# Semiautomatic Acquisition of Semantic Structures for Understanding Domain-Specific Natural Language Queries

Helen M. Meng, *Member*, *IEEE*, and Kai-Chung Siu

**Abstract**—This paper describes a methodology for semiautomatic grammar induction from unannotated corpora of information-seeking queries in a restricted domain. The grammar contains both semantic and syntactic structures, which are conducive to (spoken) natural language understanding. Our work aims to ameliorate the reliance of grammar development on expert handcrafting or on the availability of annotated corpora. To strive for reasonable coverage on real data, as well as portability across domains and languages, we adopt a statistical approach. Agglomerative clustering using the symmetrized divergence criterion groups words "spatially." These words have similar left and right contexts and tend to form semantic classes. Agglomerative clustering using mutual information groups words "temporally." These words tend to co-occur sequentially to form phrases or multiword entities. Our approach is amenable to the optional injection of prior knowledge to catalyze grammar induction. The resultant grammar is interpretable by humans and is amenable to hand-editing for refinement. Hence, our approach is semiautomatic in nature. Experiments were conducted using the ATIS (Air Travel Information Service) corpus and the semiautomatically-induced grammar $G_{SA}$ is compared to an entirely handcrafted grammar $G_H$. $G_H$ took two months to develop and gave concept error rates of 7 percent and 11.3 percent, respectively, in language understanding of two test corpora. $G_{SA}$ took only three days to produce and gave concept errors of 14 percent and 12.2 percent on the corresponding test corpora. These results provide a desirable trade-off between language understanding performance and grammar development effort.

**Index Terms**—Grammar induction, semantic processing, natural language understanding, concepts extraction, knowledge acquisition.

## 1 Introduction

Our current age of information is characterized by the convergence of computing, communication, and content. Round-the-clock, ubiquitous access to information and services is increasingly becoming a necessity in our daily lives. It is desirable to develop a human-computer interface which enables a broad range of users to consult computers for electronic information in a variety of application domains. One promising solution is the use of natural language, i.e., to ask verbal questions just as we do in human-human communication. Natural language understanding, (NLU) is the core technology behind natural language interfaces. NLU can be applied as a front-end technology to a search engine for the Web. This will enable both technical and nontechnical users to conduct advanced searches (more powerful than keyword searches) without rote memorization of syntax, e.g., for Boolean expressions. The NLU technology can also be interfaced with speech recognition in human-computer conversational systems, which can handle the user's queries in spoken form. Natural language interfaces, and the NLU technology, will become indispensable in the widespread provision of informational and transactional services, in speech-enabled electronic commerce and other similar applications.

"Understanding" a natural language query refers to the computer's ability to transform the verbal form into machine-readable semantics. In this work, we aim to understand users' information-seeking queries with a degree of semantic precision needed for future incorporation into human-computer conversational systems, i.e., for the human and computer to engage in a spoken language dialog [1]. This is distinct from the NLU technologies intended for processing free-form running text-passages, where a designated meaning frame may be given [2], [3]. Presently, we choose to focus on NLU and regard the integration of speech recognition as a next step, which is beyond the scope of the current work.

NLU involves the extraction of *key concepts* from the query, as well as inferring the *informational goal(s)* therein. State-of-the-art NLU technologies are typically applied to restricted domains in order to limit the scope of understanding. Most approaches involve parsing with a grammar that is handcrafted by a grammarian. The key concepts in an incoming query are derived from its parse tree. The underlying informational goal is in turn obtained by direct mapping according to heuristics designed by a knowledge domain expert. Due to extensive handcrafting and heuristic design in the approach, developing an NLU component for a new domain or a new language often involves significant time and effort on the part of the experts. This forms a major bottleneck in the development of spoken natural language

- *The authors are with the Human-Computer Communications Laboratory, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China. E-mail: {hmmeng, kcsiu}@se.cuhk.edu.hk.*

*Manuscript received 5 Nov. 1999; revised 4 Aug. 2000; accepted 18 Aug. 2000; posted to Digital Library 12 June 2001.*
*For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 110903.*

1041-4347/02/$17.00 © 2002 IEEE

understanding systems. Furthermore, there is no guarantee that the handcrafted grammar will have good coverage of real data when deployed in real applications. Natural language, especially in spoken form, is rife with aggramatical constructs and disfluencies. This may be further complicated with imperfect transcriptions due to speech recognition errors.

