A BAYESIAN APPROACH FOR UNDERSTANDING INFORMATION-SEEKING QUERIES

Helen M. Meng, Wai Lam and Kon Fan Low

Human-Computer Communications Laboratory
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, N.T.,
Hong Kong, China
{hmmeng, wlam, kflow}@se.cuhk.edu.hk

ABSTRACT

This paper explores a Bayesian approach for handling natural language queries for information access or retrieval. Natural language understanding is a key technology in human-computer interfaces, and often constitutes the front-end to information systems. In this work, we devised a Bayesian methodology for the identification of the user's informational goals in information enquiries. We focus on a restricted domain, the Air Travel Information Systems (ATIS) domain, where the user's informational goal often falls within a finite set of within-domain goals. We formulated the problem into *N* binary decisions, to allow for the identification of queries with multiple goals, as well as queries with out-of-domain goals. Experiments with the ATIS corpus shows that between 84.6% to 88.0% of the user queries are correctly handled via goal classification, rejection or multiple-goal identification.

I. INTRODUCTION

A human-computer interface that enables the user to query for information by means of natural language is considered desirable by many. This is because human-human communication largely utilizes the natural language medium. Understanding a natural language query requires the identification of the user's informational goal. For a restricted domain, the informational goal(s) of the user's query should belong to a finite set of within-domain goals most of the time. However, the possible ways of expressing a goal are legion. Therefore, natural language understanding involves identifying the key semantics in the user's query, and from these deduces the underlying informational goal of the query. The ultimate objective is to produce a coherent response.

As an example, we can consider an enormously simplified weather domain, which only consists of three semantic concepts: <weather>, <location> and <date>. A query, which specifies two of the three concepts, is likely to be asking for the missing one, e.g. "What do we expect for Hong Kong tomorrow?" A query containing all three concepts is likely to be asking for a yes/no response, e.g. "Will there be sunshine in Hong Kong tomorrow?" Another example is call routing in AT&T's "How May I Help You?" task [1] and other similar call center tasks [2]. Here the caller's informational goal determines the destination for call routing.

Previous approaches to this problem include the use of heuristics to map a parsed query into an interpretation. The "parse" may be the output of a grammar-based parser, e.g. [3,4] or a stochastic concept decoder; e.g. HMMs in [5], or probabilistic recursive transition networks in [6]. Here the interpretation adopted depends primarily on an evaluation among parse alternatives, and sometimes the heuristics may not identify the best interpretation if task knowledge were to be considered. As a result, [4] had proposed a "beam of interpretations" approach, where multiple interpretations from multiple parses are used.

In this work, we use a semantic tagger to transform the input query into a sequence of semantic concepts. These form the input to our naïve Bayesian approach for classifying the informational goal of our input query. We favor the probabilistic nature of this approach because it provides an attractive alternative to handcrafting the heuristics between parses and their interpretations, especially under situations when massive data is involved. The naïve Bayesian classifier has previously been applied to generate a filtering profile on the World Wide Web. The system was trained on Web pages labeled to be of interest to the user. The system can then automatically filter for other Web pages according to the user's interest. [7]

II. THE ATIS TASK DOMAIN

We have chosen the ATIS (Air Travel Information System) domain [8], which is a common task in the ARPA (Advanced Research Projects Agency) Speech and Language Program in the USA.

Our experiments are based on the Class A sentences of the ATIS-3 corpus, with disjoint training and test sets of 1564 and 448 (1993 test set) transcribed utterances respectively. Each utterance (or query) is accompanied with its corresponding SQL query for retrieving the relevant information. Thus we derive the informational goal for each utterance from the main attribute label of its SQL query. In our training set, we counted a total of 32 informational goals. We also found 43 training utterances with more than one informational goal. Examples include:

QUERY: *"chicago to san francisco on continental"*
GOAL: FLIGHT_ID
QUERY: *"give me the least expensive first class round trip ticket on u s air from cleveland to miami"*
GOALS: FLIGHT_ID, FARE_ID

III. SEMANTIC TAGGING

Semantic tagging abstracts the words in a query into a set of semantic concepts. While the main attribute label(s) in the SQL query is adopted as the informational goal(s), the remaining attribute labels are identified as key semantic concepts for the

ATIS domain, and served as a reference when we design our semantic tags for labeling an input transcription. We have a total of 60 hand-designed semantic tags. We have also devised an automatic procedure for discovering such semantic categories from unannotated corpora [9].

