Using Machine Learning Techniques to Interpret WH-questions

Ingrid Zukerman
School of Computer Science and Software Engineering
Monash University
Clayton, Victoria 3800, AUSTRALIA
ingrid@csse.monash.edu.au

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
horvitz@microsoft.com

Abstract

We describe a set of supervised machine learning experiments centering on the construction of statistical models of WH-questions. These models, which are built from shallow linguistic features of questions, are employed to predict target variables which represent a user's informational goals. We report on different aspects of the predictive performance of our models, including the influence of various training and testing factors on predictive performance, and examine the relationships among the target variables.

1 Introduction

The growth in popularity of the Internet highlights the importance of developing machinery for generating responses to queries targeted at large unstructured corpora. At the same time, the access of World Wide Web resources by large numbers of users provides opportunities for collecting and leveraging vast amounts of data about user activity. In this paper, we describe research on exploiting data collected from logs of users' queries in order to build models that can be used to infer users' informational goals from queries.

We describe experiments which use supervised machine learning techniques to build statistical models of questions posed to the Web-based Encarta encyclopedia service. We focus on models and analyses of complete questions phrased in English. These models predict a user's informational goals from shallow linguistic features of questions obtained from a natural language parser. We decompose these goals into (1) the type of information requested by the user (e.g., definition, value of an attribute, explanation for an event), (2) the topic, focal point and additional restrictions posed by the question, and (3) the level of detail of the answer. The long-term aim of this project is to use predictions of these informational goals to enhance the performance of information-retrieval and question-answering systems. In this paper, we report on different aspects of the predictive performance of our statistical models, including the influence of various training and testing factors on predictive performance, and examine the relationships among the informational goals.

In the next section, we review related research. In Section 3, we describe the variables being modeled. In Section 4, we discuss our predictive models. We then evaluate the predictions obtained from models built under different training and modeling conditions. Finally, we summarize the contribution of this work and discuss research directions.

2 Related Research

Our research builds on earlier work on the use of probabilistic models to understand free-text queries in search applications (Heckerman and Horvitz, 1998; Horvitz et al., 1998), and on work conducted in the IR arena of question answering (QA) technologies.

Heckerman and Horvitz (1998) and Horvitz *et al.* (1998) used hand-crafted models and supervised learning to construct Bayesian models that predict users' goals and needs for assistance in the context of consumer software applications. Heckerman and Horvitz' models considered words, phrases and linguistic structures (e.g., capitalization and definite/indefinite articles) appearing in queries to a help system. Horvitz *et al.*'s models considered a user's recent actions in his/her use of software, together with probabilistic information maintained in a dynamically updated user profile.

QA research centers on the challenge of enhancing the response of search engines to a user's questions by returning precise answers rather than returning documents, which is the more common IR goal. QA systems typically combine traditional IR statistical methods (Salton and McGill, 1983) with "shallow" NLP techniques. One approach to the QA task consists of applying the IR methods to retrieve documents relevant to a user's question, and then using the shallow NLP to extract features from both the user's question and the most promising retrieved documents. These features are then used to identify an answer within each document which best matches the user's question. This approach was adopted in (Kupiec, 1993; Abney et al., 2000; Cardie et al., 2000; Moldovan et al., 2000).

The NLP components of these systems employed hand-crafted rules to infer the type of answer expected. These rules were built by considering the first word of a question as well as larger patterns of words identified in the question. For example, the question *"How far is Mars?"* might be characterized as requiring a reply of type DISTANCE. Our work differs from traditional QA research in its use of statistical models to predict variables that represent a user's informational goals. The variables under consideration include the type of the information requested in a query, the level of detail of the answer, and the parts-of-speech which contain the topic the query and its focus (which resembles the type of the expected answer). In this paper, we focus on the predictive models, rather than on the provision of answers to users' questions. We hope that in the short term, the insights obtained from our work will assist QA researchers to fine-tune the answers generated by their systems.

3 Data Collection

Our models were built from questions identified in a log of Web queries submitted to the Encarta encyclopedia service. These questions include traditional *WH-questions*, which begin with "what", "when", "where", "which", "who", "why" and "how", as well as imperative statements starting with "name", "tell", "find", "define" and "describe". We extracted 97,640 questions (removing consecutive duplicates), which constitute about 6% of the 1,649,404 queries in the log files collected during a period of three weeks in the year 2000. A total of 6,436 questions were tagged by hand. Two types of tags were collected for each question: (1) tags describing linguistic features, and (2) tags describing high-level informational goals of users. The former were obtained automatically, while the latter were tagged manually.

