A Data Driven Approach to Relevancy Recognition for Contextual Question Answering

Fan Yang*
OGI School of Science & Engineering
Oregon Health & Science University
fly@cslu.ogi.edu

Junlan Feng and **Giuseppe Di Fabbrizio**
AT&T Labs - Research
180 Park Avenue, Florham Park, NJ, 07932 - USA
junlan@research.att.com, pino@research.att.com

Abstract

Contextual question answering (QA), in which users' information needs are satisfied through an interactive QA dialogue, has recently attracted more research attention. One challenge of engaging dialogue into QA systems is to determine whether a question is relevant to the previous interaction context. We refer to this task as relevancy recognition. In this paper we propose a data driven approach for the task of relevancy recognition and evaluate it on two data sets: the TREC data and the HandQA data. The results show that we achieve better performance than a previous rule-based algorithm. A detailed evaluation analysis is presented.

1 Introduction

Question Answering (QA) is an interactive human-machine process that aims to respond to users' natural language questions with exact answers rather than a list of documents. In the last few years, QA has attracted broader research attention from both the information retrieval (Voorhees, 2004) and the computational linguistic fields (http://www.clt.mq.edu.au/Events/Conferences/acl04qa/). Publicly accessible web-based QA systems, such as AskJeeves (http://www.ask.com/) and START (http://start.csail.mit.edu/), have scaled up this technology to open-domain solutions. More task-oriented QA systems are deployed as virtual customer care agents addressing questions about specific domains. For instance, the AT&T Ask Allie® agent (http://www.allie.att.com/) is able to answer questions about the AT&T plans and services; and the Ikea "Just Ask Anna!" agent (http://www.ikea.com/ms/en_US/) targets questions pertaining the company's catalog. Most of these QA systems, however, are limited to answer questions in isolation. The reality is that users often ask questions naturally as part of contextualized interaction. For instance, a question "How do I subscribe to the AT&T CallVantage® service?" is likely to be followed by other related questions like "How much will the basic plan cost?" and so on. Furthermore, many questions that users frequently want answers for cannot be satisfied with a simple answer. Some of them are too complicated, broad, narrow, or vague resulting that there isn't a simple good answer or there are many good answer candidates, which entails a clarification procedure to constrain or relax the search. In all these cases, a question answering system that is able to answer contextual questions is more favored.

Contextual question answering as a research challenge has been fostered by TREC (Text Retrieval Conference) since 2001. The TREC 2001 QA track made the first attempt to evaluate QA systems' ability of tracking context through a series of questions. The TREC 2004 re-introduced this task and organized all questions into 64 series, with each series focusing on a specific topic. The earlier questions in a series provide context for the on-going question. However, in reality, QA systems will not be

*The work was done when the first author was visiting AT&T Labs - Research.

Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006, pages 33–40,
New York City, NY, USA. June 2006.

informed about the boundaries between series in advance.

One challenge of engaging dialogue into QA systems is to determine the boundaries between topics. For each question, the system would need to determine whether the question begins a new topic or it is a follow-up question related to the current existing topic. We refer to this procedure as *relevancy recognition*. If a question is recognized as a follow-up question, the next step is to make use of context information to interpret it and retrieve the answer. We refer to this procedure as *context information fusion*. Relevancy recognition is similar to text segmentation (Hearst, 1994), but relevancy recognition focuses on the current question with the previous text while text segmentation has the full text available and is allowed to look ahead.

De Boni and Manandhar (2005) developed a rule-based algorithm for relevancy recognition. Their rules were manually deduced by carefully analyzing the TREC 2001 QA data. For example, if a question has no verbs, it is a follow-up question. This rule-based algorithm achieves 81% in accuracy when recognizing the question relevance in the TREC 2001 QA data set. The disadvantage of this approach is that it involves a good deal of human effort to research on a specific data set and summarize the rules. For a new corpus from a different domain, it is very likely that one would have to go over the data set and modify the rules, which is time and human-effort consuming. An alternative is to pursue a data driven approach to automatically learn the rules from a data set. In this paper, we describe our experiments of using supervised learning classification techniques for the task of relevancy recognition. Experiments show that machine learning approach achieves better recognition accuracy and can also be easily applied to a new domain.

The organization of this paper is as follows. In Section 2, we summarize De Boni and Manandhar's rule-based algorithm. We present our learning approach in Section 3. We ran our experiments on two data sets, namely, the TREC QA data and the HandQA data, and give the results in Section 4. In section 5, we report our preliminary study on context information fusion. We conclude this paper in Section 6.