Training utterances are automatically tagged, using a two-pass transformational procedure. This identifies the semantic concepts in the utterance transcriptions. The following shows an example of an utterance transcription and its corresponding tags:

QUERY: *what are the dinner flights from indianapolis to san diego on Wednesday may twelfth*
TAGS: <what><dummy><meal_description><flight> <from><city_name><to><city_name> <prep> <day_name><month><day>

Henceforth each training query is represented by its annotated goal and a sequence of semantic concepts. These are used as training data for Bayesian classification, described in the following.

IV. *N* BINARY DECISIONS vs. ONE *N*-ARY DECISION

Each natural language query is transformed into a sequence of semantic tags. From these, we need to infer the informational goal of the query, out of the finite set of goals in a restricted domain. One may formulate the problem as *N* binary decisions, or a single *N*-ary decision. We have chosen the former approach to facilitate the identification of cases with *multiple goals*, as well as the rejection of cases with *out-of-domain goals*.

V. THE NAÏVE BAYESIAN APPROACH

Our approach utilizes multiple Bayesian classifiers – each is distinct for making the binary decision regarding a unique goal. Let G_j be the corresponding goal of one of our Bayesian classifiers. Let **C** be the binary vector of semantic concept tags for a given input query. Our classifier tries to make a binary decision based on comparing $P(G_j=1 \mid \mathbf{C})$ and $P(G_j=0 \mid \mathbf{C})$, where G_j equals 1 if the goal is present, and 0 if absent. The two *aposteriori* probabilities are computed by applying Baye's Theorem:

$$P(G_j = g|\vec{C}) = \frac{P(\vec{C}|G_j=g)P(G_j=g)}{P(\vec{C})} \ldots\ldots\ldots(1)$$

where g = 0 or 1. Assuming that the vector $\boldsymbol{C}$ has n dimensions, i.e.

$$\vec{C} = \{c_1, c_2, c_3 \ldots c_n\}$$

Equation (1) may be simplified with two assumptions of conditional and marginal independence:

$$P(G_j = g|\vec{C}) = \prod_{i=1}^{n} \frac{P(c_i=c|G_j=g)P(G_j=g)}{P(c_i=c)} \ldots(2)$$

where c = 0 or 1.

We have chosen to fix the dimensionality of the input vectors at n=20, as suggested by our previous experiments [10]. Therefore we have implemented a feature selection algorithm, where the discriminating n concepts most indicative of each goal are selected. The identity of the concept set thus varies from goal to goal, and we will denote the concepts as C_j. We have also dropped the denominator in Equation (2) since it does not contribute towards the discrimination between the presence and absence of G_j. These procedures transforms Equation (2) to:

$$Q(G_j = g|\vec{C}_j = c) = \prod_{i=1}^{n=20} P(c_{ji} = c|G_j = g)P(G_j = g) \ldots(3)$$

V.1 Concept Selection

Concept selection (or input feature selection) for every Bayesian classifier is performed with reference to the Information Gain (IG) metric. IG measures the degree of association between a concept and a goal, taking into consideration both the presence and absence of the entities, as seen in Equation (4):

$$IG(C_k, G_i) = \sum_{c=0,1}\sum_{g=0,1} P(C_k = c, G_i = g)\log\frac{P(C_k = c, G_i = g)}{P(C_k = c)P(G_i = g)} \ldots..(4)$$

Probabilities are estimated by tallying the counts for each combination (absence and presence) of concept and goal in the training data. For each goal G_j, we compute the IG values for all the concepts, and select the set of 20 with the highest IG values: $\{c_{ji}\}$, where $i=1,2,\ldots 20$. These are deemed most indicative of the presence of G_j.

