Charles University in Prague
Faculty of Mathematics and Physics

DIPLOMA THESIS

Đinh Lê Thành

Question and Answer Classifier for Closed Domain Interactive Question Answering

Institute of Formal and Applied Linguistics

Supervisor: RNDr. Markéta Lopatková, Ph.D.
Study program: Computer Science
Study specialization: Mathematical Linguistics

European Masters Program in Language and Communication Technologies (LCT)

2009

Free University of Bozen-Bolzano
Faculty of Computer Science

DIPLOMA THESIS

FREIE UNIVERSITÄT BOZEN
LIBERA UNIVERSITÀ DI BOLZANO
FREE UNIVERSITY OF BOZEN · BOLZANO

Đinh Lê Thành

Question and Answer Classifier for Closed Domain Interactive Question Answering

Faculty of Computer Science

Supervisor: Dr. Raffaella Bernardi
Study program: Computer Science

European Masters Program in Language and Communication Technologies (LCT)

2009

I hereby declare that this diploma thesis is my own work and where it draws on the work of others it is properly cited in the text.

I agree with a public availability of the work.

Prague 03, August, 2009 Đinh Lê Thành

ACKNOWLEDGEMENT

First and foremost, I would like to express my deep gratitude to Dr. Raffaella Bernardi, my main supervisor from Free University of Bozen-Bolzano, Italy. Discussing with her is always an interesting experience by which I has broadened my knowledge to new horizons and realized how prudent a researcher should be. Even when I was not living in Italy anymore, she was still willing to help me in my research by regularly spending time discussing with me. Especially, her trust, kindness and encouragement are forces which have driven me to keep looking forward to a better and better completion during the whole course of this work.

I am also very much indebted to Dr. Markéta Lopatková. Since the first days I have been in Prague when I was so new to everything, she has always been my responsible coordinator and tutor. Without her helps, it would have been very difficult for me in study and work. Though being late connected to my work, she still gave me valuable suggestions for the improvements. Her regularly encouragement and responsibility always raised me up in the progress of doing this work.

This thesis could not have been completed without practical information and materials. Without the opportunity to collect data material by myself, I had to rely completely on the help of Manuel Kirschner (PhD researcher) in Bolzano. It is hard for me to describe my sincere appreciation to his efforts to help me despite the difficulties arose during the process.

As the start for every good thing happened, I could not forget why I am here to pursue my study in prestigious European universities and to meet wonderful people. I would like to thank European Commission and the Master program of LCT for giving me this marvelous opportunity to receive a professional education system and to experience beautiful European cultures.

Last but not least, my deepest thanks are devoted to my beloved family and friends who are always by my side, encourage me, accept my mistakes, and make me feel proud whenever I have tried my best.

Title: Question and Answer Classifier for Closed Domain Interactive Question Answering

Author: Đinh Lê Thành

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Markéta Lopatková, Ph.D.

Supervisor's e-mail address: lopatkova@ufal.mff.cuni.cz

Nowadays natural language processing has made big progress thanks to the application of statistical approaches and to the large amount of data available to train the systems. These progresses are pushed by the several evaluation campaigns. Thanks to them systems are compared and progress measured. These evaluations are mostly based on data sets artificially developed by the organizers of such evaluation campaigns. In our work we show that though useful these data sets are biased and there is the need of developing data generated in a more natural setting by real users. We consider as case studies the classification of questions. In particular we look at the classification of questions types needed in Question Answering systems, and the classification of follow up questions into topic continuation and topic shift needed in Interactive Question Answering. We evaluate classifiers first on TREC data and than on a corpus of real user's data. In both cases the performance of the classifiers drops significantly showing the need of working on more users centered systems. The results also show that the classifiers could be better fine tuned taking into account the new challenges real users data launch to NLP systems. We leave this for future research.

Keywords: taxonomy, question classification, topic shift and topic continuation classification, follow-up.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BBC – British Broadcasting Corporation

CNN - Cable News Network

DT – Decision Tree

EAT – Expected Answer Type

IQA – Interactive Question Answering

IR – Information Retrieval

NB – Naïve Bayes

NIST – National Institute of Standards and Technology

NLP – Natural Language Processing

NN – Nearest Neighbors

POS – Part of Speech

QA – Question Answering

QC – Question Classification

SNoW – Sparse Network of Winnows

SVM – Support Vector Machines

TREC – Text REtrieval Conference

CHAPTER I

INTRODUCTION

In this first chapter, after introducing some backgrounds in the domain of problems, the main aspects of our work will be presented. We will explain the context from which the idea of the study derived as well as the problems chosen to be focused on will be explained in details. Next, the problems will be broken down into smaller research questions in order to clear the objectives of the thesis. After methodology applied to study the selected problem, the structure of this paper is outlined.

I.1. Introduction

Nowadays NLP has made big progress thanks to the application of statistical approaches and to the large amount of data available to train the systems. These progresses are pushed by the several evaluation campaigns. Thanks to them systems are compared and progress measured. These evaluations are mostly based on data sets artificially developed by the organizers of such evaluation campaigns. In our work we show that though useful these data sets are biased and there is the need of developing data generated in a more natural setting by real users. We consider as case studies the classification of questions. In particular we look at the classification of question types needed in Question Answering systems, and the classification of follow up questions into topic continuation and topic shift needed in Interactive Question Answering. We evaluate classifiers first on TREC data and than on a corpus of real user's data. In both cases the performance of the classifiers drops significantly showing the need of working on more users centered systems. The results also show that the classifiers could be better fine tuned taking into account the new challenges real users data launch to NLP systems. We leave this for future research.

I.1.1. Question classification

One of the most important processes in question answering is identifying the target of intension in the given question in order to find the type of expected answer. This process of determining the question (expected answer) type for a given question is usually called *question classification*. For example, the question of "Who was Galileo?" should be classified into the type of human (person). Without a question type, it would be much more difficult or even unrealistic to select correct answers from among the possible answer candidates, which could

be all of the nouns, noun phrases or named entities in the text collections. Thus, question classification provides a powerful restriction that helps reduce the number of answer candidates to a practical number that can be evaluated by the answer selection process. The correct prediction of the expected answer types has been shown as the key success of the whole question answering system (Li & Roth, 2002; Huang et al, 2008; Hacioglu & Ward, 2003; Zhang & Lee, 2003). If this question classification is successful, the system even might use different processing strategies (Harabagiu et al, 2001) to answer different types of questions.

The question classification task can be divided into three main sub tasks: (1) a design of taxonomy of expected answer types, (2) a corpus or collections of questions annotated with appropriate answer types (for machine learn approach only) and (3) an algorithm (either rule-based or machine learn based) that makes the prediction of the types of questions.

I.1.2. Topic shift and topic continuation classification

Question answering is an interactive human – machine process that attempts to give reasonable responses to user's questions in natural language in a form of brief and exact answers rather than full length or list of documents. However, most of the QA systems only deal with or are limited to questions in isolation. The reality is that users often ask questions naturally as a part of contextual interaction so that a sequence of questions has the same topic and particular constraints. For example, in a library domain question answering system, a question "Do you have literature books?" will be likely followed by other questions like "Where can I find them?" or "How do I borrow them?". Other situations in which users tend to ask a series of questions are when their needs are complex and it is hard for the users to satisfy with only one simple question or answer. Those questions could be too complicated, broad, or narrow that there would not be a good simple answer or there would be many answer candidates. In such cases, a clarification procedure would usually occur in order to constrain or narrow their search. Thus, contextual information should be taken into consideration and question answering systems that are able to answer contextual questions are more favored in the above mentioned cases.

An important challenge of integrating contextual information into question answering systems is to determine boundaries among topics. In other words, for each question the system needs to identify whether the question begins a new topic or it is a follow-up question of the current topic in the same dialogue. We define this task as *topic shift and topic continuation classification* and the term will be used throughout of this thesis. Another term, which is proposed by (Yang et al, 2006), is *relevancy recognition*; this term could be also understood as a

synonym for this term. Thus, in topic shift and topic continuation classification, if a question is identified as a follow-up (topic continuation) question, the system could make use of the context of the previous questions for further interpretations and to retrieve the answer. The task of topic shift and topic continuation classification is similar to *text segmentation* (Hearst, 1994) but it only focuses on the current question with the previous context while text segmentation has the full text available and is allowed to look ahead (Yang et al, 2006).

I.2. Research issues

The first aim of this thesis deals with question classification task, i.e. classification of questions according to their expected answer types. This target is of great interest since many previous researches have shown that correctly predicting the expected answer types of the questions holds the key role to the success of the whole question answering system. Most of these researches have evaluated their systems against TREC corpora and showed that the results are biased due to the fact that TREC data are designed to fit a specific purpose and manual selection has been done on the corpus. Some "interesting" questions intended to test the abilities of contesting systems are also added (Sunblad, 2007). However, testing on TREC still provides a good indication before going into the real user data and at least it could give some rough performance and could be compared with other systems. Therefore, the first issue of this work is:

Research issue 1: If the TREC corpus used in previous work on question classification is biased, then how well the performance can be expected on the real user's questions?

To answer this issue, we firstly re-examine and establish an "as good as possible" performance on the TREC corpus. This is done by investigating a combination of features used in the previous works and tested on the TREC corpus and taxonomy proposed by (Li & Roth, 2002). This is the corpus and taxonomy which has been widely used by many previous works in the field. The Naïve Bayes is then chosen as a classifier for testing the system because of its proven performance. Once we have finished experimenting with the above settings, the same features and classifier are used to test against the real user corpus in order to compare the results. In the next step, a new annotated corpus from real users is prepared. The corpus is built from the logs by running an online question answering system BOB which basically answers questions about the Bozen-Bolzano University and library. Since this corpus is collected in a much narrower domain and purpose, the new taxonomy is also needed. Finally, the result on this new corpus is judged to consider whether it could help the overall system.

The second aim of this thesis is similar to the first one but focus on the topic shift and topic continuation classification problem. By solving this task on the gold-standard TREC data, we also would like to see how well the classifier and features work on the real user data and whether they could be applied to real applications. Therefore, the second research issue is:

Research issue 2: Some research results have shown good results on the topic shift and topic continuation classification on TREC corpus but poorer performance on real user data (Yang et al, 2006); thus how biased the performance can be expected on our real users' questions?

We follow the same procedure as above by firstly re-examine the previous work on topic shift and topic continuation classification on the TREC data, similarly to the work of (Yang et al, 2006). The same classifier and features are then used for testing on the collected users' corpus. For this task, Decision Tree is chosen as the classifier since it showed highly performance on this problem (Yang et al, 2006). Finally, the results on both corpora are compared.