We considered three classes of linguistic features: word-based, structural and hybrid.

Word-based features indicate the presence of specific words or phrases in a user's question, which we believed showed promise for predicting components of his/her informational goals. These are words like "make", "map" and "picture".

Structural features include information obtained from an XML-encoded parse tree generated for each question by NLPWin (Heidorn, 1999) – a natural language parser developed by the Natural Language Processing Group at Microsoft Research. We extracted a total of 21 structural features, including the number of distinct parts-of-speech (PoS) – NOUNs, VERBs, NPs, etc – in a question, whether the main noun is plural or singular, which noun (if any) is a proper noun, and the PoS of the head verb post-modifier.

Hybrid features are constructed from structural and word-based information. Two hybrid features were extracted: (1) the type of head verb in a question, e.g., "know", "be" or `action verb`; and (2) the initial component of a question, which usually encompasses the first word or two of the question, e.g., "what", "when" or "how many", but for "how" may be followed by a PoS, e.g., "how ADVERB" or "how ADJECTIVE."

We considered the following variables representing high-level informational goals: *Information Need*, *Coverage Asked*, *Coverage Would Give*, *Topic*, *Focus*, *Restriction* and *LIST*. Information about the state of these variables was provided manually by three people, with the majority of the tagging being performed under contract by a professional outside the research team.

Information Need is a variable that represents the type of information requested by a user. We provided fourteen types of information need, including `Attribute`, `IDentification`, `Process`, `Intersection` and `Topic Itself` (which, as shown in Section 5, are the most common information needs), plus the additional category `OTHER`. As examples, the question "*What*

is a hurricane?" is an `IDentification` query; *"What is the color of sand in the Kalahari?"* is an `Attribute` query (the attribute is "color"); *"How does lightning form?"* is a `Process` query; *"What are the biggest lakes in New Hampshire?"* is an `Intersection` query (a type of `IDentification`, where the returned item must satisfy a particular *Restriction* – in this case "biggest"); and *"Where can I find a picture of a bay?"* is a `Topic Itself` query (interpreted as a request for accessing an object directly, rather than obtaining information about the object).

Coverage Asked and ***Coverage Would Give*** are variables that represent the level of detail in answers. *Coverage Asked* is the level of detail of a direct answer to a user's question. *Coverage Would Give* is the level of detail that an information provider would include in a helpful answer. For instance, although the direct answer to the question *"When did Lincoln die?"* is a single date, a helpful information provider might add other details about Lincoln, e.g., that he was the sixteenth president of the United States, and that he was assassinated. This additional level of detail depends on the request itself and on the available information. However, here we consider the former factor, viewing it as an initial filter that will guide the content planning process of an enhanced QA system. The distinction between the requested level of detail and the provided level of detail makes it possible to model questions for which the preferred level of detail in a response differs from the detail requested by the user. We considered three levels of detail for both coverage variables: `Precise`, `Additional` and `Extended`, plus the additional category `OTHER`. `Precise` indicates that an exact answer has been requested, e.g., a name or date (this is the value of *Coverage Asked* in the above example); `Additional` refers to a level of detail characterized by a one-paragraph answer (this is the value of *Coverage Would Give* in the above example); and `Extended` indicates a longer, more detailed answer.

Topic, Focus and ***Restriction*** contain a PoS in the parse tree of a user's question. These variables represent the topic of discussion, the type of the expected answer, and information that restricts the scope of the answer, respectively. These variables take 46 possible values, e.g., $\texttt{NOUN}_1$, $\texttt{VERB}_3$ and $\texttt{NP}_2$, plus the category `OTHER`. For each question, the tagger selected the most specific PoS that contains the portion of the question which best matches each of these informational goals. For instance, given the question *"What are the main traditional foods that Brazilians eat?"*, the *Topic* is $\texttt{NOUN}_2$ (*Brazilians*), the *Focus* is $\texttt{ADJ}_3\texttt{+NOUN}_1$ (*traditional foods*) and the restriction is $\texttt{ADJ}_2$ (*main*). As shown in this example, it was sometimes necessary to assign more than one PoS to these target variables. At present, these composite assignments are classified as the category `OTHER`.