V.2 The Bayesian Classifier's Decision Rule

Each Bayesian classifier makes its binary decision based on the difference of the *aposteriori* probabilities. The decision rule is:

$$Diff = Q(G_i = 1|\vec{C}_j) - Q(G_i = 0|\vec{C}_j) \ldots..(5)$$

$$Diff_j > \theta_j \Rightarrow G_i = 1$$

$$Diff_j \le \theta_j \Rightarrow G_i = 0$$

We hope to achieve high values for precision and recall for G_j. *Precision (P)* is the percentage of queries in the test set correctly classified to belong to G_j, out of all the queries which are classified as G_j. *Recall (R)* is the percentage of queries in the test set correctly classified to belong to G_j, out of all the queries, which truly belong to G_j. We combine both into a single score by optimizing with the *F-measure* [11] as shown in Equation (6).

$$F_\beta(\theta_j) = \frac{(\beta^2+1)PR}{\beta^2 P + R} \ldots\ldots\ldots(6)$$

(β =1 in our experiments to treat P and R with equal importance).

We have also indicated in Equation (6) that the F-measure corresponding to G_j may be dependent on the choice of θ_j. We have investigated this in our experiments, to be described later.

V.3 Overall Goal Identification

Each of our classifiers make a binary decision regarding its corresponding goal G_j. However, our ultimate objective is to identify the informational goal(s) of our input query. To this end, we have to combine all the binary decisions into an overall decision. Assuming we have built M binary classifiers, we devised the following schemes for combination:

Single-Goal Scheme:

(i) The Rejection Step: If all binary Bayesian classifiers vote negative for the input query, it is considered *out-of-domain* (i.e. it does not belong to any prescribed domain-specific goals) and *rejected*.

(ii) Single Goal Identification: If one or more binary classifiers vote positive for the input query, we will label the query with goal G_k, where:

$$G_k = \arg\max_{n=1}^{M} Diff_n \ldots..(7)$$

and n indexes across all goals. Hence we will only identify a single output goal even for queries which really have multiple goals.

Multiple-Goal Scheme:

(i) The Rejection Step: If all binary Bayesian classifiers vote negative for the input query, it is considered out-of-domain (OOD), same as the other scheme.

(ii) Multiple-Goal Identification: If one or more binary classifiers vote positive for the input query, all the corresponding goals are considered as the identified goal(s). No maximization step is taken.

VI. EXPERIMENTS

Our training data has 32 unique goals in total, and we have indexed them from 1 to 32. Inspection of the training set reveals that only 11 goals are instantiated 10 times or more. These 11 goals cover over 95% of the training set, as listed in Table 1. Consequently, we have constructed *M=11* Bayesian classifiers, to avoid the use of sparsely trained classifiers. The remaining goals and their utterances are treated as out-of-domain.

VI.1 Experimentation with Parameter Setting for Binary Classification

Referring to Equation (5), our classifiers make their binary decisions by comparing the difference in *aposteriori* probabilities, $Diff_i$, against a threshold value, θ_i. A reasonable setting of θ_i may be 0, i.e. if the *aposteriori* probability for $G_i=1$ exceeds that of $G_i=0$, then the input is considered to test positive for G_i (and test negative otherwise). This should give us the corresponding value for the F-measure (F_i *corresponding to* G_i**),** to see how the *ith* binary classifier fares. Alternatively, we may attempt to find an "optimized" value for θ_i, which maximizes F_i based on the training data.

Figure 1 shows the relationship between F_5 and θ_5 for goal 5 (airline_name). For this goal, we chose $\theta_5^{opt} = 0.002$ as the optimized value for θ_5.

Figure 2 compares two settings of $\theta=0$ versus θ^{opt} across all the 11 binary Bayesian Classifiers. Since we tune the values of θ^{opt} using the training data, the F-measures obtained from our disjoint test set is not guaranteed to improve. Nevertheless, Our performance on the test set shows that using θ^{opt} is preferable to $\theta=0$.

Next, we investigate to see if the θ^{opt} will contribute towards improvement in the overall identification accuracy. We considered the *single-goal scheme* (as described in section V.3), and obtained the experimental results as shown in Table 2.