I.3. Research objectives

From the above mentioned research issues, the summarization of this thesis' work are the following:

- Re-examine the previous task on the question classification to set up the features and classifier for solving the problem.
- Compare the performance on the TREC and BOB data.
- Re-examine the previous task on the topic shift and topic continuation classification to set up the features and classifier for solving the problem.
- Compare the performance on the TREC and real users' data.

I.4. Thesis outline

Based on the research issues and objectives, the thesis is structured into 7 chapters as follows:

Chapter I

This current chapter provides the general information and background to the research problems. The research issues and objectives are defined and then the structure of the thesis is outlined.

Chapter II

This chapter focuses on question classification problem, including the introduction to the task, brief history and some previous works. It also contains the explanations of taxonomy and different approaches to solving the task.

Chapter III

This chapter focuses on topic shift and topic continuation classification problem. The introduction and some previous work are presented in details. The rule-based and machine learning based approaches to this task are also explained.

Chapter IV

This chapter describes in detail two machine learning classifiers used in this work, Naïve Bayes and decision tree. It also presents the method of evaluating classifiers and what measures are used for experiments.

Chapter V

This chapter provides the detailed description of methodology used to answer the first research issue about question classification. It contains the descriptions of the data, taxonomy, features, and the results of the conducted experiments.

Chapter VI

This chapter describes the methodology used for experiments of topic shift and topic continuation classification problem. It also explains about the data and features used. Lastly, it reports the results from the experiments.

Chapter VII

This chapter gives a summary of this work and some future directions for both problems.

CHAPTER II

QUESTION CLASSIFICATION

This chapter discusses in more details the problem of question classification. It begins with the definition, and then proceeds with the two main tasks of question classification, which are taxonomy designing and machinery methods for automatic classification. In the last section, it is given an overview of previous works.

II.1. Introduction

According to (Li & Roth, 2002), question classification can be loosely described as a task that, given a question (represented by a set of features), maps it to one of predefined k classes (expected answer types) which provide a semantic constraint on the sough-after answer. Another formal definition is adapted from text categorization (Sebastiani, 2002) to the problem of question classification: "Question classification is the task of assigning a boolean value to each pair $<q_j, c_i> \in Q \times C$, where Q is the domain of questions and $C = \{c_1, c,.., c_{|C|}\}$ is a set of predefined categories" (Sunblad, 2006).

The result of question classification is used in two downstream processes of question answering systems: *answer extraction* and *answer selection*. By that, it helps to select a correct answer from a large number of answer candidates extracted from the source corpora. That is to say, the result of the question classification, i.e. the expected answer types, can reduce the number of answer candidates. In particular, it is not necessary to scan and evaluate every noun phrase in the corpus to check whether it provides a correct answer. Expected answer types act as a filter and query, thus provide an efficient method of obtaining correct answers. Therefore, question classification is an important process of a question answering system. The success of question classification is supposed to result in a better performance of the question answering system.

II.2. Taxonomy

As mentioned in chapter I, the first important step in question classification is to design a taxonomy, in other words - the set of categories or of question (equivalently expected answer) types. Taxonomies are distinguished into two main types: flat and hierarchical taxonomies. While *flat taxonomies* have only one level of categories without having sub-categories, *hierarchical taxonomies* have

multi-level categories and some categories are sub-categories of the others. Table 1 presents an example of a flat taxonomy with 17 categories (Radev et al, 2002). For hierarchical taxonomies, a very common taxonomy is the one defined by (Li & Roth, 2002). There are 6 super (coarse) categories and 50 sub (fine) categories which belong to those super categories. The full taxonomy is presented in table 2.

PERSON	PLACE	DATE
NUMBER	DEFINITION	ORGANIZATION
DESCRIPTION	ABBREVIATION	KNOWNFOR
RATE	LENGTH	MONEY
REASON	DURATION	PURPOSE
NOMINAL	OTHER	

Table 1. Radev et al.'s flat taxonomy.

ABBREV.	description	country
abb	sport	mountain
exp	substance	other
ENTITY	symbol	state
animal	technique	**NUMERIC**
body	term	code
color	vehicle	count
creative	word	date
currency	**DESCRIPTION**	distance
dis. med.	definition	money
event	manner	order
food	reason	other
instrument	**HUMAN**	period
lang	group	percent
letter	individual	speed
other	title	temp
plant	description	size
product	**LOCATION**	weight
religion	city	

Table 2. Li & Roth's hierarchical taxonomy with coarse classes (in bold) are followed by their fine classes.

There are several approaches to construct expected answer types taxonomies. One possible solution is to take advantage of WordNet. (Fellbaum, 1998) used this approach and considered a subset of WordNet as the taxonomy. (Harabagiu et al, 2000) used a taxonomy in which some categories are connected to several word classes in WordNet. Another common approach is to manually analyze a corpus and derive a taxonomy from there. This approach is normally used for specific corpora and needs. And for the taxonomy used for real users QC-BOB corpus (described later), we also followed this approach.

Since the design of taxonomies is rather subjective and corpus specific, there is no standard taxonomy and numerous taxonomies have been defined. Even in the case of TREC QA-track when systems test on same corpora, most of them still define their own taxonomies. Some examples could be named here: (Li & Roth, 2002) defined a hierarchical taxonomy with 6 coarse and 50 fine grain classes, (Moldoval et al, 1999) designed a flat taxonomy with 15 categories or (Radev et al, 2002) made use of flat 17 categories taxonomy as mentioned above. In general, the design of taxonomy depends on the specificity required, the coverage of named entities over documents, the availability of training data and the required performance.

Since we work on a closed domain we have defined a taxonomy suitable for it. It is a flat one, and has been obtained by manually analyzing the corpus. It contains 14 categories obtained by analyzing the corpus manually. More details are provided in Chapter 5.

II.3. Approaches to question classification

Following taxonomy design, the next step in question classification is a machinery method for classifying the upcoming questions into the defined taxonomy. There are two main approaches for this task: hand-written rule-based and machine learning approaches.

Apparently, the most straightforward way to question classification is to use a set of predefined handwritten rules and heuristics. The rules could be just simple as, e.g., the questions starting with Who or Whom are classified as of type PERSON, ones starting with Where are classified as of type LOCATION, etc. However, the rules could also become very complicated using tagging, parsing or semantics. Some works adopted this approach are (Moldoval et al. 1999; Prager et al. 1999; Hermjakob, 2001; Radev et al. 2002). Researches show that rule-base approach has its own advantages and disadvantages such as:

- It enables maximum creativity and flexibility.
- Computation is usually cheap and fast.
- Require huge amount of tedious work and analysis of a large number of questions in order to infer appropriate rules.
- Has to correctly find various forms of each specific question to achieve reasonable accuracy and hence the number of rules could be very large to handle.
- Not sufficient to support fine grained classification.

For machine learning approach, it is firstly required an annotated corpus consisting of labeled questions. Then, a machine learning model is designed and trained on the annotated corpus. The model is assumed that useful patterns for later classification will be automatically captured from the corpus. Therefore, in this approach, the choice of features (for representing questions) and classifiers (for automatically assign questions into one or several categories of the taxonomy) are very important. Features may vary from simple surface of words or morphological ones to detailed syntactic and semantic features using linguistics analysis (Radev et al. 2002; Li & Roth 2002; Huang et al. 2008). Similarly, there are also various number of choices for classifiers, such as: Naïve Bayes (Zhang & Lee, 2003b; Sunblad, 2006), decision tree (Zhang & Lee, 2003b; Sunblad 2006), Sparse Network of Winnows (SNoW) (Sunblad, 2006; Li & Roth, 2002), Support Vector Machines (SVM) (Zhang & Lee, 2003b; Suzuki et al. 2003; Hacioglu & Ward, 2003; Sunblad, 2006) or Language Model based (Pinto et al. 2002).

Compare to rule-based approaches, machine learning gains some superior advantages that has made it more attractive to several researches recently:

- It only needs to define a relatively small number of "types" of features, which are then expanded in a data-driven way to a larger number of features.

- It is more flexible for machine learning systems to reconstruct than manual ones when changing the data or taxonomy because it can be trained on a new data or taxonomy in a very short time without re-writing the entire rules.

- When given more training data, the performance of machine learning systems usually improves.

Given the dominance of machine learning methods over the rule-based ones, we decided to apply this approach in order to solve the question classification problem.

II.4. Previous work on question classification

This section gives a short survey of some previous works on machine learning methods for question classification. This information is useful for better understanding the picture in the field, as well as to explain why we chose the data set, taxonomy and classifier for the experiments. The content of this section is mostly adapted from Sunblad's work (2006) along with some additions.

(Radev et al. 2002) follows the machine learning approach using decision tree classifier with set-valued features. This is a standard decision tree learner but has been adapted so that instead of being restricted to features with single values, the classifier can also handle features with set of values. The expected answer type taxonomy is flat and consists of 17 categories. The training data is TREC-8, TREC-9 and the testing data is TREC-10. There are 13 features for question classification in which 9 are semantic features derived from WordNet.

(Li & Roth, 2002) use SNoW as the classifier for classifying question types. This is a hierarchical classifier that makes use of two simple classifiers, coarse and file classifiers, each utilizing the Winnow algorithm within SNoW. The taxonomy is hierarchical and consists of 6 coarse and 50 fine semantic categories. The training data contains 5500 questions from different sources, i.e. manually constructed data and TREC-8, 9 data. The testing data has 500 questions from the TREC-10. There are 6 primitive feature types, including: words, POS tags, chunks, named entities, head chunks and semantic related words. These primitive features are then combined together by using operators to compose more complex features.

(Zhang & Lee, 2003) used the same taxonomy as well as the training and testing data as (Li & Roth, 2002). In their first experiment, they compared different machine learning approaches, including: Nearest Neighbors, Decision Tree, Naïve Bayes and SNoW. The features used for question representation were bag-of-words and bag-of-ngrams. In the second experiment, they made use of a SVM classifier but the default linear kernel of the SVM was replaced by their proposed tree kernel. The features were binary feature vectors.

(Suzuki et al. 2003b) also used a SVM classifier with a replaced kernel. The developed kernel was hierarchical directed acyclic graph kernel. The firstly proposed taxonomy consisted of 150 categories, but later after removing the categories with too few samples (less than 10), it contained only 68 categories. The training and testing data were in Japanese and consisted of 1011 questions from NTCIR-QAC, 2000 questions of CLR-QA data, and 2000 other questions reported to be TREC style.

(Hacioglu & Ward, 2003) used a default linear kernel SVM with error correcting codes to convert the multi-class classification problem into a number of binary ones. It firstly split the multi-class data into m binary class data, then m SVM classifier can be designed and their output combined. The taxonomy as well as the training and testing data were used the same as (Li & Roth, 2002). The features used for experiments were bag-of-ngrams and named entities.