2 Rule-Based Approach

De Boni and Manandhar (2005) observed the following cues to recognize follow-up questions:

- *Pronouns and possessive adjectives*. For example, if a question has a pronoun that does not refer to an entity in the same sentence, this question could be a follow-up question.
- *Cue words*, such as "precisely" and "exactly".
- *Ellipsis*. For example, if a question is not syntactically complete, this question could be a follow-up question.
- *Semantic Similarity*. For example, if a question bears certain semantic similarity to previous questions, this question might be a follow-up question.

  De Boni and Manandhar (2005) proposed an algorithm of calculating the semantic similarity between the current question Q and a previous question Q'. Supposed Q consists of a list of words $(w_1, w_2, ..., w_n)$ and Q' consists of $(w'_1, w'_2, ..., w'_m)$:

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

  The value of $WordSimilarity(w, w')$ is the similarity between two words, calculated from WordNet (Fellbaum, 1998). It returns a value between 0 (w and w' have no semantic relations) and 1 (w and w' are the same).

Motivated by these observations, De Boni and Manandhar (2005) proposed the rule-based algorithm for relevancy recognition given in Figure 1. This approach can be easily mapped into an hand-crafted decision tree. According to the algorithm, a question follows the current existing topic if it (1) contains reference to other questions; or (2) contains context-related cue words; or (3) contains no verbs; or (4) bears certain semantic similarity to previous questions or answer. Evaluated on the TREC 2001 QA context track data, the recall of the algorithm is 90% for recognizing first questions and 78% for follow-up questions; the precision is 56% and 76% respectively. The overall accuracy is 81%.

Given the current question Q_i and a sequence of history questions Q_{i-n}, ..., Q_{i-1}:

1. *If Q_i has a pronoun or possessive adjective which has no references in the current question, Q_i is a follow-up question.*
2. *If Q_i has cue words such as "precisely" or "exactly", Q_i is a follow-up question.*
3. *If Q_i does not contain any verbs, Q_i is a follow-up question.*
4. *Otherwise, calculate the semantic similarity measure of Q_i as*

   $$SimilarityMeasure(Q_i) = \max_{1 \le j \le n} f(j) \cdot SentenceSimilarity(Q_i, Q_{i-j})$$

   Here f(j) is a decay function. If the similarity measure is higher than a certain threshold, Q_i is a follow-up question.
5. *Otherwise, if answer is available, calculate the semantic distance between Q_i and the immediately previous answer A_{i-1}: $SentenceSimilarity(Q_i, A_{i-1})$. If it is higher than a certain threshold, Q_i is a follow-up question that is related to the previous answer.*
6. *Otherwise, Q_i begins a new topic.*

Figure 1: Rule-based Algorithm

3 Data Driven Approach

3.1 Decision Tree Learning

As a move away from heuristic rules, in this paper, we make an attempt towards the task of relevancy recognition using machine learning techniques. We formulate it as a binary classification problem: a question either begins a new topic or follows the current existing topic. This classification task can be approached with a number of learning algorithms such as support vector machines, Adaboost and artificial neural networks. In this paper, we present our experiments using Decision Tree. A decision tree is a tree in which each internal node represents a choice between a number of alternatives, and each leaf node represents a decision. Learning a decision tree is fairly straightforward. It begins from the root node which consists of all the training data, growing the tree top-down by recursively splitting each node based on maximum information gain until certain criteria is met. Although the idea is simple, decision tree learning is often able to yield good results.

3.2 Feature Extraction

Inspired by De Boni and Manandhar's (2005) work, we selected two categories of features: syntactic features and semantic features. Syntactic features capture whether a question has certain syntactic components, such as verbs or pronouns. Semantic features characterize the semantic similarity between the current question and previous questions.

3.2.1 Syntactic Features

As the first step, we tagged each question with part-of-speech tags using GATE (Cunningham et al., 2002), a software tool set for text engineering. We then extracted the following binary syntactic features:

PRONOUN: whether the question has a pronoun or not. A more useful feature would be to label whether a pronoun refers to an entity in the previous questions or in the current question. However, the performances of currently available tools for anaphora resolution are quite limited for our task. The tools we tried, including GATE (Cunningham et al., 2002), LingPipe (`http://www.alias-i.com/lingpipe/`) and JavaRAP (Qiu et al., 2004), tend to use the nearest noun phrase as the referents for pronouns. While in the TREC questions, pronouns tend to refer to the topic words (focus). As a result, unsupervised anaphora resolution introduced more noise than useful information.