Goal (index)	Goal (name)	Example Query
1	aircraft_code	*what is the smallest aircraft available flying from pittsburgh to baltimore arriving on may seventh*
4	airline_code	*list all the airlines that fly into general mitchell international*
5	airline_name	*could you tell me what abbreviation u s stands for*
6	airport_code	*where does canadian airlines international fly*
8	airport_name	*what does I a h mean*
14	class_description	*what is fare code h*
19	fare_id	*what is the first class fare from indianapolis to orlando*
20	fare_basis_code	*what does m mean*
24	flight_id	*show me the dinner flights from baltimore to oakland*
25	flight_number	*what are the flight numbers of the flights which go from san francisco to washington via indianapolis*
29	city_code	*show me all the cities that midwest express serves*

Table 1. Examples of ATIS queries and their informational goals.

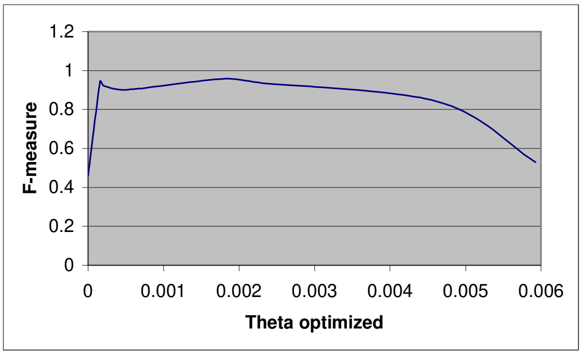

Figure 1. Plot of the F-measures attained at various threshold settings of θ, for the goal 5 (airline_name)

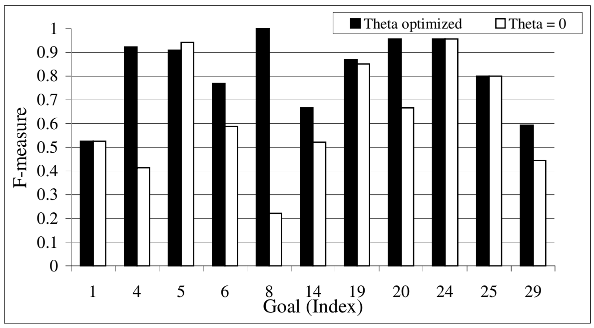

Figure 2. Experimental results comparing $\theta=0$ against θ^{opt} across the 11 binary Bayesian classifiers.

The evidence from Table 2 also suggests that θ^{opt} is preferable to $\theta=0$. There are 8 queries in our test set which are labeled with multiple goals. Since we are using the single-goal scheme to produce a single output goal, we consider the (multiple-goal) test query to be correctly handled if our hypothesized goal matches one of the multiple reference goals. This explains the third row in Table 2. The goal identification of queries with a single reference goal degraded with the user of θ^{opt}, as seen in the fourth row – here all queries are within-domain. The degradation is mainly due to the increased number of incorrectly rejections from 22 for $\theta=0$ to 41 for θ^{opt} (refer to the sixth row of Table 2). However, this degradation is offset by the percentage of correctly rejected queries. These are queries whose reference goals do not belong to our set of 11. Overall, θ^{opt} produced a net improvement in our (test set) goal identification performance, as seen in the last row of Table 2, or a reduction of error rate by 17%.

Observations \ Param. Settings	θ^{opt}	$\theta=0$
Total # queries	448	448
Correctly Identified (multiple goals)	100% (8/8)	100% (8/8)
Correctly Identified (single goals)	88.6% (359/405)	90.9% (368/405)
Incorrect rejection	41	22
Correct rejection	77.1% (27/35)	20.0% (7/35)
Correctly handled	88.0% (394/448)	85.5% (383/448)

Table 2: Experimental results comparing $\theta=0$ against θ^{opt} for overall goal identification (single-goal scheme).

