)	1.0	1.0
OP(Recall)	1.0	1.0
PR(Precision)	0.911	0.925
PR(Recall)	0.986	0.962
PR(Precision)	0.912	0.911
RP(Recall)	0.912	0.921

Table 4. Precision and recall measure for every class of questions with a change in question order.

We did 4 runs and in every run, we have selected the first question to train randomly and made sure that the same question does not get selected twice. We did the same kind of training by picking the most dissimilar questions to train the system and did 4 runs for that case as well.

The average of the runs is listed in table 4. We can observe from the average run that, no significant change in accuracy, precision and recall are noticed with the change occurring in the question order although the FSMs built from different ordering of the questions had different states or transitions.

4 Discussion and future works

In this current work we have tried to take a practical yet simpler approach towards the question classification problem. The approach came into existence when we realized that most of the time, we don't need to go through all the words and their semantic meanings in detail to map the questions to different classes. We thought it may be useful to give the machine this kind syntactic knowledge and a little semantic understanding to some extent to make it capable of classifying questions to its various classes. Instead of deriving handcrafted rules by watching each of the questions manually, we tried to establish a formalism through the Finite State Machines where the syntactic structure of the sentences could be learnt gradually with the example instances.

The state of the art techniques used so far have already used various mechanisms for addressing question classification problem. Support Vector Machine (SVM) and Conditional Random field used for classifying questions in 6 major classes have achieved an accuracy rate of 93.4% (Zhiheng Huang et al., 2008) whereas, the use of SVM for coarse grain questions have achieved an accuracy rate as high as 97%. Maximum Entropy Model could achieve an accuracy rate around 93.6% and the combined approach of Decision tree and SVM demonstrated a 90% accuracy rate.

The result that we achieved in this work shows that, the approach can be handy and may cope with various types of syntactic structure used for creating question sentences.

Because of not using deep semantic meaning analysis, our system failed to classify some of the questions to their corresponding classes. Lack of a proper recognizable structure was responsible for the failure in those cases. The system also made a few wrong classifications when some very similar structures belonging to two different classes of questions came into existence and this can be observed from the result that we achieved from the recall parameter measurement of the each of the classes. The voting function we used rescued us to some extent to handle such situations.

The result that we achieved encourages us to carry on with this approach further to improve it and use it in the other problem domain such as identifying question focus or may be in classifying questions to their finer classes.

References

A. Carlson, C. Cumby, J. Rosen, and D. Roth. 1999. *The snow learning architecture*, Technical Report UIUCDCS-R99-2101, University of Illinois at Urbana-Champaign.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database. Cambridge*, MA: MIT Press.

Dell Zhang and Wee Sun Lee. 2003. *Question Classification using Support Vector Machines*, In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval.

Fan Bu, Xingwei Zhu, Yu Hao and Xiaoyan Zhu. 2010. *Function-based question classification*, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, MIT, Massachusetts, USA, pages 1119–1128.

George A. Miller. 1995. *WordNet: A Lexical Database for English,* Communications of the ACM Vol. 38, No. 11: 39-41.

K. Hacioglu and W. Ward. 2003. *Question classification with support vector machines and error correcting codes*, In Proceedings of NAACL/HLT-2003, Edmonton, Alberta, Canada, pages 28–30.

Kohavi R., and F. Provost. 1998. *On Applied Research in Machine Learning,* In Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, Columbia University, volume 30, New York.

Mitchell Tom M. 2002. 2nd edition, *Machine Learning*, McGraw-Hill, New York.

Moldovan, M. Pasca, S. Harabagiu, and M. Surdeanu. 2003. *Performance issues and error analysis in an open domain question answering system*, ACM Trans. Inf. Syst., 21(2):133–154.

Peters, M. Braschler, J. Gonzalo, and M. Kluck. 2002. *Advances in Cross-Language Information Retrieval*, Third Workshop of the Cross-Language Evaluation Forum (CLEF), Rome, Italy.

QA4MRE. 2013. *Question Answering for machine reading evaluation track of CLEF*, http://celct.fbk.eu/ResPubliQA/index.php?page=Pages/pastCampaigns.php, accessed on May 29, 2013.

Thamar Solorio, Manuel Perez, Manuel Montes-y-G´omez, Luis Villasenor-Pineda and Aurelio L´opez. 2004. *A Language Independent Method for Question Classification*, Proceedings of the 20th international conference on Computational Linguistics, Article No. 1374.

Voorhees. 2001. *Overview of the TREC 2001 question answering track*, In Proceedings of the 10th Text Retrieval Conference (TREC01), NIST, Gaithersburg, pages 157–165.

X. Li and D. Roth. 2002. *Learning question classifiers*. In Proceedings of the 19th International Conference on Computational Linguistics (COLING'02).

Zhang and W. Sun Lee. 2003. *Question classification using support vector machines*, In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 26–32, Toronto, Canada. ACM Press.

Zhiheng Huang, Marcus Thint and Zengchang Qin. 2008. Question *Classification using Head Words and their Hypernyms*, In Proceedings of Empirical Methods in Natural Language Processing, pages 927–936, Honolulu.