ProperNoun: whether the question has a proper noun or not.

NOUN: whether the question has a noun or not.

VERB: whether the question has a verb or not.

DefiniteNoun: if a question has a definite noun phrase that refers to an entity in previous questions, the question is very likely to be a follow-up question. However, considering the difficulty in automatically identifying definite noun phrases and their referents, we ended up not using this feature in our training because it in fact introduced misleading information.

3.3 Semantic Features

To compute the semantic similarity between two questions, we modified De Boni and Manandhar's formula with a further normalization by the length of the questions; see formula (2).

$$SentenceSimilarity(Q, Q') \quad (2)$$
$$= \frac{1}{n} \sum_{1 \le j \le n} (\max_{1 \le i \le m} WordSimilarity(w_j, w'_i))$$

This normalization has pros and cons. It removes the bias towards long sentences by eliminating the accumulating effect; but on the other hand, it might cause the system to miss a related question, for example, when two related sentences have only one key word in common.[1]

Formula (2) shows that sentence level similarity depends on word-word similarity. Researchers have proposed a variety of ways in measuring the semantic similarity or relatedness between two words (to be exact, word senses) based on WordNet. For example, the *Path (path) measure* is the inverse of the shortest path length between two word senses in WordNet; the *Wu and Palmer's (wup) measure* (Wu and Palmer, 1994) is to find the most specific concept that two word senses share as ancestor (least common subsumer), and then scale the path length of this concept to the root note (supposed that there is a virtual root note in WordNet) by the sum of the path lengths of the individual word sense to the root node; the *Lin's (lin) measure* (Lin, 1998) is based on information content, which is a corpus based measure of the specificity of a word; the *Vector (vector) measure* associates each word with a gloss vector and calculates the similarity of two words as the cosine between their gloss vectors (Patwardhan, 2003). It was unclear which measure(s) would contribute the best information to the task of relevancy recognition, so we just experimented on all four measures, path, wup, lin, and vector, in our decision tree training. We used Pedersen et al.'s (2004) tool *WordNet::Similarity* to compute these four measures. WordNet::Similarity implements nine different measures of word similarity. We here only used the four described above because they return a value between 0 and 1, which is suitable for using formula (2) to calculate sentence similarity, and we leave others as future work. Notice that the WordNet::Similarity implementation can only measure *path*, *wup*, and *lin* between two nouns or between two verbs, while it uses all the content words for the *vector* measure. We thus have the following semantic features:

path_noun: sentence similarity is based on the nouns[2] similarity using the *path* measure.

path_verb: sentence similarity is based on the non-trivial verbs similarity using the *path* measure. Trivial verbs include "does, been, has, have, had, was, were, am, will, do, did, would, might, could, is, are, can, should, shall, being".

wup_noun: sentence similarity is based on the nouns similarity using the *Wu and Palmer's* measure.

wup_verb: sentence similarity is based on the non-trivial verbs similarity using the *Wu and Palmer's* measure.

lin_noun: sentence similarity is based on the nouns similarity using the *Lin's* measure.

lin_verb: sentence similarity is based on the non-trivial verbs similarity using the *Lin's* measure.

vector: sentence similarity is based on all content words (nouns, verbs, and adjectives) similarity using the *vector* measure.

4 Results

We ran the experiments on two sets of data: the TREC QA data and the HandQA data.

4.1 Results on the TREC data

TREC has contextual questions in 2001 context track and 2004 (Voorhees, 2001; Voorhees, 2004). Questions about a specific topic are organized into a session. In reality, the boundaries between sessions are not given. The QA system would have to recognize the start of a new session as the first step of question answering. We used the TREC 2004 data as training and the TREC 2001 context track data as testing. The training data contain 286 factoid and list questions in 65 sessions[3]; the testing data contain 42 questions in 10 sessions. Averagely each session has about 4-5 questions. Figure 2 shows some example questions (the first three sessions) from the TREC 2001 context track data.

[1] Another idea is to feed the decision tree training both the normalized and non-normalized semantic similarity information and see what would come out. We tried it on the TREC data and found out that the normalized features actually have higher information gain (i.e. appear at the top levels of the learned tree.

[2] This is to filter out all other words but nouns from a sentence for measuring semantic similarity.

[3] In the TREC 2004 data, each session of questions is assigned a phrase as the topic, and thus the first question in a session might have pronouns referring to this topic phrase. In such cases, we manually replaced the pronouns by the topic phrase.