To automate the grammar generation process, a corpus-based grammar induction may be adopted as an alternative [4], [5]. Corpus-based approaches are desirable in that the grammar can model real data closely. The corpora are annotated with some domain-dependent semantic tags or domain-independent syntactic tags [6], [7].[1] Various grammar induction algorithms can automatically capture patterns in which syntactic structures and semantic categories interleave into a multitude of surface forms. However, hand-annotation of corpora may also be costly.

We attempt to devise a methodology to semiautomatically capture language structures from *unannotated* corpora. These structures need to be conducive towards language understanding. This is essentially a grammar induction process. The resultant grammar should contain language structures that tightly couple semantics with syntax. Our work is an attempt to expedite the process of grammar design for natural language understanding in a prescribed domain. We conceive of several desirable features for such a methodology:

1. It may be corpus-based, but should ameliorate reliance on annotated corpora.
2. It should be easily portable across different restricted domains, as well as across languages.
3. The output grammar should have reasonable coverage of domain-specific data, and reject out-of-domain data.
4. The output grammar should be intuitive and amenable to interactive refinement by a human.
5. The process should accommodate the *optional* injection of prior knowledge to aid grammar induction.

We conceive of possible applications in data mining, information extraction, and meta-data abstraction.

This paper is organized as follows: Section 2 describes some previous work in semiautomatic NLU. Section 3 details our statistical, data-driven approach. Sections 4 to 7 recounts the application of our approach in the ATIS (Air Travel Information Service) domain. Section 8 presents our experimental results, as well as their analyses. Conclusions and future directions are provided in Section 9.

## 2 Previous Approaches

NLU for restricted domains has been an active area of research since the late 1980s. NLU forms a core component in spoken language systems (SLS), most of which are developed as question-answering systems within the dialog context of a human-computer conversation. Research in NLU was spurred by the launch of the DARPA Spoken Language Systems (SLS) program in the United States, the Esprit SUNDIAL[2] (Speech Understanding and DIALog) and SUNSTAR programs in Europe, [1], [8]. Related projects resulted in the development of SLU components for a number of restricted domains that correspond to real applications. The domains range from air travel (e.g., the Air Travel Information Systems, ATIS [9], train schedules (e.g., Railway Telephone Information Service, RAILTEL [10], ESPIRIT Multimodal-Multimedia Automated Service Kiosk, MASK [11], restaurant guide (e.g., The Berkeley Restaurant Project, BeRP [12], ferry timetables (e.g., WAXHOLM) [13], weather [14] and electronic automobile classifieds [15]. The languages concerned include English and a variety of European languages. Similar research was also initiated recently for Mandarin Chinese in the navigation [16] and banking domains [17].

There are generally, two streams in approaches to NLU, one is primarily rule-based, while the other mainly data-driven. However, there is no harsh distinction between them. Each has its pros and cons and ideas and techniques continue to cross-pollinate between the two streams.

### 2.1 The Rule-Based Approaches

Rule-based approaches generally involve hand-engineering a grammar [12], [13], [18] to be used in parsing (with chart parsers, GLR parsers, finite-state parsers, etc.). Grammars may handle syntax only, semantics only, or a mixture of both. In systems where parsing only involves syntax checking [19], understanding (semantic checking) is performed by a "semantic interpreter." This maps the content words in the parsed constituents into meaningful entries in the semantic frame and the mapping is also based on hand-engineered heuristics. Parsing with a grammar which intermixes syntax and semantics [18] produces a parse tree whose nonterminals may be directly mapped into entries in the semantic frame. To handle disfluencies in spoken queries, *robust parsing* was used [20], [21] to allow the production of partial parsers from fragments of the input, as well as the skipping of nonsensical filled pauses and false starts. Parsed fragments are individually converted into semantic frames, these are subsequently combined by hand-designed heuristics to produce an overall semantic frame for the entire input [20]. An emerging trend for tackling robustness is to write *purely semantic* grammars [22], [23], [24]. These systems gain flexibility by spotting key words and phrases in a query as semantic fragments, which are later combined according to handcrafted heuristics.