LIST is a boolean variable which indicates whether the user is looking for a single answer (`False`) or multiple answers (`True`).

4 Predictive Model

We built decision trees to infer high-level informational goals from the linguistic features of users' queries. One decision tree was constructed for each goal: *Information Need, Coverage Asked, Coverage Would Give, Topic, Focus, Restriction* and *LIST*. Our decision trees were built using `dprog` (Wallace and Patrick, 1993) – a procedure based on the Minimum Message Length principle (Wallace and Boulton, 1968).

The decision trees described in this section are those that yield the best predictive performance (obtained from a training set comprised of "good" queries, as described Section 5). The trees themselves are too large to be included in this paper. However, we describe the main attributes identified in each decision tree. Table 2 shows, for each target variable, the size of the decision tree (in number of nodes) and its maximum depth, the attribute used for the first split, and the attributes used for the second split. Table 1 shows examples and descriptions of the attributes in Table 2.[1]

We note that the decision tree for *Focus* splits first on the initial component of a question, e.g., "how ADJ", "where" or "what", and that one of the second-split attributes is the PoS following the initial component. These attributes were also used to build the hand-crafted rules employed by the QA systems described in Section 2, which concentrate on determining the type of the expected

[1]The meaning of "Total PRONOUNS" is peculiar in our context, because the NLPWin parser tags words such as "what" and "who" as PRONOUNs. Also, the clue attributes, e.g., `Comparison clues`, represent groupings of different clues that at design time where considered helpful in identifying certain target variables.

Table 1: Attributes in the decision trees

Attribute	Example/Meaning
`Attribute clues`	e.g., “name”, “type of”, “called”
`Comparison clues`	e.g., “similar”, “differ”, “relate”
`Intersection clues`	superlative ADJ, ordinal ADJ, relative clause
`Topic Itself clues`	e.g., “show”, “picture”, “map”
`PoS after Initial component`	e.g., `NOUN` in “which country is the largest?”
`verb-post-modifier PoS`	e.g., `NP without PP` in “what is a choreographer”
`Total` *PoS*	number of occurrences of *PoS* in a question, e.g., `Total NOUNs`
`First NP plural?`	Boolean attribute
`Definite article in First NP?`	Boolean attribute
`Plural quantifier?`	Boolean attribute
`Length in words`	number of words in a question
`Length in phrases`	number of NPs + PPs + VPs in a question

Table 2: Summary of decision trees

Target Variable	Nodes/Depth	First Split	Second Split
Information Need	207/13	`Initial component`	`Attribute clues, Comparison clues, Topic Itself clues, PoS after Initial component, verb-post-modifier PoS, Length in words`
Coverage Asked	123/11	`Initial component`	`Topic Itself clues, PoS after Initial component, Head verb`
Coverage Would Give	69/6	`Topic Itself clues`	`Initial component, Attribute clues`
Topic	193/9	`Total NOUNs`	`Total ADJs, Total AJPs, Total PRONOUNs`
Focus	226/10	`Initial component`	`Topic Itself clues, Total NOUNs, Total VERBs, Total PRONOUNs, Total VPs, Head verb, PoS after Initial component`
Restriction	126/9	`Total PPs`	`Intersection clues, PoS after Initial component, Definite article in First NP?, Length in phrases`
LIST	45/7	`First NP plural?`	`Plural quantifier?, Initial component`

answer (which is similar to our *Focus*). However, our *Focus* decision tree includes additional attributes in its second split (these attributes are added by `dprog` because they improve predictive performance on the training data).

5 Results

Our report on the predictive performance of the decision trees considers the effect of various training and testing factors on predictive performance, and examines the relationships among the target variables.

5.1 Training Factors

We examine how the quality of the training data and the size of the training set affect predictive performance.