CTX1a	Which museum in Florence was damaged by a major bomb explosion in 1993?
CTX1b	On what day did this happen?
CTX1c	Which galleries were involved?
CTX1d	How many people were killed?
CTX1e	Where were these people located?
CTX1f	How much explosive was used?
CTX2a	Which industrial sector supplies the most jobs in Toulouse?
CTX2b	How many foreign companies were based there in 1994?
CTX2c	Name a company that flies there.
CTX3a	What grape variety is used in Chateau Petrus Bordeaus?
CTX3b	How much did the future cost for the 1989 Vintage?
CTX3c	Where did the winery's owner go to college?
CTX3d	What California winery does he own?

Figure 2: Example TREC questions

4.1.1 Confusion Matrix

Table 1 shows the confusion matrix of the decision tree learning results. On the testing data, the learned model performs with 90% in recall and 82% in precision for recognizing first questions; for recognizing follow-up questions, the recall is 94% and precision is 97%. In contrast, De Boni and Manandhar's rule-based algorithm has 90% in recall and 56% in precision for recognizing first questions; for follow-up questions, the recall is 78% and precision is 96%. The recall and precision of our learned model to recognize first questions and follow-up questions are all better than or at least the same as the rule-based algorithm. The accuracy of our learned model is 93%, about 12% absolute improvement from the rule-based algorithm, which is 81% in accuracy. Although the size of the data is too small to draw a more general conclusion, we do see that the data driven approach has better performance.

	Training Data			
	Predicted Class			
True Class	First	follow-up	Total	
First	63	2	65	
follow-up	1	220	221	
Total	64	222	286	
	Testing Data			
	Predicted Class			
True Class	First	follow-up	Total	Recall
First	9	1	10	90%
follow-up	2	30	32	94%
Total	11	31	42	
Precision	82%	97%		

Table 1: Confusion Matrix for TREC Data

4.1.2 Trained Tree

Figure 3 shows the first top two levels of the tree learned from the training data. Not surprisingly, *PRONOUN* turns out to be the most important feature which has the highest information gain. In the TREC data, when there is a pronoun in a question, the question is very likely to be a follow-up question. In fact, in the TREC 2004 data, the referent of pronouns very often is the topic phrase. The feature *path_noun*, on the second level of the trained tree, turns out to contribute most information in this recognition task among the four different semantic similarity measures. The similarity measures using *wup*, *wup_noun* and *wup_verb*, and the *vector* measure do not appear in any node of the trained tree.

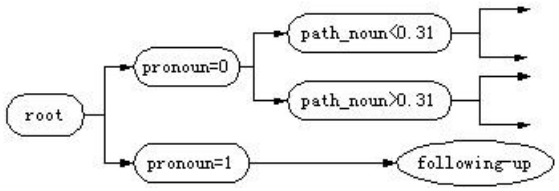

Figure 3: Trained Tree on TREC Data

The following are rules generated from the training data whose confidence is higher than 90%. Confidence is defined as out of the training records for which the left hand side of the rule is true, the percentage of records for which the right hand side is also true. This measures the accuracy of the rule.

- If *PRONOUN*=1 then follow-up question
- If *path_noun*$\geq$0.31 then follow-up question
- If *lin_noun*$\geq$0.43 then follow-up question
- If *path_noun*$<$0.15 and *PRONOUN*=0 then first question

De Boni and Manandhar's algorithm has this rule:"if a question has no verb, the question is follow-up question". However, we did not learn this rule from the data, nor the feature *VERB* appears in any node of the trained tree. One possible reason is that this rule has too little *support* in the training set (support is defined as the percentage of which the left hand side of the rule is true). Another possible reason is that this rule is not needed because the combination of other features is able to provide enough information for recognizing follow-up questions. In any case, the decision tree learns a (local)

optimized combination of features which captures most cases, and avoids redundant rules.

4.1.3 Error Analysis

The trained decision tree has 3 errors in the testing data. Two of the errors are mis-recognition of follow-up questions to be first questions, and one is the vice versa.

The first error is failure to recognize the question "which galleries were involved?" (CTX1c) as a follow-up question (see Figure 2 for context). It is a syntactically complete sentence, and there is no pronoun or definite noun in the sentence. Semantic features are the most useful information to recognize it as a follow-up question. However, the semantic relatedness in WordNet between the words "gallery" in the current question and "museum" in the first question of this session (CTX1a in Figure 2) is not strong enough for the trained decision tree to relate the two questions together.