(Zaanen et al. 2005) combined machine learning technique to extract patterns and regular expression rules to classify questions. There were two developed systems which were Alignment-Based Learning (ABL) and Trie Classifier. The taxonomy and the corpus data for training and testing were also the same as (Li & Roth, 2002).

(Huang et al. 2008) experimented with a linear kernel SVM and a Maximum Entropy model as the classifiers for question classification. The taxonomy and training and testing data were the same with (Li & Roth, 2002). The features set used was compact, including: question wh-word, head word, WordNet semantic features for head word, word grams and word shape.

CHAPTER III

TOPIC FOLLOW-UP CLASSIFICATION

This chapter gives a more detailed explanation of the problem of topic shift and topic continuation classification. It begins with an introduction and definition. Next, two main approaches for solving this task are discussed, namely rule-based and machine learning methods. The last section describes some related work in the field.

III.1. Introduction

As mentioned in chapter I, it is widely acknowledged that answering follow-up questions (a question asked after another one) is a different task than answering isolated questions. Therefore, Interactive Question Answering systems have to tackle different problems than QA systems, in addition to the traditional ones. One challenge is to determine whether or not a question is related to the previous interaction context. In other words, it is to determine the boundaries between topics. We refer to this task as topic shift and topic continuation classification or relevancy recognition as an equivalent term by (Yang et al, 2006).

Topic shift and topic continuation classification is the first step in contextual question answering. In the next step, the results of this procedure are used to take into account information from the context information in order to interpret and retrieve the answers. In particular, if a question is recognized as a topic continuation, the context information, which is derived from the previous questions, could be integrated into the query construction module. For example, instead of using only keywords from the current question to formulate the query to retrieve a set of answer candidates, we could add the topic words, the topic noun phrases or the topic pronouns to improve the relevant answer set. It has been shown that this context information can improve the performance of document retrieval (Yang et al, 2006).

III.2. Approaches to topic shift and topic continuation classification

Similarly to the question classification problem, there are two main approaches to topic shift and topic continuation classification, namely rule-based and machine learning based approaches. In the former method, the hand written rules are manually deduced by carefully analyzing the corpus. For example, (De

Boni & Manandhar, 2005) defined a set of rules, such as: if a question has no verb, then it is a follow-up question or if a question has pronouns and possessive adjectives, it is also a follow-up in the dialogue. For reasons similar to those discussed in the previous chapter, the disadvantage of this method is that it requires a great deal of human effort to research and analyze a specific data set and design the rules. If there is a need to work on a new corpus from different domain, it is much likely that one would have to repeat almost the same amount of work to go over the corpus and modifies the existing rules or adds more rules. This is again time and human effort consuming.

Alternatively, a recently interested approach is to pursue a data driven (machine learning) approaches to automatically learn the rules and patterns from a data set. This approach requires much less human effort on analyzing a specific data set and on summarizing rules from the observation. In fact, one only needs to investigate a set of features which can be also automatically extracted. Then, it is straightforward to train a model on the corpus. When a new corpus from a different domain comes, another model is trained based on the previous set of features without a huge effort in exploiting the new corpus. Moreover, the machine learning approaches also have shown also a better performance than the rule-based ones, as reported by (Yang et al, 2006). Hence, with those mentioned reasons, we have decided to apply the machine learning method to tackle the topic shift and topic continuation problem.

III.3. Previous work on topic shift and topic continuation classification

This section gives a brief summary of two previous works on the topic shift and topic continuation classification problem. The first work of (De Boni & Manandhar, 2005) deals with a rule-based approach while the second work of (Yang et al, 2006) follows a machine learning method. Some materials from (De Boni & Manandhar, 2005) work have been applied for extracting semantic features of this thesis experiments and most of (Yang et al, 2006) work has been re-implemented as the re-examination experiment of the topic shift and topic continuation classification problem.

In their research, De Boni & Manandhar has developed a rule-based algorithm for topic shift and topic continuation classification. They manually deduced a set of rules by carefully analyzing the TREC 2001 corpus. They report the observations below about the following cues to recognize follow-up (here called topic continuation):

- Pronouns and possessive adjectives: if a question has a pronoun that does not refer to the same entity in the same question, this is considered as a follow-up question.

- Cue words: a question consists of words like “precisely”, “exactly”, etc.

- Ellipsis: if a question is syntactically incomplete, for example it contains no verb phrase, it might be a follow-up question.

- Semantic Similarity: if a question has certain semantic similarity to previous questions, this might be a follow-up question. Their proposed algorithm for calculating the semantic similarity between the current question Q={w_1, w_2, .., w_n} and previous question Q’={w'_1, w'_2, .., w'_m} follows:

$$SentenceSimilarity(Q, Q') = \frac{1}{n} \sum_{1 \le j \le n} \max_{1 \le i \le m} WordSimilarity(w_j, w'_i)$$

  The value of WordSimilarity(w, w’) is the similarity between two words, calculated from WordNet (Fellbaum, 1998).

Motivated by these cues, (De Boni & Manandhar, 2005) proposed a rule-based algorithm for topic shift and topic continuation classification. Basically, a question is classified as a follow-up if it (1) contains references to previous questions; (2) contains cue words; (3) contains no verbs; or (4) bears certain semantic similarity to previous question or answer. The recall of the algorithm is 90% for recognizing topic shift questions and 78% for follow-up questions. The precision is 56% and 96%, respectively. The overall accuracy is 81%.

More recently, (Yang et al, 2006) followed a data-driven (machine learning) approach to deal with the problem of topic shift and topic continuation classification and evaluated algorithm on two data sets: the TREC data and the one containing real users interaction with a machine, HandQA data. They made use of decision tree method as the classifier. Inspired by (De Boni & Manandhar, 2005), they also selected two types of features: morphologic/syntactic and semantic features. Syntactic features capture whether a question has certain morphologic/syntactic components, such as: pronouns, verbs, proper noun, etc. Semantic features characterize the semantic similarity between the current question and previous questions in the same dialogues. The formulas used for computing the semantic features are also modified from the formula proposed by (De Boni & Manandhar, 2005) but with normalization. Evaluated on the TREC data (TREC 2004 for training and TREC 2001 for testing), the recall is 90% for recognizing topic shift questions and 94% for follow-up questions. The precision is 82% and 97%, respectively. The overall accuracy is 93%. On the real life HandQA data, the recall is 73% and 75% and the precision is 62% and 84%. The overall accuracy is 74%.

CHAPTER IV

CLASSIFIERS

This chapter presents some basic theoretical background about two classical yet effective classifiers: decision tree and Naïve Bayes. These two algorithms have been chosen as classifiers for experimenting with our problems of question classification and topic shift and topic continuation classification. The first section explains the model of decision tree along with some examples. Next, the second section starts with Bayes theorem and then Naïve Bayes model. The pros and cons of both algorithms are also mentioned in both sections. The content is mostly summarized from the book (Mitchell, 1997) where one can find more details about these algorithms and other methods. In the last section, the chapter concludes with details on how the results of experiments are evaluated and what measures are used.

IV.1. Decision tree

Decision tree is an inductive inference algorithm and approximates the target function. It can also be considered as a set of *if-then* rules based on the feature values of the data. The classification is therefore obtained by traversing the tree starting from the root node to its leaves. Each node corresponds to an attribute and each edge corresponds to a value of an attribute. A leaf determines a classification of an instance and hence the target function is discrete. This algorithm is proven robust to the noisy data. Moreover, it does not need to contain all the attribute values hence it can handle disjunction of conjunctions of attributes.

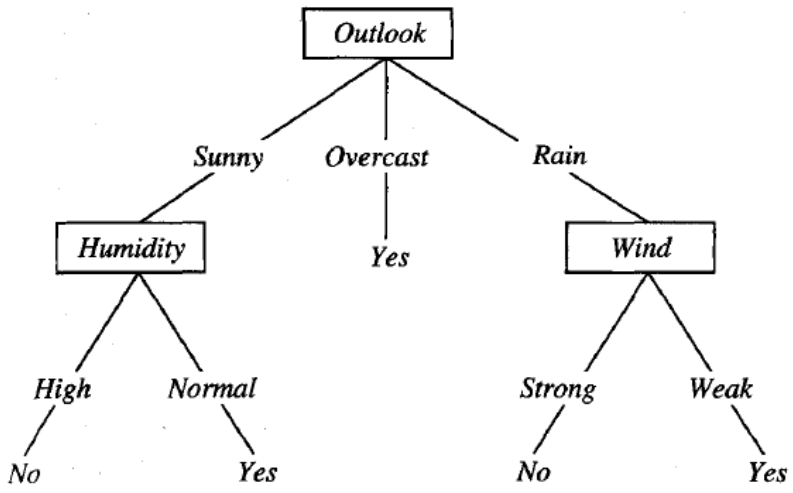

Figure 1. An example of decision tree.

The basic decision tree algorithm is started by building the root node which corresponds to all the training examples. Then following the "top-down" procedure, the children are added according to the attributes of the training data. A key question in the algorithm is "which attribute is the best choice for a given node". Depending on existing algorithms, several techniques are used among which the most popular ones are "Gini impurity" (CART algorithm) and "Information Gain" (ID3, C4.5, C5.0).

Gini impurity is based on squared probabilities of membership for each target category in the node. It reaches its minimum (zero) when all cases in the node fall into a single target category.

Suppose y takes on values in {1, 2,.., m}, and let f(i, j) = probability of getting value j in node i. That is, f(i, j) is the proportion of records assigned to node i for which y = j.

$$I_G(i) = 1 - \sum_{j=1}^{m} f(i, j)^2 = \sum_{j \neq k} f(i, j)\, f(i, k)$$

Information gain is based on the concept of entropy used in information theory.

$$Gain(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v)$$

where

$D = \text{training data}$

$A = \text{attribute}$

$D_v = \{d \in D; A(d) = v\}$

$Values(A) = \text{set of all possible values of attribute A}$

$Entropy(D) = \sum -p_i \log_2 p_i$

Following is an example of using the "Information gain" to decide which attribute will be chosen in the next step to build the decision tree (Mitchell, 1997):

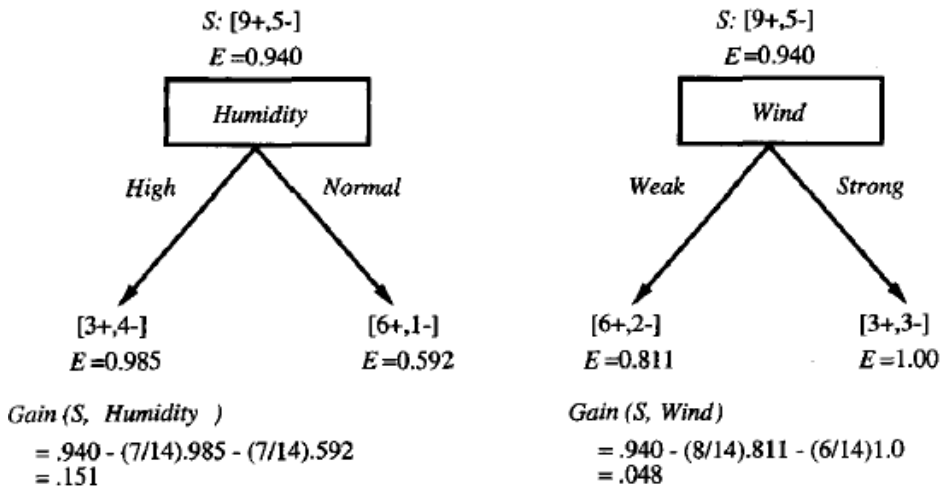

Figure 2. Humidity provides greater information gain than Wind, relative to the target function. Hence, humidity will be chosen as the next attribute in the tree.