Quality of the data. In our context, the quality of the training data is determined by the wording of the queries and the output of the parser. For each query, the tagger could indicate whether it was a BAD QUERY or whether a WRONG PARSE had been produced. A BAD QUERY is incoherent or articulated in such a way that the parser generates a WRONG PARSE, e.g., “*When its hot it expand?*”. Figure 1 shows the predictive performance of the decision trees built for two training sets: `All5145` and `Good4617`. The first set contains 5145 queries, while the second set contains a subset of the first set comprised of “good” queries only (i.e., bad queries and queries with wrong parses were excluded). In both cases, the same 1291 queries were used for testing. As a baseline measure, we also show the predictive ac-

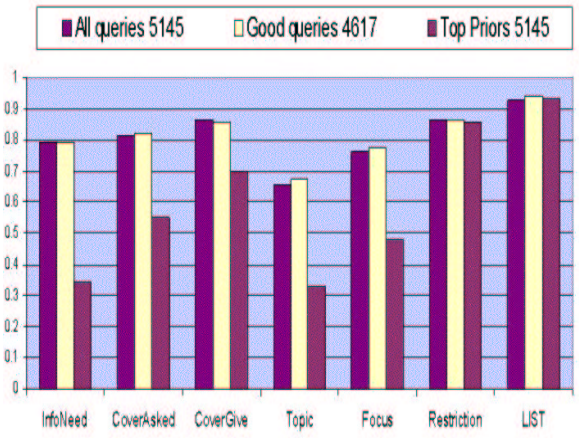

Figure 1: Performance comparison: training with all queries versus training with good queries; prior probabilities included as baseline

	Small	Medium	Large	X-Large
Train/All	1878	2676	3765	5145
Train/Good	1679	2389	3381	4617
Test	376	662	934	1291

Table 3: Four training and testing set sizes

curacy of using the maximum prior probability to predict each target variable. These prior probabilities were obtained from the training set `All5145`. The *Information Need* with the highest prior probability is `IDentification`, the highest *Coverage Asked* is `Precise`, while the highest *Coverage Would Give* is `Additional`; $\texttt{NOUN}_1$ contains the most common *Topic*; the most common *Focus* and *Restriction* are `NONE`; and *LIST* is almost always `False`. As seen in Figure 1, the prior probabilities yield a high predictive accuracy for *Restriction* and *LIST*. However, for the other target variables, the performance obtained using decision trees is substantially better than that obtained using prior probabilities. Further, the predictive performance obtained for the set `Good4617` is only slightly better than that obtained for the set `All5145`. However, since the set of good queries is 10% smaller, it is considered a better option.

Size of the training set. The effect of the size of the training set on predictive performance was assessed by considering four sizes of training/test sets: Small, Medium, Large, and X-Large. Table 3 shows the number of training and test queries for each set size for the "all queries" and the "good queries" training conditions.

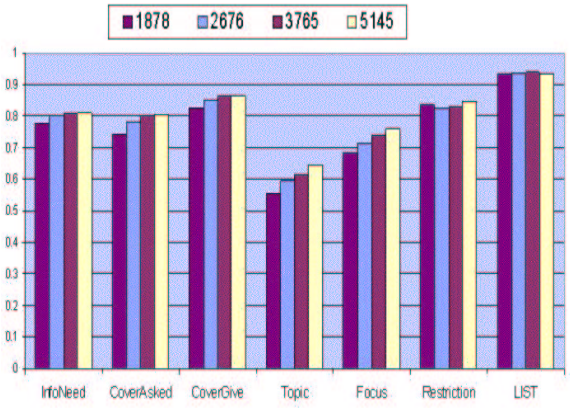

Figure 2: Predictive performance for four training sets (1878, 2676, 3765 and 5145) averaged over 5 runs – All queries

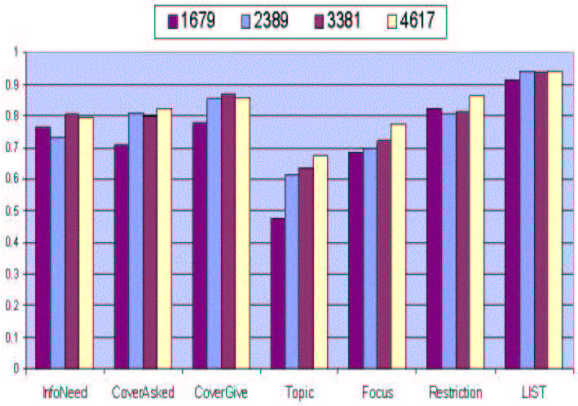

Figure 3: Predictive performance for four training sets (1679, 2389, 3381 and 4617) – Good queries

The predictive performance for the all-queries and good-queries sets is shown in Figures 2 and 3 respectively. Figure 2 depicts the average of the results obtained over five runs, while Figure 3 shows the results of a single run (similar results were obtained from other runs performed with the good-queries sets). As indicated by these results, for both data sets there is a general improvement in predictive performance as the size of the training set increases.