VI.2 Experimentation with Probability Smoothing

In using the IG metric to perform input feature selection for our Bayesian classifiers, we often encounter features which are selected because their *absence* is highly indicative of the *presence* of a goal, i.e. the "negative evidence" for the corresponding goal. This is in contrast to concepts whose *presence* is indicative of the *presence* of the goal, i.e. the "positive evidence". In making the binary decision, a total of 20 pieces of evidence (be it positive or negative) are considered, as shown in Equation (3), which shows the multiplication of 20 probabilities. Among these, should there occur a zero probability value for any piece of evidence, the effect of the all other pieces of evidence will be completely nullified. To avoid this phenomenon, we smoothen the zero probabilities by the following method of estimation:

$$P(C_{ji}=1|G_j=1)$$
$$=\frac{N(C_{ji}=1,G_j=1)*r+1}{N(C_{ji}=1,G_j=1)*r+N(C_{ji}=0,G_j=1)*r+2}\ldots(8)$$

In Equation (8), we have given the concept-goal combinations ($C_{ji}=1$, $G_j=1$) and ($C_{ji}=0$, $G_j=1$) each a floor count of 1, as well as an amplification factor $r(=5)$ for the instances that actually occur, i.e. $N(C_{ji}=1, G_j=1)$ is the number of times when concept C_{ji} and goal G_j both occurred. This technique is applied throughout the estimation of $P(C_{ji}=c \mid G_j = g)$, where c and g can be 0 or 1.

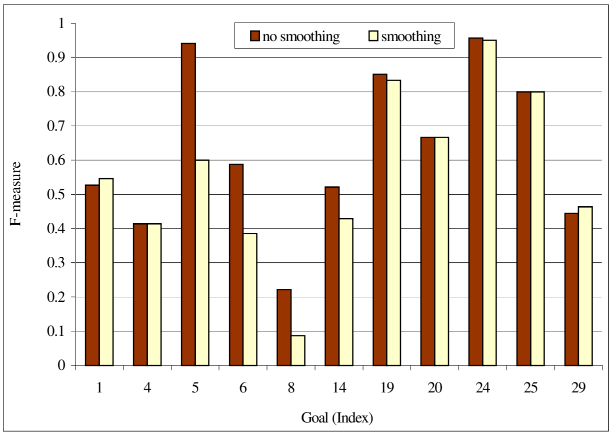

Figure 3. Experiments showing the effect of probability smoothing on F-measures, using $\theta=0$.

Figures 3 and 4 display the effect of probability smoothing on the F-measures, with $\theta=0$ and θ^{opt} respectively. In the case of $\theta=0$, probability smoothing led to F-measure degradations, possibly because the distributions are not as sharp as before for making the binary decisions. This effect is neutralized by the use of θ^{opt}, as shown in Figure 4.

Tables 3 and 4 display the effect of probability smoothing on the overall goal identification performance using the single-goal scheme, with $\theta=0$ and θ^{opt} respectively. Overall goal identification performance takes into account the queries which have been correctly identified, as well as those which have been correctly rejected, out of all the queries in the test set.

In the case of $\theta=0$, probability smoothing caused the F-measures to degrade, but Table 3 shows that the overall goal identification performance improved slightly. This is surprising, as the two effects seem inconsistent. However, detailed investigation reveals that the smoothing process generally led to increase recalls and decreased precisions for the binary decisions. Consequently we have F-measure degradations. However, the increased recall means that the

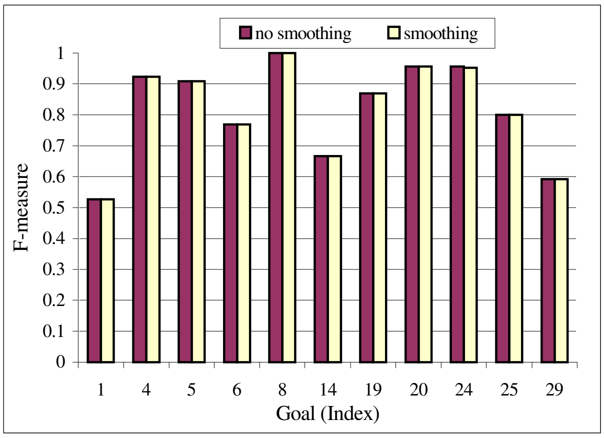

Figure 4. Experiments showing the effect of probability smoothing on F-measures, using θ^{opt}.

single-goal scheme can select the goal with maximum *Diff* from a larger pool of options, which led to 3 additional testing queries attaining correct goal identification (see Table 3).