There are some advantages of using decision tree for classification:

- Simple to understand and interpret.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- Able to handle both numerical and categorical data.
- Use a white box model. If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Robust, perform well with large data in a short time. Large amounts of data can be analyzed using personal computers in a time short enough to enable stakeholders to take decisions based on its analysis.

IV.2. Naïve Bayes

Naïve Bayes is also an inductive inference algorithm and it is suitable for tasks when each instance x is represented as a conjunction of attribute values and the target function f(x) can take any value from a finite set V. A set of training examples along with the target function are provided and a new instance is presented by a tuple of attribute values $<a_1, a_2,.., a_n>$. The learner is then asked to classify this new instance.

The Naïve Bayes approach, it will assign the new instance to the most probable target value, v_{MAP} as following:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, .., a_n)$$

By using Bayes theorem, we can rewrite this as:

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, .., a_n \mid v_j) P(v_j)}{P(a_1, a_2, .., a_n)}$$

Since $P(a_1,a_2,..,a_n)$ does not depend on v_j:

$$v_{MAP} \sim \arg\max_{v_j \in V} P(a_1, a_2, .., a_n \mid v_j) P(v_j)$$

It is easy to estimate $P(v_j)$ simply by counting the frequency of v_j in the training data. However, estimating $P(a_1, a_2,.., a_n)$ in this fashion is not feasible unless having the very large set of training data.

Therefore, the Naïve Bayes classifier deals with this problem by simply assuming that the attribute values are conditionally independent given the target value. In other words, the assumption is that, the probability of observing the conjunction $<a_1, a_2,.., a_n>$ is just the product of the probabilities of the individual attributes: $P(a_1, a_2,.., a_n|v_j) = \prod P(a_i|v_j)$. Substituting this into the above equation, we have the Naïve Bayes classifer:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

The probabilities in the Naïve Bayes model could be also easily estimated by counting the frequency (log likelihood) as mentioned above. However, this may raise two difficulties. First, it produces a biased underestimation of the probability. Second, when this probability estimation is zero, this will dominate the whole model even for some non-zero probabilities in the future. To overcome this problem, we can adopt a Bayesian approach to estimating the probability, using m-estimate (Laplace) as follows:

$$\frac{n_c + mp}{n + m}$$

where n corresponds to the total number of times v_j appears in the data and n_c is the number of times of v_j in the presence of the attribute value a_i, p is to assume uniform priors ; that is if an attribute has k possible values, then p=1/k (Mitchell, 1997).

Advantages of using Naïve Bayes classifer are:

- The Naive Bayes algorithm could achieve fast, highly scalable model for training and testing.

- It scales linearly with the number of predictors and instances.

- It requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

- Naive Bayes can be used for both binary and multiclass classification problems.

IV.3. Evaluation measures

This section describes the measures used for evaluating the performance of the classifiers and experiments. It firstly shows the basic methods of information retrieval in general and classification problems in particular, such as: true false positives and negatives, precision, recall, F-measure.

Classification effectiveness is usually measured by means of the classic IR notions of precision and recall which can be adapted to the cases of question classification and topic shift and topic continuation classification. These measures may be estimated from a contingency table for category c_i as shown below.

Category c_i		Human judgments	
		Yes	No
Classifier judgments	Yes	TP_i	FP_i
	No	FN_i	TN_i

Table 3. Contingency table for category c_i.

Hence, to estimate precision and recall, we firstly need to define the following notions:

True positive

If a classifier correctly assign a question q_j to a category c_i, and a human expert also assigned q_j as belonging to c_i, this classification is referred as a true positive (TP_i).

False positive

If a classifier incorrectly assigns a question q_j to a category c_i, while a human expert rejected this q_j as belonging to c_i, this classification is referred as a false positive (FP_i).

True negative

If a classifier correctly rejects a question q_j to a category c_i, and a human expert also rejects q_j as belonging to c_i, this classification is referred as a true negative (TN_i).

False negative

If a classifier incorrectly rejects a question q_j to a category c_i, while a human expert assigned this q_j as belonging to c_i, this classification is referred as a false negative (FN_i).

Hence, the precision and recall with respect to the category c_i are calculated as follows:

$$precision_i = \frac{TP_i}{TP_i + FP_i}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i}$$

We then can compute the macro-averaged precision and recall over all categories as follows:

$$\overline{precision} = \frac{\sum_{i=1}^{|C|} precision_i}{|C|}$$

$$\overline{recall} = \frac{\sum_{i=1}^{|C|} recall_i}{|C|}$$

Usually, a system trying to reach a high precision will have to lower its recall and vice versa. This leads to a measure that combines these two measures as a weighted harmonic mean, the F-measure:

$$F = (1+\beta^2)\frac{precision.recall}{\beta^2.precision + recall}$$

F-measure balances precision and recall by means of a weight β. When β=1, precision and recall have equal weight, this is a special case of F-measure, namely F_1 measure:

$$F = \frac{2.precision.recall}{precision + recall}$$

Two other commonly used F-measure are the F_2 measure, which weights recall twice as much as precision, and $F_{0.5}$ measure, which weights precision twice as much as recall.

CHAPTER V

QUESTION CLASSIFICATION: EXPERIMENTS AND RESULTS

The chapter describes the results on the first research issue mentioned in Chapter I. First, the methodology for the problem of question classification is explained, followed by the descriptions of the TREC, BOB corpus, taxonomy as well as the features used for experiments. Next, the results of question classification are presented, including two main experiments on LR-TREC and QC-BOB. We conclude the chapter with some discussions about the results.

V.1. Methodology

This section explains the methodology used for the question classification problem. Re-stating the research issue in the chapter I, we want to answer the following question: ***Since the TREC corpus used in previous work on question classification seems to be biased (Sunblad, 2006), then how well the performance can be expected on the real users' questions?***

In order to address the issue, a series of experiments have been conducted. First, we will establish a baseline to which we can compare the results. This baseline is established by re-examining the performance of the classifier on the corpus and taxonomy used by (Li & Roth, 2002). We also investigated features of questions which are necessary for classifiers. The work then was continued with a new annotated corpus (BOB) collected from real users, based on the same set of features and classifier as in the baseline. However, the taxonomy has to be modified and adapted to the new domain because the BOB domain is much narrower than the open domain of the TREC corpus. A machine classifier is then tested on this new corpus, and the results are compared to the results from baseline experiments. For present purposes, the Naïve Bayes algorithm is chosen as the classifier. This classifier has been widely used in previous work and it is also quite easy to implement.

To sum up, the work for this section is consisted of:

- Investigate a set of features used for experiments.
- Implement a Naïve Bayes classifier.
- Run the Naïve Bayes classifier using the above investigated features on the corpus and taxonomy established by (Li & Roth, 2002) in order to setup the baseline measure.
- Design a taxonomy for BOB corpus.

- Annotate a corpus from the BOB system logs using the proposed taxonomy.

- Run the Naïve Bayes classifier using the same set of features on the BOB corpus and taxonomy, compare the results.

V.2. Data descriptions

As previously mentioned, there are two corpus used for experiments of question classification. One such corpus is widely used in many previous works (Li & Roth, 2002; Zhang & Lee, 2003; Hacioglu & Ward 2003; Zaanen et al. 2005; Huang et al. 2008) for evaluating question answering systems. It was proposed by (Li & Roth, 2002) and could be named here as LR-TREC. Another corpus is collected from BOB logs. In this section, we will describe those corpora and how they are used for experiments.

LR-TREC corpus

This corpus[1] is collected from 4 sources: 4500 English questions published by USC (Hovy et al. 2001), about 500 questions manually constructed for balancing the corpus and 894 questions from TREC 8 and TREC 9. Those ~5900 questions serve as the training set. The 500 questions from TREC 10 are used as the test set. All those questions are already manually annotated according to Li & Roth taxonomy which is described later. For some technical purposes, the classes' names format is modified and slightly differs from the original data but this does not affect the content of the classes as well as the questions. Here are a few examples taken from LR-TREC corpus:

HUM__ Who is the current UN Secretary General ?
HUM__ Who provides telephone service in Orange County , California ?
LOC__ Where did guinea pigs originate ?
LOC__ What two countries ' coastlines border the Bay of Biscay ?
DESC__ Why did Egyptians shave their eyebrows ?
HUM__ What organization did Mr. Waverly assign agents for ?
NUM__ When was the slinky invented ?
HUM__ What are the characters ' names in the Scooby-Doo cartoon ?
ENTY__ What is a fear of cold ?
HUM__ What was Eduard Shevardnadze 's job in the Soviet Union ?
DESC__ What causes rust ?
ABBR__ What is the full form of .com ?

Figure 3. Sample questions from LR-TREC corpus annotated with 6 coarse classes.

[1] http://l2r.cs.uiuc.edu/~cogcomp/data.php

BOB-QC corpus

BOB is a chatterbot library system[2] that helps users answer the questions and finds information around topics like infrastructure and organization of the University Library, lending, ordering, borrowing, reserving, picking up, returning, renewing books etc. More details about the system could be found at the project webpage[2]. The questions from users are collected into log files and after refining the garbage, there are about 2500 raw questions. Then we manually annotated the data set using the proposed taxonomy which is described in the next section. This set is served as both training and testing set by splitting techniques. Some sample questions taken from the BOB questions set are as follows:

LOCATION__ where can i access the internet?
YESNO__ do you have payphones in the library?
NUMERIC__ how many books does the library possess?
ENTITY__ which topics does the library collection cover?
INSTRUCTION__ How can I browse through the movie collection of the library?
DATE__ when is the birthday of the library?
PERSON__ who is the director of the library?
MONEY__ what are the tuition fees for the faculty of economics?
DEFINITION__ what does the tag at disposal mean?