The rule-based approach has been demonstrated in a number of systems to achieve NLU for restricted domains. However, handcrafting grammar and heuristics remains an expensive process which requires substantial expertise and time. As the rules need to capture domain-specific knowledge, pragmatics, semantics, and syntactics altogether, it is difficult to write a rule-set that has good coverage of real data without becoming unwieldy. Furthermore, expansion of the scope of the domain, or migration to other domains, often requires significant effort [11]. The only leverage is to reuse prior grammars for new domains whenever appropriate [25].

[1] For example, part-of-speech tags, as in the Penn Treebank [6] and the tagged Brown Corpus [7].

[2] SUNSTAR focuses on the integration and design of speech understanding interfaces.

## 2.2 The Data-Driven Approaches

This approach attempts to decode the semantics of an input query by means of a stochastic model. Examples of this approach include AT&T-CHRONUS [26], BBN-HUM [27], and the LIMSI-CNRS systems [28]. Semantic decoding is accomplished by searching for some meaning $M$ such that $P(M|W)$ is maximized for the word sequence $W$. This approach involves learning the correspondence between designated semantic labels (concepts) and words from a large annotated corpus. A suite of modeling techniques have been applied. For example, Hidden Markov Models (HMMs) were used [26], [28], where words are modeled as observations and concepts are the hidden states. Probabilistic recursive transition networks were also used [27], where understanding involves searching through the state space for the "best" path, which corresponds directly to a meaning tree. In addition, an information-theoretic source-channel model [29] has been used to map spoken language into a formal language especially designed to represent meaning. Finally, decision trees were grown stochastically with reference to annotated training data [30].

Stochastic approaches attempt to circumvent the tedium and expertise required in handcrafting grammar rules. Since model parameters are estimated directly from training data, these approaches tend to have good data coverage. However, stochastic modeling requires large training corpora annotated with semantic units or concepts and performance degrades drastically with sparse training data problems. Therefore, the problems with these approaches are that manual annotation is costly and the acquisition of sufficient amounts of training data may be formidable for some knowledge domains, e.g., the Yellow Pages.

There is also the "phrase-spotting" approach that has emerged in some recent work. It involves the process of automatic phrase extraction using some association or similarity measures, such as, Mutual Information and Kullback-Liebler distance. Some of these are considered to be "salient" phrases that are "significant and frequently co-occurring patterns relevant to the domain-specific subject." These phrases are clustered into "grammar fragments" in [4], which are subsequently used for call-type classification in AT&T's *"How May I Help You?"* telephone application. Call-type classification is achieved by computing and maximizing the association probabilities between the grammar fragment and various call-types. Alternatively, call-type classification may be achieved by vector-based information retrieval techniques applied to keywords [31]. The phrase-spotting approach is also used in a Chinese system for telephone directory assistance in the banking domain [32]. The extracted phrases are clustered and each cluster is labeled with a concept tag name.

# 3 A Semiautomatic, Data-Driven Approach

In this work, we devise a semiautomatic methodology to capture language structures from unannotated corpora. We strive to induce a grammar for natural language understanding in a prescribed domain. Ours is a statistical, data-driven approach, inspired by previous work on language modeling for speech recognition by McCandless and Glass [33]. We would like to extend a similar framework to accomplish understanding of natural language.

An iterative procedure is used to cluster the words from a corpus of sentences in a restricted domain. Clustering is implemented both *spatially* and *temporally*. By spatially clustering, we are grouping words which have similar left contexts as well as right contexts. These clusters generally consist of words with similar semantics. By temporal clustering, we are grouping words which tend to co-occur sequentially. These clusters generally constitute phrases or multiword named entities.

For spatial clustering, we considered the Kullback-Liebler distance [34]—an information-theoretic distance between two probability distributions $p_1$ and $p_2$ (1):

$$D(p_1||p_2) = \sum_{i=1}^{V} p_1(i) \log \frac{p_1(i)}{p_2(i)}, \tag{1}$$

where $V$ is the vocabulary size within the given context. It should be noted that $D(p_1||p_2) = 0$ if $p_1$ and $p_2$ are equivalent. In order to acquire a symmetric distance measure, we used the *divergence* measure (2):

$$Div(p_1, p_2) = D(p_1||p_2) + D(p_2||p_1), \tag{2}$$

All probabilities are estimated by tallying counts from the training sentences, with appropriate smoothing. At the onset of spatial clustering, all the words in the training set (with at least the preset minimum occurrence) are considered pairwise. We compute the "distance" (see (3)) between a pair of words (or word clusters for later iterations), which is the sum of the divergences of probability distributions of the words to the left and right of the entities ($e_1$, $e_2$):

$$Dist(e_1, e_2) = Div(p_1^{left}, p_2^{left}) + Div(p_1^{right}, p_2^{right}). \tag{3}$$

The $N$ most similar pairs are clustered and assigned the spatial cluster label $SC_i$, where $i$ is a counter which increments automatically as spatial clusters are formed. Subsequently, all the words in the training set are substituted with their corresponding spatial cluster label. Spatial clustering is expected to produce semantic categories. After spatial clustering, the process proceeds to temporal clustering.