Param. Settings / Observations	θ=0, no smoothing	θ=0, smoothing
Total number of testing queries	448	448
Correctly identified (multiple-goal queries)	100% (8/8)	100% (8/8)
Correctly identified (single-goal queries)	90.9% (368/405)	90.9% (371/405)
Incorrect rejection	22	17
Correct rejection	20% (7/35)	20% (7/35)
Correctly handled	85.5% (383/448)	86.2% (386/448)

Table 3. Experiments showing the effect of probability smoothing on overall goal identification performance, using $\theta=0$.

	θ^{opt}, no smoothing	θ^{opt}, smoothing
Total number of testing queries	448	448
Correctly identified (multiple goals)	100% (8/8)	100% (8/8)
Correctly identified (single goals)	88.6% (359/405)	88.6% (359/405)
Incorrect rejection	41	40
Correct rejection	77.1% (27/35)	77.1% (27/35)
Correctly handled	88.0% (394/448)	88.0% (394/448)

Table 4. Experiments showing the effect of probability smoothing on overall goal identification performance, using θ^{opt}.

In the case of θ^{opt}, probability smoothing did not change the F-measures, nor did it change the overall goal identification. Table 4 shows that an additional testing query was rejected incorrectly when smoothing was used. This query was incorrectly identified before (without smoothing), and as a consequence there was no change in the overall goal identification performance.

With these results, we find that the process of probability smoothing did not definitively improve our task. Hence we chose to proceed with unperturbed probabilities.

VI.3 Experimentation with Multiple-Goal Identification

We have also experimented with multiple-goal identification, relying on the collective decisions of our 11 binary Bayesian classifiers. Results are shown in Table 5. We have categorized our test set into three portions[1]:

[1] As indicated by the shading in Table 5.

Portion 1 contains 8 queries which are labeled with 2 reference goals (ref.) each. Each of these is counted twice – once for each goal. Hence Table 5 shows the total number of queries as 456 (second row), and the number of queries with multiple goals as 16. Our classifiers have labeled 5 of these 8 queries were labeled with 2 hypothesized goals, and both hypotheses (hyp.) are correct – this is counted as 10 instances of correct labeling (see fourth row). Consequently within Portion I we see 3 instances of deletion errors (DEL) – these are errors where the classifiers have missed a reference goal.

Portion 2 contains 405 within-domain (WD) queries, each labeled with a single reference goal (SG). For some of these queries, our classifiers have provided only a single hypothesized goal, which was incorrect, and these are considered substitution errors (SUB). There are also other queries for which our classifiers have provided *multiple* hypothesized goals, even though there should only be a single reference goal. The additional hypotheses are considered insertion errors (INS), which are penalized in our performance evaluation.

Portion 3 contains 35 out-of-domain (OOD) queries, each labeled with a single reference goal. Again, we observe both substitution and insertion errors in this portion of our testing data set.

Comparing the second and third columns in Table 5, we observe the importance of threshold optimization, i.e. using θ^{opt} instead of $\theta=0$. θ^{opt} creates a more stringent decision criterion before a goal is deemed present in the binary classifier. As a result, we reject many more test queries as OOD. There are also fewer cases of having multiple classifiers vote positive for a given input query. These effects are manifested differently in our data portions:

In Portion 2, θ^{opt} has more SUB than $\theta=0$, because these WD goals are rejected as OOD. This disadvantage is outweighed by the advantage of having fewer INS. In Portion 3, we see again that θ^{opt} produces fewer INS, but SUB are reduced as well, as contributed by the higher number of correct rejections (OOD classified as OOD). Overall, θ^{opt} proved superior to $\theta=0$, by producing a performance accuracy of 84.6% versus 64.5%, which is a 56.6% reduction in error rate.