Figure 4. Sample questions from BOB corpus annotated with 14 classes

V.3. Taxonomy

This section will describe in depth the taxonomy of (Li & Roth, 2002), which is used for testing on the LR-TREC corpus, and the taxonomy used for BOB corpus.

Li & Roth taxonomy

Li & Roth defined a taxonomy of two layers which represents a natural semantics for typical answers in the TREC task. The coarse layer consists of 6 classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION, NUMERIC) and the finer layer consists of 50 classes. The full taxonomy is given in the table 3. The table also shows the distributions of these classes in 500 questions of TREC 10. Each coarse class contains a non-overlapping set of fine classes.

[2] http://www.unibz.it/EN/LIBRARY/ABOUT/PROJECTS/bob-project.html

Class	#	Description	Class	#	Description
ABBREV.	9	abbreviation	description	7	Description of smth.
abb	1	abbreviation	manner	2	Manner of an action
exp	8	expression abbreviated	reason	6	Reasons
ENTITY	94	Entities	**HUMAN**	65	Human beings
animal	16	Animals	group	6	A group or organization of people
body	2	Organs of body	individual	55	An individual
color	10	Colors	title	1	Title of a person
creative	0	Inventions, books and other creative pieces.	description	3	Description of a person
currency	6	Currency names	**LOCATION**	81	Locations
dis. med.	2	Diseases and medicine	city	18	Cities
event	2	Events	country	3	Countries
food	4	Food	mountain	3	Mountains
instrument	1	Musical instrument	other	50	Other locations
lang	2	Languages	state	7	States
letter	0	Letters like a-z	**NUMERIC**	113	Numeric values
other	12	Other entities	code	0	Postcodes or other codes
plant	5	Plants	count	9	Number of smth.
product	4	Products	date	47	Dates
religion	0	Religions	distance	16	Linear measures
sport	1	Sports	money	3	Prices
substance	15	Elements and substances	order	0	Ranks
symbol	0	Symbols and signs	other	12	Other numbers
technique	1	Techniques and methods	period	8	The lasting time of smth.
term	7	Equivalent terms	percent	3	Fractions
vehicle	4	Vehicles	speed	6	Speed
word	0	Words with a special property	temp	5	Temperature
DESCRIPTION	138	Description and abstract concepts	size	0	Size, area and volume
definition	123	Definition of smth.	weight	4	Weight

Table 4. Li & Roth's taxonomy and the distribution of 500 TREC 10 questions over the taxonomy. Coarse classes (in bold) are followed by their fine classes.

BOB taxonomy

Since the taxonomy proposed by Li & Roth applied only for factoid questions and open domain data (i.e TREC), it is not suitable for classifying questions of BOB corpus. Beside the factoid questions like when, where why,.., the BOB corpus also contains many yes-no questions and even non-questions such as: "Show me the plan of the university's buildings", "university's buildings" or very often just a chat message like "we are colleagues". Moreover, the BOB questions mainly focus on topics just around the university and library. Therefore, there is a need to design a new taxonomy which adapts these differences. Taking the inspiration from Li & Roth's taxonomy in addition with the analysis of the current and prospective BOB corpus, we re-design the new taxonomy for BOB corpus . It is a flat taxonomy and has 14 classes (YESNO, INSTRUCTION, CHAT, PERSON, REASON, DEFINITION, LOCATION, ENTITY, DATE, MONEY, DISTANCE, PERIOD, NUMERIC, QUALITY). The below table 4 shows the distributions of these classes over 2469 questions of the BOB corpus.

Class	#	Description
YESNO	579	Yes-no questions
INSTRUCTION	552	Questions or sentences related to helps and instructions
CHAT	624	Chat, general talks, non-questions
PERSON	42	Names of persons
REASON	17	Entities
DEFINITION	166	Locations
LOCATION	176	Dates
ENTITY	139	An individual
DATE	26	Dates
MONEY	48	Prices
DISTANCE	0	Linear measures
PERIOD	21	The lasting time of smth.
NUMERIC	76	Count numeric
QUALITY	3	Quality measures

Table 5. BOB taxonomy and the distributions of 2469 questions over the taxonomy.

V.4. Feature extraction

In any common classification task, features are the key to obtain an accurate classifier. As a consequence, if the set of features is flawed, then it does not matter which algorithm or classifier is used (Rendell & Cho, 1990, Mitchell, 1997). In the question classification task, many research have been experimented with different number and types of features. Some can define a rich set of features, e.g Li and Roth's who made use of hundreds of features, ranging from simple features like bag of words, POS, etc. to semantic ones. In contrast, recently, there are also some results with an only compact list of features, such as (Huang et al, 2008) with only 5 types of features: Wh word, head word, WordNet semantic feature, n-grams and word shape. Interestingly, this outperforms the best previously reported accuracy of 86.2% by the accuracy of 89% (Huang et al, 2008). Inspired by this work, a set of 7 types of features is defined as the representation of each question: Wh-word, n-grams, Part of Speech of the n-grams, first noun, first verb, head word and Wordnet semantic

feature. The further discussions of each feature are described in the next sections.

Wh-word The wh-word feature is simply the question wh-word in given question. For example, the wh-word of the question "Where is the library" is "where". There are 10 types of wh-word have been defined: what, which, when, where, who, why, how, whom, yesno and none. The wh-word YESNO is doubtful questions like "Do you know about the history of the university?" and NONE for non-question questions like "Tell me the place of art department" or just "computer".

Unigrams Many researches showed that simple n-grams features are very informative (Li & Roth, 2002; Huang et al, 2008). The first two unigrams after the wh-word are chosen as the features. For example, the two unigrams of the question "Which floor is the library?" are "floor" and "is".

POS of unigrams In addition to unigram features, POS of unigrams could be useful when capturing more general patterns than unigrams do. For example, for the cases of two unigrams like "can, it" or "may, I", the pattern of "MD, PP" for POS features is extracted. For extracting the POS features, the TreeTagger utility was used.

First noun In many cases, especially in the BOB data, the first nouns tend to be very useful feature. Hence, this is chosen as a feature for question representation. For example in the above example, the first noun will be "floor".

First verb The first verb is extracted in a similar way as the first noun. For example in the above example, the first verb will be "is".

Head word (Li & Roth, 2002) and (Krishnan et al. 2005) used head chunks as their features. In both approaches, Huang showed that noises could be introduced (Huang et al, 2008). To tackle the problem, Huang made use of the head noun feature which is exactly one single word in the noun chunk. For example, in the question "Where are the literature books?" the head noun will be "books" instead of "literature books". To obtain the head noun feature, a syntactic tree of the question is needed. The detailed algorithm for extracting the head noun is described in (Huang et al, 2008)

In the scope of this thesis, instead of extracting the head noun using the syntactic tree and the rules as in (Huang et al, 2008), we have limited ourselves to use only the rules set from the Huang's algorithm along with the first noun to achieve the head word.

Wordnet semantic feature There were several works made use of the package WordNet::Similarity for extracting semantic features. The idea of WordNet::Similarity is to model the length of path traveling from one word to another in the WordNet network. The semantic similarity based on the path is then computed. There are several methods of computing this measure, such as: Leacock & Chodorow, Wu & Palmer, Resnik,...(Pedersen et al, 2004). Based on the WordNet::Similarity package, the similarity between the first noun feature and

each description of expected answer types in the taxonomy is computed. The EAT with the description that has the highest similarity with the head word will be marked as the feature. For example, as the head word "price" of the question "What is the price of internet in the university?" has the highest similarity to the description of "MONEY" which is also "price", then the semantic feature for this question is marked as "MONEY".

Below is an example of question representation for the sentence "PERIOD__ for how long may I remain in the library?"

Wh-word	1st unigram	2nd unigram	1st unigram' POS	2nd unigram's POS	1st noun	1st verb	Head word	WordNet similarity
how	long	may	RB	MD	library	remain	long	PERIOD

Table 6: An example of question representation.

V.5. Results on LR-TREC data

The first experiment is an examination of how the classifier performs in order to setup the baseline and features settings. As described in chapter V, we have used the corpus and taxonomy originally developed by (Li & Roth, 2002). The testing set contains 500 questions from TREC-10 and the training set is about 6000 questions collected from several different sources. Questions were represented by 9 features which are: Wh-word, first word, second word, first word's POS, second word's POS, first noun, first verb, head word and WordNet semantic similarity. The chosen classifier is Naïve Bayes and default Laplace smoothing parameter is 10. It is implemented in Perl and could be found in more details in the appendix. This experiment has been done under two different sets of taxonomy: coarse (6 classes) and fine-grain (50 classes).

The performance along with the confusion matrix of the learner in the first setting can be found in table 7 and 8 results from the second setting are found in table 9.

	Predicted class					
True class	NUM	ABBR	DESC	ENTY	HUM	LOC
NUM	**92**	0	10	1	5	5
ABBR	0	**8**	1	0	0	0
DESC	2	6	**122**	6	0	2
ENTY	2	0	15	**61**	7	9
HUM	0	0	1	3	**61**	0
LOC	1	0	6	3	0	**71**

Table 7. Confusion matrix on LR-TREC data using 6 coarse classes taxonomy.

Precision	**Recall**	**F-Measure**	**Class**
0.948	0.814	0.876	NUM
0.787	0.884	0.833	DESC
0.571	0.889	0.695	ABBR
0.824	0.649	0.726	ENTY
0.816	0.877	0.845	LOC
0.836	0.938	0.884	HUM
0.797	**0.842**	**0.819**	

Table 8. Performance results on LR-TREC data using 6 coarse classes taxonomy

In the confusion matrix table, the diagonal values correspond to the number of correctly classified questions with respect to the row category and are highlighted in boldface. The macro averaged precision, recall and F_1-measure are 0.797, 0.842 and 0.819, respectively. ABBR and ENTY classes have the least proportions of correctly predicted questions and brought down the performance of the overall result.