For temporal clustering, we adopt Mutual Information ($MI$) [24] as our distance measure (see (4)), to indicate the degree of cooccurrence of two consecutive entities (words, or word sequences).

$$MI(e_1, e_2) = P(e_1, e_2) \log \frac{P(e_2|e_1)}{P(e_2)}. \tag{4}$$

Again, only words with at least the minimum number of occurrences are considered. The $N$ pairs of entities with the highest $MI$ are selected to form temporal clusters, which are labeled $TC_i$, where $i$ is a counter which increments automatically as the temporal clusters are formed. The training sentences then undergo a pass whereby appropriate entities are substituted with their $TC$ labels. Temporal clustering is expected to produce phrasal structures. The

TABLE 1
The ATIS-3 Class A Corpus

| | 1993 Training | 1993 Test | 1994 Test |
|---|---|---|---|
| **Transcribed Utterances** | 1,564 | 448 | 444 |

process then alternates to the next iteration of spatial clustering.

Iterative clustering produces a context-free grammar, which is postprocessed with hand-editing. Hence, our approach is semiautomatic. The hand-revision process serves to organize grammar nonterminals ($SC$ and $TC$), identify those that contribute to language understanding and label them with semantically relevant tags. The resultant grammar should reflect the ontology of the domain. The grammar may be evaluated by either comparing the semiautomatically derived language structures with those which are handcrafted, should the latter be available or by implementing a semantic parser which utilizes the grammar to parse data sets, and examining the language understanding performance.

## 4 Experimental Corpus

Our experimental corpus is based on the training and test sets of the ATIS (Air Travel Information Service) domain [9]. ATIS is a common task in the ARPA (Advanced Research Projects Agency) Speech and Language Program in the USA. We used the Class A sentences of the ATIS-3 corpus. ATIS-3 [35] is based on a domain-specific database, the Official Airline Guide (OAG). The corpus of spontaneous speech utterances is divided into disjoint training and test sets, as shown in Table 1. Text transcriptions of these utterances are provided, as well as the corresponding SQL queries for retrieval from the relational database. "Class A" sentences refer to ones whose interpretation is independent of the dialog context. Some examples include:

- *"chicago to san francisco on continental,"*
- *"give me the least expensive first class round trip ticket on u s air from cleveland to miami,"*
- *"what is the smallest aircraft available flying from pittsburgh to baltimore arriving on May seventh?"*

Each query also has its corresponding SQL tag for database retrieval, e.g.:

- *"Show me the northwest flights from detroit to boston on sunday."*
- Select FLIGHT_ID from ORIGIN, DESTINATION where AIRLINE_NAME = "*northwest*" and ORIGIN.CITY_NAME = "*detroit*" and DESTINATION.CITY_NAME = "boston" and DAY_NAME = "*sunday*."

## 5 Unsupervised Agglomerative Clustering

Our unsupervised agglomerative clustering procedure requires two parameters: $M$, the minimum number of occurrences of a word before the procedure will operate and $N$, the number of clusters (merges) produced per iteration. We intend to use $M$ to prevent grammar induction based on sparse data and $N$ to prevent overly aggressive clustering which forms heterogeneous clusters. Based on several experimental trials, we chose to set $M = 5$. We experimented with the number of merges per iteration, $N$, for the values $N = 1$ and $N = 5$.