VII. SUMMARY AND FUTURE WORK

In summary, we have demonstrated the use of a naïve Bayesian approach towards the identification of informational goals of natural language queries. We have formulated the problem in terms of *N* binary Bayesian classification in order to handle multiple-goal queries, as well as out-of-domain queries. Decision threshold optimization was found to produce signification improvement in our goal identification performance accuracy, while probability smoothing did not. Overall, we achieved a test-set accuracy of 88.0% using a single-goal scheme, and 84.6% accuracy using a multiple-goal scheme. Our future direction will be geared towards the use of Bayesian Networks to tackle this problem [10,12].

Param. Settings / Observations	θ^{opt}	$\theta=0$
Total # queries	456	456
Portion 1: 8 Queries with Multiple Goals (MG) 2 refs. each, total count =16		
# queries with 2 hyp., both correct	10	10
# queries with a single, correct hyp. (*i.e.* the *other ref. is a DEL)*	3	3
# errors for MG queries (total count:16)	3 DEL	3 DEL
Portion 2: 405 Single-Goal (SG) Queries, within-domain (WD)		
# queries with a single, correct hyp.	346	315
# queries with a single, incorrect hyp. *(i.e. SUB)*	2	4
# queries with multiple hyp. (in bold) *(only hyp. is correct, others are INS)*	16 (**2**)	57 (**2**) 6 (**3**)
# queries with multiple hyp. (in bold) (*i.e. 1 SUB, others are INS)*	0	1 (**2**)
# queries which are incorrectly rejected *(all hyp. incorrect, i.e. INS)*	41	22
# errors for SG, WD queries (total count: 405)	16 INS 43 SUB	70 INS 27 SUB
Portion 3: 35 Queries with Single Goals (SG), out-of-domain (OOD)		
# queries correctly rejected	27	7
# queries with a single, incorrect hyp. *(i.e. SUB)*	8	9
# queries with multiple hyp. (in bold), *(only one hyp. correct, others are INS)*	0	11 (**2**) 3 (**3**) 3 (**4**) 2 (**5**)
# errors for SG, OOD queries (total count: 35)	8 SUB 0 INS	28 SUB 34 INS
# errors (Overall)	70	162
Correctly Handled Queries (%)	84.6% (386/456)	64.5% (294/456)

Figure 5: Experimental results with multiple-goal hypotheses.

References:

[1] K. Arai, J. Wright, G. Riccardi and A. Gorin, "Grammar Fragment Acquisition using Syntactic and Semantic Clustering," Proceedings of the ICSLP 1998.

[2] B. Carpenter and J. Chu-Carroll, "Natural Language Call Routing: A Robust, Self-Organizing Approach," Proceedings of the ICSLP 1998.

[3] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," Computational Linguistics, Vol. 18, No. 1, 61-86, 1992.

[4] W. Ward and S. Issar, "Recent Improvements in the CMU Spoken Language Understanding System," Proceedings of the ARPA Human Language Technology Workshop, 1994, pp. 213-216.

[5] R. Pieraccini, E. Tzoukermann, Z. Gorelov, J. Gauvain, E. Levin, C. Lee and J. Wilpon, "A Speech Understanding System Based on Statistical Representation of Semantics," Proceedings of ICASSP, 1992, pp. I-193 to I-196.

[6] S. Miller and R. Bobrow, "Statistical Language Processing Using Hidden Understanding Models," Proceedings of the Human Language Technology Workshop, 1994, pp. 278-282.

[7] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting Web sites," Machine Learning, Vol. 27, No. 3, 1997, pp. 313-331.

[8] Price, P., "Evaluation of Spoken Language Systems: The ATIS Domain," Proceedings of the ARPA Human Language Technology Workshop, 1990, pp. 91-95.

[9] K. C. Siu and H. Meng, "Semi-automatic Acquisition of Domain-specific Semantic Structures," Proceedings of Eurospeech, Budapest, 1999, forthcoming.

[10] H. Meng, W. Lam and C. Wai, "To Believe is to Understand," Proceedings of the European Conference on Speech Communication, Budapest, 1999, forthcoming.

[11] van Rijsbergen, C. J., Information Retrieval. London. Butterworth. 1979.

[12] D. Heckerman and E. Horvitz, "Inferring Informational Goals from Free-Text Queries: A Bayesian Approach," Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 230-238.