Precision	Recall	F-Measure	Class
0.862	0.909	0.885	HUM_ind
0.667	0.250	0.364	ENTY_animal
0.889	0.800	0.842	LOC_other
1.000	1.000	1.000	HUM_desc
0.250	0.333	0.286	NUM_money
0.000	0.000	0.000	ENTY_dismed
0.833	0.833	0.833	LOC_city
0.000	0.000	0.000	NUM_code
0.667	1.000	0.800	ENTY_lang
0.000	0.000	0.000	ENTY_cremat
0.000	0.000	0.000	ENTY_event
1.000	1.000	1.000	NUM_speed
0.500	0.250	0.333	ENTY_veh
0.250	1.000	0.400	ENTY_instru
1.000	0.400	0.571	NUM_temp
0.700	0.875	0.778	NUM_period
0.429	1.000	0.600	LOC_country
0.417	0.833	0.556	DESC_reason
0.000	0.000	0.000	ENTY_body
0.500	1.000	0.667	HUM_title
0.500	0.250	0.333	ENTY_product
0.500	0.750	0.600	ABBR_exp
0.600	0.750	0.667	ENTY_food
0.500	0.571	0.533	DESC_desc
1.000	1.000	1.000	DESC_manner
0.000	0.000	0.000	ENTY_sport
1.000	0.563	0.720	NUM_dist
0.467	1.000	0.637	LOC_state
1.000	0.417	0.589	NUM_other
0.900	0.900	0.900	ENTY_color
1.000	0.667	0.800	NUM_perc
1.000	1.000	1.000	ABBR_abb
0.333	0.500	0.400	HUM_gr
0.333	0.167	0.222	ENTY_other
1.000	0.889	0.941	NUM_count
1.000	0.667	0.800	ENTY_currency
0.778	0.467	0.584	ENTY_substance
1.000	0.500	0.667	NUM_weight
0.849	0.870	0.859	DESC_def
0.375	0.857	0.522	ENTY_termeq
1.000	0.200	0.333	ENTY_plant
0.500	1.000	0.667	ENTY_techmeth
0.904	1.000	0.950	NUM_date
0.500	0.667	0.572	LOC_mount
0.614	**0.617**	**0.615**	

Table 9. Performance results on LR-TREC data using 50 fine-grain classes taxonomy.

In the above results, we skip showing the confusion matrix since it is too large and sparse, hence is difficult to follow. Looking at the table 9 from the experiments on the fine-grain taxonomy, we see that the overall performance decreased from 81% into about 61% in which some categories have no correct classifications. These categories, i.e ENTY_dismed, NUM_code, ENTY_cremat, ENTY_event, ENTY_body, ENTY_sport, have only frequency of 1 or 2 thus they have very high chance of unrecognizing by the classifier. Consequently, they lowered down the overall performance.

V.6. Results on QC-BOB data

The second experiment is to investigate the performance of the same classifier and features on the corpus consisting of real life user's questions, QC-BOB. In this experiment, 2469 questions collected and refined from BOB system logs were used and the new taxonomy is proposed (see chapter V for details). The data serve as both training and testing and 10 folds cross-validation was used for splitting the data. Following that, the corpus is repeatedly partitioned into 10 disjoint subsets of equal sizes. It then learns and tests the algorithm 10 times, using each partitioning in turn as the testing set, and using the remaining data as the training set. In this way, the learning algorithm is tested on 10 independent test sets, and the mean of measures are returned as the measure for the algorithm. The results can be found in table 10.

Precision	Recall	F-Measure	Class (All)
0.699	0.718	0.708	ENTITY
0.583	0.627	0.604	REASON
0.870	0.937	0.902	LOCATION
0.767	0.797	0.782	PERSON
0.567	0.507	0.535	DATE
0.870	0.851	0.860	NUMERIC
0.785	0.847	0.815	INSTRUCTION
0.893	0.788	0.837	YESNO
0.715	0.922	0.805	DEFINITION
0.974	0.885	0.927	CHAT
0.630	0.642	0.636	PERIOD
0.781	0.850	0.814	MONEY
0.761	**0.781**	**0.771**	

Table 10: Performance results using 10 folds cross-validation on QC-BOB data with all classes

Precision	Recall	F-Measure	Class (Factoid)
0.699	0.718	0.708	ENTITY
0.583	0.627	0.604	REASON
0.870	0.937	0.902	LOCATION
0.767	0.797	0.782	PERSON
0.567	0.507	0.535	DATE
0.870	0.851	0.860	NUMERIC
0.715	0.922	0.805	DEFINITION
0.630	0.642	0.636	PERIOD
0.781	0.850	0.814	MONEY
0.720	**0.761**	**0.740**	

Table 11. Performance results using 10 folds cross-validation on QC-BOB data with factoid question classes.

As can be seen in the table, the performance in terms of precision, recall and F-measure is higher on the coarse classes LR-TREC corpus than on the QC-BOB corpora, though the difference is not so big, about 5%. However, if we consider only factoid questions in QC-BOB data in which we do not take into accounts the INSTRUCTION, YESNO and CHAT categories, the difference gap is higher, up to 8%. These results follow our expectation since the number of coarse categories of LR-TREC is smaller than of QC-BOB, with 6 compared to 12. Another reason is that the questions in LR-TREC tend to be more well-formed and there is less noises than in QC-BOB.

In contrast, the results on fine-grain LR-TREC have shown a poorer performance than on QC-BOB with a quite big difference of 16%. This could be explained by the mentioned reason that some low-frequency categories in the testing set lowered down the overall result. Another possible reason is the variation of questions in LR-TREC, especially some "interesting" intended questions added by the designers. These variations might not be covered by the current set of features, while these features can easily handle the generally simpler patterns in QC-BOB.

CHAPTER VI

TOPIC SHIFT AND TOPIC CONTINUATION CLASSIFICATION: EXPERIMENTS AND RESULTS

This chapter describes the results of the experiments on topic shift and topic continuation classification. The first section explained the methodology for solving the problem. Next, the data set descriptions and feature extraction used for experiments are described in detail. We then examine the results of the experiments, including those on contextual TREC and FUQC-BOB. We conclude with the discussions about the overall results.

VI.1. Methodology

The second aim of the thesis is topic shift and topic continuation classification. As stated in the research issue 2, we would like to compare the performance between the user-collected data and the gold-standard contextual TREC data. In this section, the methodology for answering this question will be explained in more details.

In order to address the problem, a similar procedure of conducting series of experiments as on question classification is performed. First, we establish a baseline to be able to compare the results. The baseline is basically setup by re-examining the previous work of (Yang et al, 2006). Following that, the TREC 2001, 2004 corpora are used for training and testing. The set of features proposed by Yang et al and the decision tree using Weka tool are also chosen as question representation and classifier for the experiments. In the next step, we continue experimenting with the annotated BOB corpus, while still based on the same other settings such as: features and classifier. Notice that the annotation of the BOB corpus in this task is different from the previous task; it marks a question with only either a topic or a continuation, not with the taxonomy. In the last step, we compare the results on this corpus with those from the baseline experiments.

To conclude, the works for this section are:

- Re-examine previous work to extract the features using for question representation.
- Run the decision tree classifier using the extracted features on TREC corpus in order to setup the baseline measure.
- Annotate a corpus from the BOB system user logs.

- Run the decision tree classifier using the same features on the annotated BOB corpus and compare the results.

VI.2. Data descriptions

Similarly to the question classification task, there are also two corpus used for experimenting with topic shift and topic continuation classification. The TREC 2001, 2004 corpus is used for setting up the baseline. This corpus has been commonly used by previous works, i.e (Yang et al, 2006) and (De Boni et al, 2003). The other corpus is the BOB system logs which are the same logs using for question classification but different annotation. In this section, the descriptions of those corpora are given and we also explain how they serve for experiments.

TREC-2001, 2004

The TREC-2001 QA track began to include a "context" task which aimed at testing system's ability to track context through a series of questions (Voorhees, 2002). In the first attempt, the NIST staff prepared a set of 42 questions, divided into 10 dialogues/series of related questions. In 2004, the task was introduced again with a new corpus with total of 286 questions split into 65 dialogues, with each dialogue focus on a specific topic. The annotation of the corpora is done by marking the first sentence of the dialogue as topic and the rest as continuation questions. Note that we also need some modifications in the TREC-2004 in which the first question in a session might have pronouns referring to the topic of the session and thus we need to replace those pronouns by the topic phrase. This is necessary for extracting the features as described in the next section. Thus, we will use the TREC-2001 corpus as the testing set and the TREC-2004 as the training set. Figure 7 shows some examples of annotated questions from the TREC-2004 data (the first three sessions):

T__ When was the first Crip gang started?
F__ What does the name mean or come from?
F__ Which cities have Crip gangs?
F__ What ethnic group/race are Crip members?
F__ What is their gang color?
T__ What is the name of Durst's group?
F__ What record company is he with?
F__ What are titles of the group's releases?
F__ Where was Durst born?
T__ When was the Hale Bopp comet discovered?
F__ How often does it approach the earth?
F__ In what countries was the comet visible on its last return?

Figure 5. Sample annotated questions from TREC-2004 corpus.

BOB-FUQC

We use the same BOB system log files for collecting the data for experiments of topic shift and topic continuation classification task. However, for annotating

these logs, we no longer consider the taxonomy but the interactive dialogues between the user and the system. The annotation was done mainly by subjective decisions and sometimes it was hard to decide whether a sentence is a topic shift or not. For example, two consecutive questions "When can I return a book?" and "How can I borrow a book?" could be considered as belonging to the same topic about "book". However, if we distinguish between the topic about "returning book" and "borrowing book", they should be annotated with different topics. There are also a lot of refinements of the corpus since many noises are included in the original data. The final corpus contains 1364 questions divided into 485 dialogues. Figure 8 shows some samples of annotated questions from the BOB-FUQC corpus:

T__ do you know stella?
F__ who are your colleagues?
F__ colleague?
F__ frasnelli?
T__ How can I use Metalib?
F__ How can I download the article from metalib?
F__ Does the library possess journals that can be consulted online?
T__ I need an article from a scientific journal
F__ other publications
F__ otherwise
T__ where can i find a thesis paper
F__ particular
F__ topic
F__ detailed literature
F__ yes
F__ ok

Figure 6. Sample annotated questions from BOB-FUQC corpus.

VI.3. Feature extraction

This section will describe the features used for the topic shift and topic continuation classification task. This is mainly a re-examine work which has been done by (Yang et al, 2006). The feature set contains two main categories of features: syntactic and semantic features.

Syntactic features

These features capture some certain syntactic information of each question. Thus, in the first step, each question is tagged with part-of-speech information. This is done by using the TreeTagger tool which is used for annotating text with part-of-speech and lemma information. Next, the following binary syntactic features are extracted:

pronoun: whether a question has a pronoun or not.

propernoun: whether a question has a proper noun or not.

noun: whether a question has a noun or not.

verb: whether a question has a verb or not.

Semantic features

These features capture the semantic similarity between the current question and the previous ones, the context. The algorithm for computing similarity between the current question and its context is shown in the below. The more details could be found in (De Boni & Manandhar, 2005; Yang et al, 2006).