5.2 Testing Factors

We examine the effect of two factors on the predictive performance of our models: (1) query length (measured in number of words), and (2) information need (as recorded by the tagger). These effects were studied with respect to the predictions generated by the decision trees obtained from the set `Good4617`, which had the best performance overall.

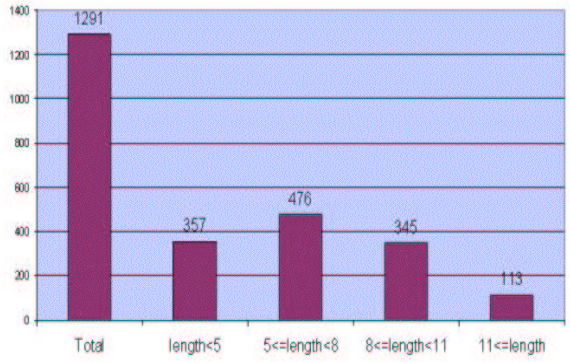

Figure 4: Query length distribution – Test set

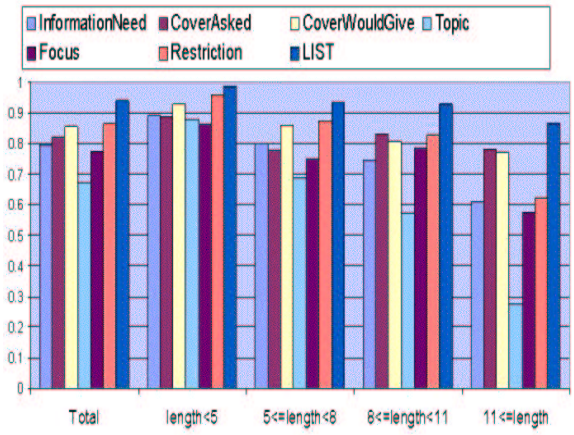

Figure 5: Predictive performance by query length – Good queries

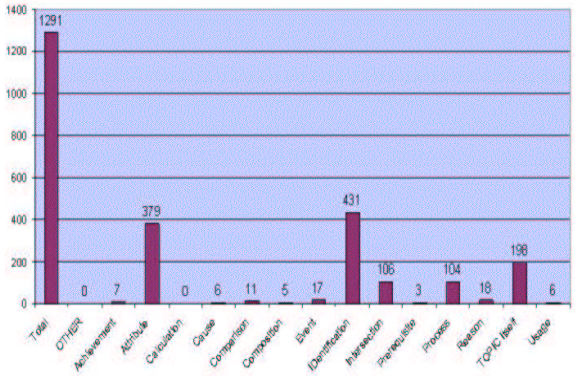

Figure 6: Information need distribution – Test set

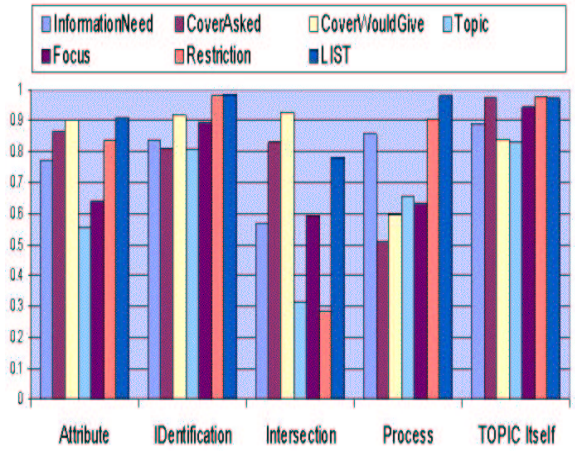

Figure 7: Predictive performance for five most frequent information needs – Good queries