The second error is failure to recognize the question "Where did the winery's owner go to college?" (CTX3c) as a follow-up question. Similarly, part of the reason for this failure is due to the insufficient semantic relatedness between the words "winery" and "grape" (in CTX3a) to connect the questions together. However, this question has a definite noun phrase "the winery" which refers to "Chateau Petrus Bordeaux" in the first question in this session. We did not make use of the feature *DefiniteNoun* in our training, because it is not easy to automatically identify the referents of a definite noun phrase, or even whether it has a referent or not. A lot of definite noun phrases, such as "the sun", "the trees in China", "the first movie", and "the space shuttles", do not refer to any entity in the text. This does not mean that the feature *DefiniteNoun* is not important, but instead that we just leave it as our future work to better incorporate this feature.

The third error, is failure to recognize the question "What does transgenic mean?" as the first question that opens a session. This error is due to the over-fitting of decision tree training.

4.1.4 Boosting

We tried another machine learning approach, Adaboost (Schapire and Singer, 2000), which is resistant (but not always) to over-fitting. It calls a given weak learning algorithm repeatedly in a series of rounds $t = 1, ..., T$. Each time the weak learning algorithm generates a rough "rule of thumb", and after many rounds Adaboost combines these weak rules into a single prediction rule that, hopefully, will be more accurate than any one of the weak rules. Figure 2 shows the confusion matrix of Adaboost learning results. It shows that Adaboost is able to correctly recognize "What does transgenic mean?" as beginning a new topic. However, Adaboost has more errors in recognizing follow-up questions, which results in an overall accuracy of 88%, slightly lower than decision tree learning.

	Training Data			
	Predicted Class			
True Class	First	follow-up	Total	
First	64	1	65	
follow-up	1	220	221	
Total	65	221	286	
	Testing Data			
	Predicted Class			
True Class	First	follow-up	Total	Recall
First	10	0	10	100%
follow-up	5	27	32	84%
Total	15	27	42	
Precision	67%	100%		

Table 2: Confusion Matrix Using Adaboosting

4.2 Results on the HandQA data

We also conducted an experiment using real-world customer-care related questions. We selected our test data from the chat logs of a deployed online QA system. We refer to this system as HandQA. HandQA is built using a telecommunication ontology database and 1600 pre-determined FAQ-answer pairs. For every submitted customer question, HandQA chooses one of these 1600 answers as the response. Each chat session contains about 3 questions. We assume the questions in a session are context-related.

The HandQA data are different from the TREC data in two ways. First, HandQA questions are real typed questions from motivated users. The HandQA data contain some noisy information, such as typos and bad grammars. Some users even treated this system as a search engine and simply typed in the keywords. Second, questions in a chat session basically asked for the same information. Very often, when the system failed to get the correct answer to

the user's question, the user would repeat or rephrase the same question, until they gave up or the system luckily found the answer. As an example, Figure 4 shows two chat sessions. Again, we did not use the system's answer in our relevancy recognition.

How to make number non published? Non published numbers How to make number non listed?
Is my number switched to Call Vantage yet? When will my number be switched? When is number transferred?

Figure 4: Example questions in HandQA

A subset of the HandQA data, 5908 questions in 2184 sessions are used for training and testing the decision tree. The data were randomly divided into two sets: 90% for training and 10% for testing.

4.2.1 Confusion Matrix

Table 3 shows the confusion matrix of the decision tree learning results. For recognizing first questions, the learned model has 73% in recall and 62% in precision; for recognizing follow-up questions, the recall is 75% and precision is 84%. The accuracy is 74%. A base line model is to have all questions except the first one as following up questions, which results in the accuracy of 64% (380/590). Thus the learned decision tree yields an absolute improvement of 10%. However, the results on this data set are not as good as those on the TREC data.

	Training Data Predicted Class			
True Class	First	follow-up	Total	
First	1483	490	1973	
follow-up	699	2646	3345	
Total	2182	3136	5318	

	Testing Data Predicted Class			
True Class	First	follow-up	Total	Recall
First	153	58	211	73%
follow-up	93	286	379	75%
Total	246	344	590	
Precision	62%	84%		

Table 3: Confusion Matrix for HandQA Data

4.2.2 Trained Tree

Table 5 shows the top two levels of the tree learned from the training data, both of which are on the semantic measure *path*. This again confirms that *path* best fits the task of relevancy recognition among the four semantic measures.