Given the current question Q_i and a sequence of history questions Context={Q_{i-n}, ..,Q_{i-1}}, the semantic similarity measure of Q_i and the Context is following:

$$ContextSimilarity(Q_i, Context) = \max_{1 \le j \le l} d(j).SentenceSimilarity(Q_i, Q_{i-j})$$

Following that, f(j) is a decay function. It gives more similarity weight to the closer questions in the same context with respect to the current question.

$$d(x) = 1 - \frac{1}{1 + e^{(l-x)}}$$

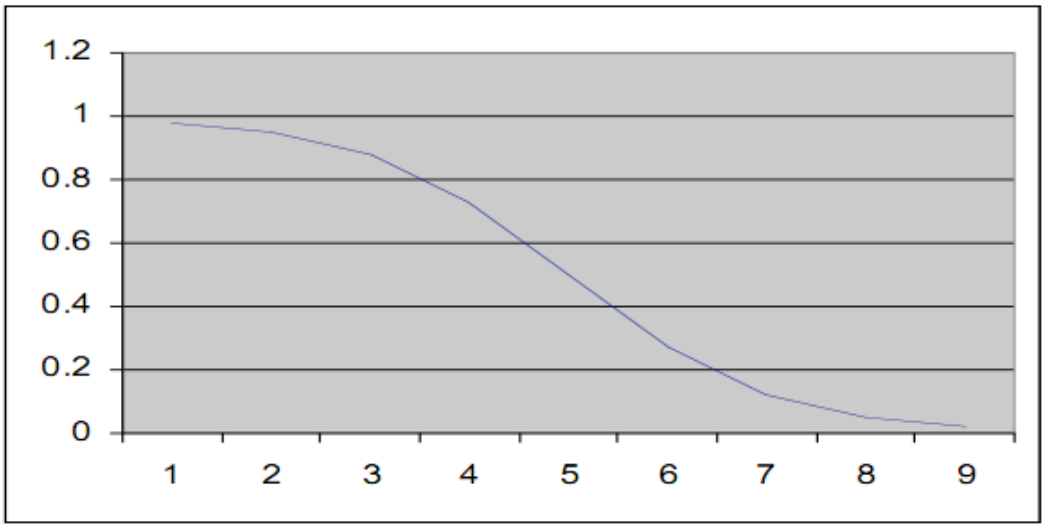

Figure 7. Decay function graph.

The SentenceSimilarity in the above formula is given as below:

$$SentenceSimilarity(Q, Q') = \frac{1}{n} \sum_{1 \le j \le n} \max_{1 \le i \le m} WordSimilarity(w_j, w'_i)$$

$$Q = \{w_1, w_2, .., w_n\}$$

$$Q' = \{w_1, w_2, .., w_m\}$$

The above formula shows that sentence similarity depends on word similarity. There have been some works proposing various ways of computing this measure based on WordNet. For example, the Path measure is the inverse of the shortest path length between two word senses in WordNet; the Lin measure is based on the information content, etc. More details about the measures could be found in (Perdersen et al, 2004). For implementing these measures, the WordNet::Similarity tool by (Pedersen et al, 2004) was used. In this work, we have used four measures in the package tool to compute the semantic similarity features and those are described in the following:

path_noun: sentence similarity based on the noun similarities using the path measure.

path_verb: sentence similarity based on the non-trivial verb similarities using the path measure. The trivial verbs include "does, do, did, will, would, might, could, should, shall".

wup_noun: sentence similarity based on the noun similarities using the Wu & Palmer's measure.

wup_verb: sentence similarity based on the noun similarities using the Wu & Palmer's measure.

lin_noun: sentence similarity based on the noun similarities using the Lin's measure.

lin_verb: sentence similarity based on the noun similarities using the Lin's measure.

VI.4. Results on contextual TREC data

The first experiment which is similar to dealing with question classification was performed in order to establish the baseline and features set. It is basically a re-examination of (Yang et al, 2006). As explained in the previous section, we used the same corpora as (Yang et al, 2006), including TREC 2001 and TREC 2004. The TREC 2001 data set contains 42 questions/10 dialogues and it was used as the testing set. The TREC 2004 data set consists of 286 question/65 dialogues served as the training set. Questions are represented by 10 features, namely: pronoun, proper noun, noun, verb, path_noun, path_verb, lin_noun, lin_verb, wup_noun and wup_verb. The chosen classifier is RandomForest, which is an extension of the decision tree, and it was implemented in Weka tool. All the algorithm parameters were used with default values. The algorithm was trained using the TREC 2004 training set but tested with two different settings which in turn used TREC 2004 and TREC 2001 as the testing sets. This is useful for comparing with the previous works which are also mentioned in this section.

The performances of both settings can be found in the tables 12, 13, 14, 15 and the comparisons with previous works are found in the table 16.

	Predicted class	
True class	Topic	Follow-up
Topic	65	0
Follow-up	0	221

Table 12. Confusion matrix result using TREC 2004 as both training and testing sets.

Precision	Recall	F-Measure	Class
1	1	1	Topic
1	1	1	Follow-up
1	**1**	**1**	

Table 13. Performance result using TREC 2004 as both training and testing sets.

	Predicted class	
True class	Topic	Follow-up
Topic	9	1
Follow-up	7	25

Table 14. Confusion matrix result using TREC 2004 as training set and TREC 2001 as testing set.

Precision	Recall	F-Measure	Class
0.563	0.900	0.692	Topic
0.962	0.781	0.862	Follow-up
0.867	**0.810**	**0.822**	

Table 15. Performance result using TREC 2004 as training set and TREC 2001 as testing set.

The above results show that using the training set gave a better performance than using the testing set. This makes sense because the classifier was tested on the data it had been learned before. However, the over good result on the training data with no errors also suggest that it could be reasoned by over-fitting and it was proven by the results in the tables 14 and 15.

	Precision	Recall	F-Measure
De Boni et al, 2003	0.760	0.840	0.798
Yang et al, 2006	**0.895**	**0.920**	**0.862**
This work	0.867	0.810	0.822

Table 16. Performance result using TREC 2004 as training and testing sets

Comparing the result with previous works, the performance of this work lied in between the results of De Boni & Manandhar and Yang et al. Though this work was a re-examination of Yang et al.'s work, it has not been achieved the same results because of several reasons. First, Yang et al. made use of different tagger tool to extract syntactic features. They chose GATE (Cunningham et al, 2002) while we have used TreeTagger (Schmid, 1995). The differences between two taggers could lead to some differences in syntactic feature extraction. Second, though we also used decision tree as the classifier as Yang et al. but the implementations could be different and it is also not clear whether they made any particular parameter tunings. However, with the achieved results, we still

have to setup a relatively good and reliable baseline for next comparisons described in the next section.

VI.5. Results on FUQC-BOB data

The second experiment is to investigate the performance of the classifier using the same features on the real life users' dialogues, FUQC-BOB. In the experiment, 1364 questions which correspond to 485 dialogues are extracted from the BOB system logs. The corpus served as both training and testing by using 10 folds cross-validation technique as described in the previous section. All other settings including the classifier and features remained the same. The detailed results are shown in the tables below.

	Predicted class	
True class	Topic	Follow-up
Topic	229	256
Follow-up	183	694

Table 17. Confusion matrix result using 10 folds cross-validation on FUQC-BOB data.

Precision	**Recall**	**F-Measure**	**Class**
0.556	0.472	0.511	Topic
0.741	0.791	0.760	Follow-up
0.668	**0.678**	**0.671**	

Table 18. Performance result using 10 folds cross-validation on FUQC-BOB data.

As can be seen in the result tables, the performance in terms of precision, recall and F-measure achieved on the FUQC-BOB corpus was much lower than on TREC corpus, only about 67% compared to 82% by F-measure. Especially, the chance of predicting the topic class was only around 50%, which is very low. If we consider a baseline model of having all the questions marked as topics, the accuracy of topic predictions would be 35% (485/1364). In the results described by (Yang et al, 2006), it also showed a performance downward on the real user data, so these results are expected and some explanations will be provided in the next section.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

The chapter presents a discussion of the results obtained from the experiments described in the previous chapters. It also contains some possible directions for future work on these tasks.

Question Classification

The results of this research so far have shown that there is a remarkable difference between the system using coarse and fine grain taxonomy and that the taxonomy has a big impact on the performance of a question classification system. However, when looking at the performance of the system working on the QC-BOB corpus, it is hard to draw any absolute conclusions on the comparisons because the taxonomy which applied to QC-BOB is different.

One possible reason is that the category distribution in the QC-BOB corpus is also significantly different in comparison with the LR-TREC, i.e. there are many instances of a few categories, and we might expect that the results are biased. Moreover, the domain of the QC-BOB corpus is also much narrower, the topics tend to be limited to small numbers and most notable questions have common patterns like "Where can I find …", "Who is …", "How do I …". Hence, the results on the QC-BOB could easily outperform the results on the more complicated fine grain classes LR-TREC corpus but worse than on the coarse classes LR-TREC corpus.

Though, by comparing the results on two corpora, it seems that the results and settings taken from the standard data like TREC could be well applied into our specific data. This is important because many researchers have been testing their systems on this corpus and it gives a good indication about the performance of the system. Thus, a good approach could be that one starts his work on the TREC corpus to solve "as good as he can" the problem by exploiting the features, classifiers or taxonomy, etc. Then these settings could be applied and adapted into the specific need data.

The results of this research open several directions for future work. First, the larger question corpus is necessary so that it will be more balanced and improve its coverage. It will give a more reliable corpus for testing classifiers. Second, there is a need to verify and/or design a new taxonomy for QC-BOB corpus. This work has used one taxonomy exclusively for the QC-BOB corpus and yet this taxonomy has not been approved concerning its correctness as well as completeness. Another important task is to exploit a better feature set to capture complicated patterns and to improve the performance of classifiers. This task is

considered challenging and is still a big topic that being attracted by researches to solve.

Topic shift and topic continuation classification

As can be seen in the results from both experiments, there was a big difference in term of performance between the baseline results on contextual TREC data and results on FUQC-BOB data. There are several possible reasons to explain this observation.

First, when we look at the TREC corpus and FUQC-BOB corpus, there are obvious distinctions. The TREC corpus in most cases contains fully well-formed questions and sentences; the dialogues are longer and usually shifted into totally different topics. For example, in TREC 2004, one dialogue with the topic of "Rhodes scholars" is followed by the "animal agouti" dialogue and then by the "Black panthers organization". In contrast, the FUQC-BOB corpus contains many noisy and very short sentences. The users also often tend to repeat or rephrase their questions and the topics in general are quite related together. For instance, there are many inputs from users with only one word like "yes", "more", "specific", "color print", "print", etc. The consequence of topics could be "library", "people in library", "borrow book", "return book", etc. and these topics share a large proportion of common words and even sentences. Therefore, predicting correct topic sentences in TREC corpus seems to be easier than in FUQC-BOB. In this problem of topic shift and topic continuation classification, open domain can be considered as an easier task than closed domain, in contrast to the question classification problem.