Query length. The queries were divided into four length categories (measured in number of words): length < 5, $5 \leq$ length < 8, $8 \leq$ length < 11 and $11 \leq$ length. Figure 4 displays the distribution of queries in the test set according to these length categories. According to this distribution, over 90% of the queries have less than 11 words. The predictive performance of our decision trees broken down by query length is shown in Figure 5. As shown in this chart, for all target variables there is a downward trend in predictive accuracy as query length increases. Still, for queries of less than 11 words and all target variables except *Topic*, the predictive accuracy remains over 74%. In contrast, the *Topic* predictions drop from 88% (for queries of less than 5 words) to 57% (for queries of 8, 9 or 10 words). Further, the predictive accuracy for *Information Need*, *Topic*, *Focus* and *Restriction* drops substantially for queries that have 11 words or more. This drop in predictive performance may be explained by two factors. For one, the majority of the training data consists of shorter questions. Hence, the applicability of the inferred models to longer questions may be limited. Also, longer questions may exacerbate errors associated with some of the independence assumptions implicit in our current model.

Information need. Figure 6 displays the distribution of the queries in the test set according to *Information Need*. The five most common *Information Need* categories are: `IDentification`, `Attribute`, `Topic Itself`, `Intersection` and `Process`, jointly accounting for over 94% of the queries. Figure 7 displays the predictive performance of our models for these five categories. The best performance is exhibited for the `IDentification` and `Topic Itself` queries. In contrast, the lowest predictive accuracy was obtained for the *Information Need*, *Topic* and *Restriction* of `Intersection` queries. This can be explained by the observation that `Intersection` queries tend to be the longest queries (as seen above, predictive accuracy drops for long

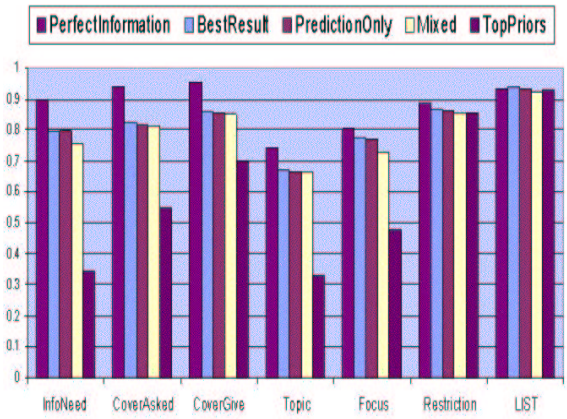

Figure 8: Performance comparison for four prediction models: `PerfectInformation`, `BestResults`, `PredictionOnly` and `Mixed`; prior probabilities included as baseline

queries). The relatively low predictive accuracy obtained for both types of *Coverage* for `Process` queries remains to be explained.

5.3 Relations between target variables

To determine whether the states of our target variables affect each other, we built three prediction models, each of which includes six target variables for predicting the remaining variable. For instance, *Information Need*, *Coverage Asked*, *Coverage Would Give*, *Focus*, *Restriction* and *LIST* are incorporated as data (in addition to the observable variables) when training a model that predicts *Focus*. Our three models are: `PredictionOnly` – which uses the *predicted* values of the six target variables both for the training set and for the test set; `Mixed` – which uses the actual values of the six target variables for the training set and their predicted values for the test set; and `PerfectInformation` – which uses actual values of the six target variables for both training and testing. This model enables us to determine the performance boundaries of our methodology in light of the currently observed attributes.

Figure 8 shows the predictive accuracy of five models: the above three models, our best model so far (obtained from the training set `Good4617`) – denoted `BestResult`, and prior probabilities. As expected, the `PerfectInformation` model has the best performance. However, its predictive accuracy is relatively low for *Topic* and *Focus*, suggesting some inherent limitations of our methodology. The performance of the `PredictionOnly` model is comparable to that of `BestResult`, but the performance of the `Mixed` model seems slightly worse. This difference in performance may be attributed to the fact that the `PredictionOnly` model is a "smoothed" version of the `Mixed` model. That is, the `PredictionOnly` model uses a consistent version of the target variables (i.e., predicted values) both for training and testing. This is not the case for the `Mixed` model, where actual values are used for training (thus the `Mixed` model is the same as the `PerfectInformation` model), but predicted values (which are not always accurate) are used for testing.