No syntactical features appear in any node of the learned tree. This is not surprising because syntactic information is noisy in this data set. Typos, bad grammars, and mis-capitalization affect automatic POS tagging. Keywords input also results in incomplete sentences, which makes it unreliable to recognize follow-up questions based on whether a question is a complete sentence or not. Furthermore, because questions in a session rarely refer to each other, but just repeat or rephrase each other, the feature *PRONOUN* does not help either. All these make syntactic features not useful. Semantic features turn out to be more important for this data set.

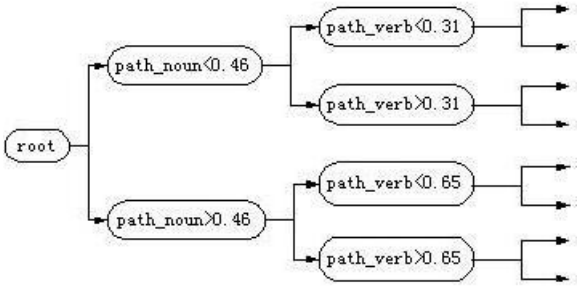

Figure 5: Trained Tree on HandQA Data

4.2.3 Error Analysis

There are two reasons for the decreased performance in this data set. The first reason, as we analyzed above, is that syntactical features do not contribute to the recognition task. The second reason is that consecutive chat sessions might ask for the same information. In the handQA data set, questions are basically all about telecommunication service, and questions in two consecutive chat sessions, although by different users, could be on very similar topics or even have same words. Thus, questions, although in two separate chat sessions, could have high semantic similarity measure. This would introduce confusing information to the decision tree learning.

5 Making Use of Context Information

Relevancy recognition is the first step of contextual question answering. If a question is recognized as following the current existing topic, the next step is to make use of the context information to interpret it

and retrieve the answers. To explore how context information helps answer retrieval, we conducted preliminary experiments with the TREC 2004 QA data. We indexed the TREC documents using the Lucene search engine (Hatcher and Gospodnetic, 2004) for document retrieval. The Lucene search engine takes as input a query (a list of keywords), and returns a ranked list of relevant documents, of which the first 50 were taken and analyzed in our experiments. We tried different strategies for query formulation. Simply using the questions as the query, only 20% of the follow-up questions find their answers in the first 50 returned documents. This percentage went up to 85% when we used the topic words, provided in TREC data for each section, as the query. Because topic words are usually not available in real world applications, to be more practical, we tried using the noun phrases in the first question as the query. In this case, 81% of the questions are able to find the answers in the returned documents. When we combined the (follow-up) question with the noun phrases in the first question as the query, the retrieved rate increases to 84%. Typically, document retrieval is a crucial step for QA systems. These results suggest that context information fusion has a big potential to improve the performance of answer retrieval. However, we leave the topic of how to fuse context information into the follow-up questions as future work.

6 Conclusion

In this paper, we present a data driven approach, decision tree learning, for the task of relevancy recognition in contextual question answering. Experiments show that this approach achieves 93% accuracy on the TREC data, about 12% improvement from the rule-based algorithm reported by De Boni and Mananhar (2005). Moreover, this data driven approach requires much less human effort on investigating a specific data set and less human expertise to summarize rules from the observation. All the features we used in the training can be automatically extracted. This makes it straightforward to train a model in a new domain, such as the HandQA. Furthermore, decision tree learning is a white-box model and the trained tree is human interpretable. It shows that the *path* measure has the best information gain among the other semantic similarity measures. We also report our preliminary experiment results on context information fusion for question answering.

7 Acknowledgement

The authors thank Srinivas Bangalore and Mazin E. Gilbert for helpful discussion.

References

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th ACL*.

Marco De Boni and Suresh Manandhar. 2005. Implementing clarification dialogues in open domain question answering. *Natural Language Engineering*. Accepted.

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

Erik Hatcher and Otis Gospodnetic. 2004. *Lucene in Action*. Manning.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of 32nd ACL*, pages 9–16.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning*.

Siddharth Patwardhan. 2003. Incorporating dictionary and corpus information into a context vector measure of semantic relatedness. master's thesis, University of Minnesota, Duluth.

Ted Pederson, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the 9th AAAI*. Intelligent Systems Demonstration.

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2004. A public reference implementation of the rap anaphora resolution algorithm. In *Proceedings of LREC*, pages 291–294.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 question answering track. In *Proceedings of TREC-10*.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proceedings of TREC-13*.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of 32nd ACL*, pages 133–138.