With $N = 1$, each iteration searches through the space of all entity-pairs, to produce an $SC$ and a $TC$. This process is computationally expensive. Having proceeded through 17 iterations of clustering, our algorithm produced 34 spatial and temporal categories, two of which we deemed irrelevant because their constituents do not form a coherent semantic class for database access. Example rules, including the two irrelevant rules, are shown in the following:

$$\begin{array}{lll} SC_0 & \rightarrow & \text{layover} \mid \text{stopover} \\ SC_3 & \rightarrow & \text{numbers} \mid \text{times} \quad \textit{irrelevant} \\ SC_6 & \rightarrow & \text{cheapest} \mid \text{last} \quad \textit{irrelevant} \\ SC_{10} & \rightarrow & \text{could} \mid \text{can} \\ SC_{12} & \rightarrow & \text{nashville} \mid \text{toronto} \\ TC_0 & \rightarrow & \text{flights from} \\ TC_8 & \rightarrow & \text{round trip} \\ TC_{10} & \rightarrow & \text{show me all} \\ TC_{15} & \rightarrow & \text{los angeles} \\ TC_{16} & \rightarrow & \text{salt lake.} \end{array}$$

Upon investigation, $SC_3$ and $SC_6$ may be pardonable offenses. The words "*numbers*" and "*times*" were merged, due to many instances of "*...flight numbers from...*" and "*... flight times from...*". The words "*cheapest*" and "*last*" were merged, due to many instances of "*...the cheapest flight...*" and "*...the last flight...*".

With $N = 5$, our algorithm produced 47 spatial and temporal categories after only five iterations, equivalent to one fifth of the previous processing time. One might expect to obtain more spatial and temporal categories, because five iterations each with five spatial merges and five temporal merges should produce 50 categories. However, there are cases when multiple merges from the same iteration were collapsed. For example, three of the five proposed merges from one iteration were (*nashville*, *toronto*), (*nashville*, *tampa*), and (*detroit*, *nashville*). In this case, our algorithm produced a single spatial category and, therefore, we are able to quickly generate nonterminals with a greater number of terminals. For example:

$$SC_i \rightarrow \text{nashville} \mid \text{toronto} \mid \text{tampa} \mid \text{detroit.}$$

Similar phenomena emerged for the temporal categories. For example, in one iteration the proposed merges were (*salt lake*), (*lake city*), etc. Our algorithm considers across the proposed merges and looks for cases where the second candidate of a pair coincides with the first candidate of another pair. For these cases, the algorithm exhaustively generates the possible combinations, e.g., "*salt lake city*." All

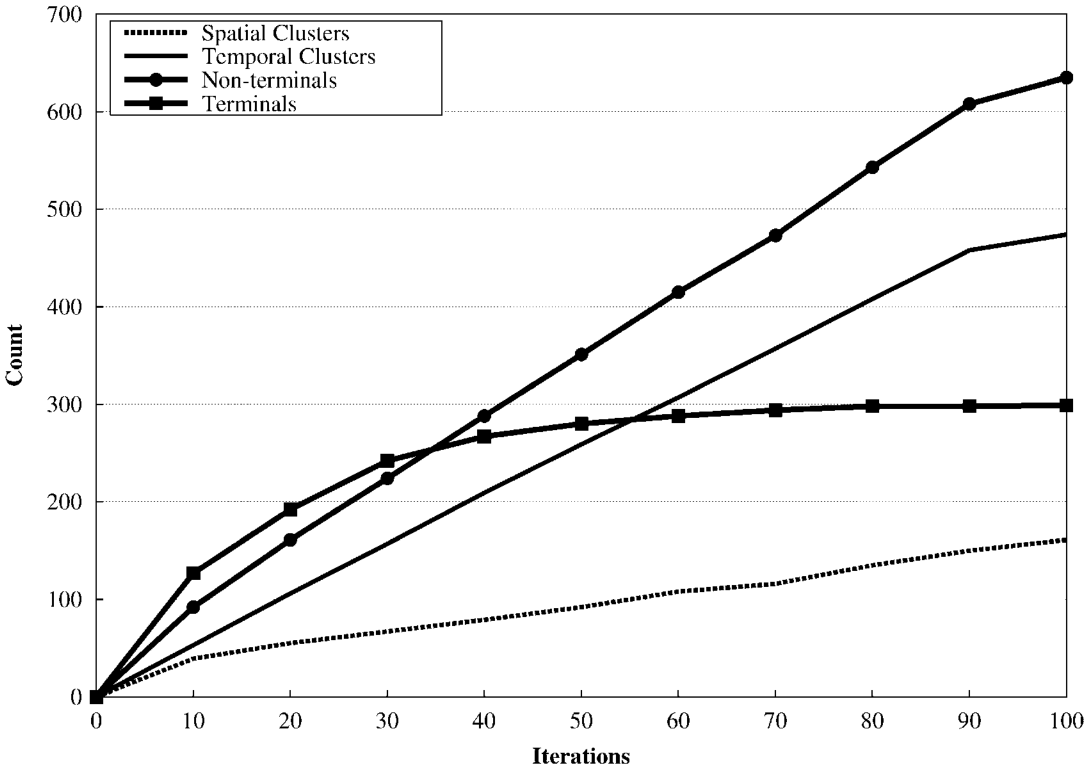


Fig. 1. Growth of grammar units along increasing iterations in the grammar induction process.