Finally, *Information Need* features prominently both in the `PerfectInformation/Mixed` model and the `PredictionOnly` model, being used in the first or second split of most of the decision trees for the other target variables. Also, as expected, *Coverage Asked* is used to predict *Coverage Would Give* and vice versa. These results suggest using modeling techniques which can take advantage of dependencies among target variables. These techniques would enable the construction of models which take into account the distribution of the predicted values of one or more target variables when predicting another target variable.

6 Discussion and Future Work

We have introduced a predictive model, built by applying supervised machine-learning techniques, which can be used to infer a user's key informational goals from free-text questions posed to an Internet search service. The predictive model, which is built from shallow linguistic features of users' questions, infers a user's information need, the level of detail requested by the user, the level of detail deemed appropriate by an information provider, and the topic, focus and restrictions of the user's question. The performance of our model is encouraging, in particular for shorter queries, and for queries with certain information needs. However, further improvements are required in order to make this model practically applicable.

We believe there is an opportunity to identify additional linguistic distinctions that could improve the model's predictive performance. For example, we intend to represent frequent combinations of PoS, such as $\texttt{NOUN}_1$+$\texttt{NOUN}_2$, which are currently classified as `OTHER` (Section 3). We also

propose to investigate predictive models which return more informative predictions than those returned by our current model, e.g., a distribution of the probable informational goals, instead of a single goal. This would enable an enhanced QA system to apply a decision procedure in order to determine a course of action. For example, if the `Additional` value of the *Coverage Would Give* variable has a relatively high probability, the system could consider more than one *Information Need*, *Topic* or *Focus* when generating its reply.

In general, the decision-tree generation methods described in this paper do not have the ability to take into account the relationships among different target variables. In Section 5.3, we investigated this problem by building decision trees which incorporate predicted and actual values of target variables. Our results indicate that it is worth exploring the relationships between several of the target variables. We intend to use the insights obtained from this experiment to construct models which can capture probabilistic dependencies among variables.

Finally, as indicated in Section 1, this project is part of a larger effort centered on improving a user's ability to access information from large information spaces. The next stage of this project involves using the predictions generated by our model to enhance the performance of QA or IR systems. One such enhancement pertains to query reformulation, whereby the inferred informational goals can be used to reformulate or expand queries in a manner that increases the likelihood of returning appropriate answers. As an example of query expansion, if `Process` was identified as the *Information Need* of a query, words that boost responses to searches for information relating to processes could be added to the query prior to submitting it to a search engine. Another envisioned enhancement would attempt to improve the initial recall of the document retrieval process by submitting queries which contain the content words in the *Topic* and *Focus* of a user's question (instead of including all the content words in the question). In the longer term, we plan to explore the use of *Coverage* results to enable an enhanced QA system to compose an appropriate answer from information found in the retrieved documents.

Acknowledgments

This research was largely performed during the first author's visit at Microsoft Research. The authors thank Heidi Lindborg, Mo Corston-Oliver and Debbie Zukerman for their contribution to the tagging effort.

References

S. Abney, M. Collins, and A. Singhal. 2000. Answer extraction. In *Proceedings of the Sixth Applied Natural Language Processing Conference*, pages 296–301, Seattle, Washington.

C. Cardie, V. Ng, D. Pierce, and C. Buckley. 2000. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering system. In *Proceedings of the Sixth Applied Natural Language Processing Conference*, pages 180–187, Seattle, Washington.

D. Heckerman and E. Horvitz. 1998. Inferring informational goals from free-text queries: A Bayesian approach. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 230–237, Madison, Wisconsin.

G. Heidorn. 1999. Intelligent writing assistance. In *A Handbook of Natural Language Processing Techniques*. Marcel Dekker.

E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. 1998. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, Wisconsin.

J. Kupiec. 1993. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 181–190, Pittsburgh, Pennsylvania.

D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. 2000. The structure and performance of an open-domain question answering system. In *ACL2000 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 563–570, Hong Kong.

G. Salton and M.J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw Hill.

C.S. Wallace and D.M. Boulton. 1968. An information measure for classification. *The Computer Journal*, 11:185–194.

C.S. Wallace and J.D. Patrick. 1993. Coding decision trees. *Machine Learning*, 11:7–22.