The second reason is that the features used for representing the question sentences might be not relevant for FUQC-BOB data. As already mentioned, this corpus contains mostly short sentences, so a chance that important syntactic features like pronoun, proper noun do not occur is quite high. The length of the dialogue is also usually short, with about 2-4 questions so that the quality of features is often low.

Further work is needed in order to improve or solve some still open problems. First, there is a need to enlarge as well as improve the quality of corpora. This could be done by collecting more data from users and propose a good approach for refining and annotating them. Second, we can improve the performance by exploiting a better set of features; especially ones can adapt the domain and specific needs. Some current suggestions could be definite noun feature which is mentioned by (Yang et al, 2006) or using co-references. Another important possibility is to use more context information in dialogues by taking advantage of the answers to the user interactions. These answers are usually well prepared and they provide rich information on the topic as well as on the features. However, in this case, it is necessary to be careful when tackling a raised problem that the answers must be relevant to the user questions and topic. Otherwise, it will mislead and even lower down the performance of the system.

REFERENCES

De Boni M. C., Manandhar S. 2005. **Implementing clarification dialogs in open domain question answering**. *Natural Language Engineering.*

Burke R., Hammond K., Kulyukin V., Lytinen S., Tomuro N., and Schoenberg S. **Question answering from frequently asked question files: Experiences with the FAQFinder system**. *AI Magazine, 18(2):57--66, 1997.*

Cunningham H. **GATE, a General Architecture for Text Engineering**. *Computers and the Humanities, volume 36, pp. 223-254, 2002.*

Dridan R., Kordoni V., Nicholson J. **Enhancing Performance of Lexicalised Grammars.** *In Proceedings of ACL-08: HLT, Columbus, USA (2008).*

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

Hacioglu K. and Ward W. 2003. **Question classification with support vector machines and error correcting codes**. *In Proceedings of HLT-NACCL 2003*.

Harabagiu S., Maiorano J., Pasca M. **Open-domain textual question answering techniques**, *Natural Language Engineering 9(3): 231–267, 2003.*

Harabagiu S., Moldovan D., Paşca M. 2001. **Answering complex, list and context questions with LCC's Question-Answering Server.** *Proceedings of TREC 2001.*

Hermjakob U. **Parsing and question classification for question answering**. *Proceedings of the workshop on ARABIC language processing: status and prospects, p.1-6, July 06, 2001, Toulouse, France*

Huang Z., Thint M., Qin Z. question classification **using Head Words and their Hypernyms**. *EMNLP 2008: 927-936.*

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

Li X., Roth D. **Learning question classifiers: the role of semantic information.** *Natural Language Engineering, v.12 n.3, p.229-249, September 2006.*

Li W. **question classification Using Language Modeling**. *In CIIR Technical Report: University of Massachusetts, Amherst. (2002)*

Lytinen S., Tomuro N. **The use of question types to match questions in FAQFinder**. *In Proc. AAAI-2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases, pages 46--53, 2002.*

Moldovan D., Harabagiu S., Mihalcea R., Goodrum R., Girju R., Rus V. **Lasso. A Tool for Surfing the Answer Net.** *In Proceedings of the 8th Text REtrieval Conference (TREC-8), 1999.*

Mitchell, T. M.**, Machine Learning**, *1997*.

Pasca M. 2003. **Open-Domain Question Answering from Large Text Collections**. *CSLI Publication, Center for the Study of Language and Information, 157p.*

Pedersen T., Patwardhan S., Michelizzi J. **"WordNet::Similarity - Measuring the Relatedness of Concepts".** *In: Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004) pp. 1024-1025. San Jose, CA. July, 2004.*

Radev D. R., Fan W., Qi H., Wu H. and Grewal A. **Probabilistic Question Answering from the Web**. *In Proceedings of the 11th World Wide Web Conference (WWW2002), Hawaii, 2002.*

Sunblad H. **A Re-examination of question classification**. *In Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007, pages 394-397.*

Sunblad H. **question classification in Question Answering Systems**. *Thesis No. 1320, Department of Computer and Information Science.*

Suzuki J., Taira H., Sasaki Y., and Maeda E. 2003b. **Question classification using HDAG kernel**. *In The ACL 2003 Workshop on Multilingual Summarization and Question Answering.*

Yang F., Feng J. and Fabbrizio G. D. **A Data Driven Approach to Relevancy Recognition for Contextual Question Answering**. *HLT-NAACL 2006 Workshop on Interactive Question Answering. New York, USA, June 8-9, 2006*.

Zaanen V., Pizzato M., Molla L. A.. 2005. **question classification by Structure Induction**. *In International Joint Conference on Artificial Intelligence, 2005, pages 1638-1639.*

Zhang D. and Lee W. S. 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.*

http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

http://wn-similarity.sourceforge.net/

APPENDIX

INSTALLATION AND TOOL DESCRIPTIONS

---

1. Installation

Prerequisites

The following software are required to run the system:

- bash (if run on Linux systems)
- csh (if run on Linux systems)
- Perl version 5.6 or later.
- Java VM 1.5 or later (required by Stanford Parser)
- WordNet 3.0 (required by WordNet::Similarity)
- Text::Similarity (required by WordNet::Similarity)
- WordNet::QueryData: (required by WordNet::Similarity)
- WordNet::Similarity

1.1. Linux

Install bash, csh, perl and Java

In most Linux systems, these packages will be installed by default. If not, one can use the following commands to search and download them accordingly:

```
$apt-cache search package_name

$apt-get install package_name
```

Install WordNet 3.0

```
$cd /tmp

$wget http://wordnet.princeton.edu/3.0/WordNet-3.0.tar.gz

$tar -zxvf WordNet-3.0.tar.gz

$cd /tmp/WordNet-3.0
```

```
$./configure

($./configure –-prefix=/tmp/wn3.0 # non-root user)

$make

$make install
```

Install WordNet::QueryData

This assumes that you have already installed Perl.

For root user:

```
$sudo perl –MCPAN –e shell

>cpan install WordNet::QueryData

...

cpan> quit
```

For non-root user:

Download and unpack the package:

```
$cd /tmp

$wget http://search.cpan.org/CPAN/authors/id/J/JR/JRENNIE/ \
WordNet-QueryData-1.46.tar.gz

$tar -zxvf WordNet-QueryData-1.46.tar.gz
```

Set the WNHOME environment variable to the WordNet3.0 location:

```
$export WNHOME=/tmp/wn3.0
```

Then, install as usual:

```
$perl Makefile.PL PREFIX=/tmp/wnqd1.46

$make

$make test

$make install
```

Install Text::Similarity

For root user:

```
$sudo perl –MCPAN –e shell

>cpan install Text::Similarity
...
```

```
cpan> quit
```

For non-root user:

Download and unpack the package:

```
$cd /tmp

$wget http://search.cpan.org/CPAN/authors/id/J/JA/JASONM/ \
Text-Similarity-0.02.tar.gz

$tar -zxvf Text-Similarity-0.02.tar.gz
```

Install:

```
$perl Makefile.PL PREFIX=/tmp/ts0.02

$make

$make test

$make install
```

Install WordNet::Similarity

For root user:

```
$sudo perl –MCPAN –e shell

>cpan install WordNet::Similarity

...

cpan> quit
```

For non-root user:

Download and unpack the package:

```
$cd /tmp

$wget http://search.cpan.org/CPAN/authors/id/T/TP/TPEDERSE/ \
WordNet-Similarity-2.01.tar.gz

$tar -zxvf WordNet-Similarity-2.01.tar.gz
```

Set the PERL2LIB environment variables (could be set in .bashrc file):

```
export \
PERL5LIB="/tmp/ws2.01/lib/perl5/site_perl/5.8.8:/tmp/wnqd1.46/
\
lib/perl5/site_perl/5.8.8:/tmp/ts0.02/lib/perl5/site_perl/5.8.
8”
```

Install:

```
$perl Makefile.PL PREFIX=/tmp/ws2.01

$make

$make test

$make install
```

I.2. Windows

Install WordNet 2.0

Download and install the package from the website:

http://wordnet.princeton.edu/oldversions

Install WordNet::QueryData 1.38

Download and unzip the package from the link:

http://people.csail.mit.edu/jrennie/WordNet/WordNet-QueryData-1.38.tar.gz

Copy the file "QueryData.pm" into the directory "C:\Perl\site\lib\WordNet" (create the WordNet directory if it is not existed).

Install Text::Similarity 0.07

Download and unzip the package from the link:

http://archive.cpan.cz/authors/id/T/TP/TPEDERSE/Text-Similarity-0.07.tar.gz

Open the unzipped folder, copy the directory "Text" which includes the files "OverlapFinder.pm", "Similarity.pm" and "Similarity" into ""C:\Perl\site\lib\Text" (create the Text directory if it is not existed).

Install nmake

Download the package from the link:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q132084

Run the downloaded executable to extract the files. Copy both the NMAKE.EXE and the NMAKE.ERR file to the "C:\Perl\bin" directory.

Install WordNet::Similarity 0.16

Download and unzip the package from the link:

http://backpan.perl.org/authors/id/T/TP/TPEDERSE/WordNet-Similarity- \ 0.16.tar.gz

Type the following commands in cmd promt (note to cd into the current directory):

```
>perl Makefile.pl

>nmake

>nmake test

>nmake install
```

2. Tool descriptions

Followings are the descriptions of the scripts used for the experiments of this work.

Script's name	Description
extract_questions.pl	A Perl script to extract user questions in desired format from log files since the log files contain other information, such as: markers, answers, etc.
extract_TREC_questions.pl	A Perl script to extract questions in desired format from TREC context track data since they are in XML format.
input_types.pl	A Perl script to help improve the process of annotating questions extracted from log files.
annotate_with_treetagger.pl	A Perl script to parse questions to get the lemmas and POS information using TreeTagger utility.
convert_to_question_features.pl	A Perl script to convert questions into question classification features as described in chapter V.1.3
convert_to_FU_question_features.pl	A Perl script to convert questions into topic shift and topic continuation features as described in chapter V.2.2
naïve_bayes.pl	A Perl script of the implementation of Naïve Bayes algorithm.
learn_with_naive_bayes.pl	A Perl script to learn a Naïve Bayes model from a provided training corpus.
classify_testset_with_naive_bayes.pl	A Perl script to evaluate a testing corpus on a trained Naïve Bayes model.
classify_cross_validation _with_naive_bayes.pl	A Perl script to evaluate a corpus using cross-validation technique using Naïve Bayes.