the generated combinations are merged first. The merge sequence is in descending order according to the size of combinations. Then, merging of proposed pairs is in decreasing order of $MI$. As a result, we produced $TC_{16} \rightarrow$ *salt lake city*, $TC_{19} \rightarrow$ *salt lake*, but the merge of (*lake city*) is discarded as all the occurrences have been replaced.

Our resultant grammar (47 categories, $N = 5$) is a superset of the previous grammar (34 categories, $N = 1$). The extra 13 categories are all relevant, e.g.,

$$\begin{array}{lcl} SC_{14} & \rightarrow & \text{francisco} \mid \text{jose} \\ SC_{15} & \rightarrow & \text{los angeles} \\ SC_{16} & \rightarrow & \text{san } SC_{14}. \end{array}$$

We concluded that $N = 5$ is a better parameter setting for this domain. The merging is more aggressive and seems to produce an equally good grammar with fewer iterations.

Clustering was allowed to proceed to 100 iterations. We monitored its progress by keeping track of the nonterminal ($SC$ and $TC$) and terminal categories in the grammar (see Fig. 1). As the grammar grows, the number of terminals saturated at around iteration 50, to a count of 280. This covers a fraction of the vocabulary (531 words in all) from the training set. The remaining words are those which did not meet our minimum count requirement. The number of $SCs$ grew slowly to 161 at iteration 100, and they were mainly semantic categories and inflectional variations. The growth rate of $TCs$ dominated the overall growth rate of the nonterminals, reaching 474 $TCs$ at iteration 100.

The $TCs$ were mainly phrasal structure as expected. Examples of $SCs$ and $TCs$ include:

| | | | |
|---|---|---|---|
| $SC_4$ | → | december\|february | *month* |
| $SC_7$ | → | nashville\|toronto\|tampa\|detroit\| $SC_8$ | *city name* |
| $SC_{17}$ | → | june \| march | *month* |
| $SC_{24}$ | → | serve \| serves | *infectional forms* |
| $SC_{28}$ | → | monday \|wednesday \| thursday | *day of week* |
| $TC_{22}$ | → | to $SC_7$ to | *stopover* |
| $TC_{23}$ | → | $SC_7$ to | *origin* |
| $TC_{27}$ | → | to $SC_7$ | *destination* |
| $TC_{39}$ | → | first class | *class type* |
| $TC_{45}$ | → | one way | |
| $TC_{229}$ | → | flights from $SC_7$ to $SC_{12}$ | *a phrase.* |

As we tracked the clustering process, we noticed that within the first 10 iterations, the algorithm has already discovered 11 useful semantic categories as $SCs$, as well as some proper names spanning two words, e.g., "*los angeles*." Between the iterations 10 and 20, only two more useful semantic categories were discovered. The $TCs$ produced at this stage begin to have three words e.g., "new york city," Beyond iteration 20, we begin to see merging of $SCs$ and $TCs$ into phrase fragments e.g.,

$$\begin{array}{lcl} TC_{323} & \rightarrow & SC_{27} \text{ flights from } SC_7 \text{ to} \\ \text{where} & & \\ SC_7 & \rightarrow & \text{nashville} \mid \text{toronto} \mid \text{tampa} \mid \text{detroit} \mid \ldots \\ SC_{27} & \rightarrow & \text{list} \mid \text{show} \mid TC_{28} \text{ (list the)} \mid \\ & & TC_{94} \text{ (show me all the )} \mid \\ & & SC_{37} \mid TC_{38} \text{ (please list the)} \\ SC_{37} & \rightarrow & TC_4 \text{ (show me)} \mid TC_{47} \text{ (what are the),} \end{array}$$

It is noticeable that some automatically discovered categories capture domain-specific knowledge that one could have easily given to the algorithm such as, *city name*,

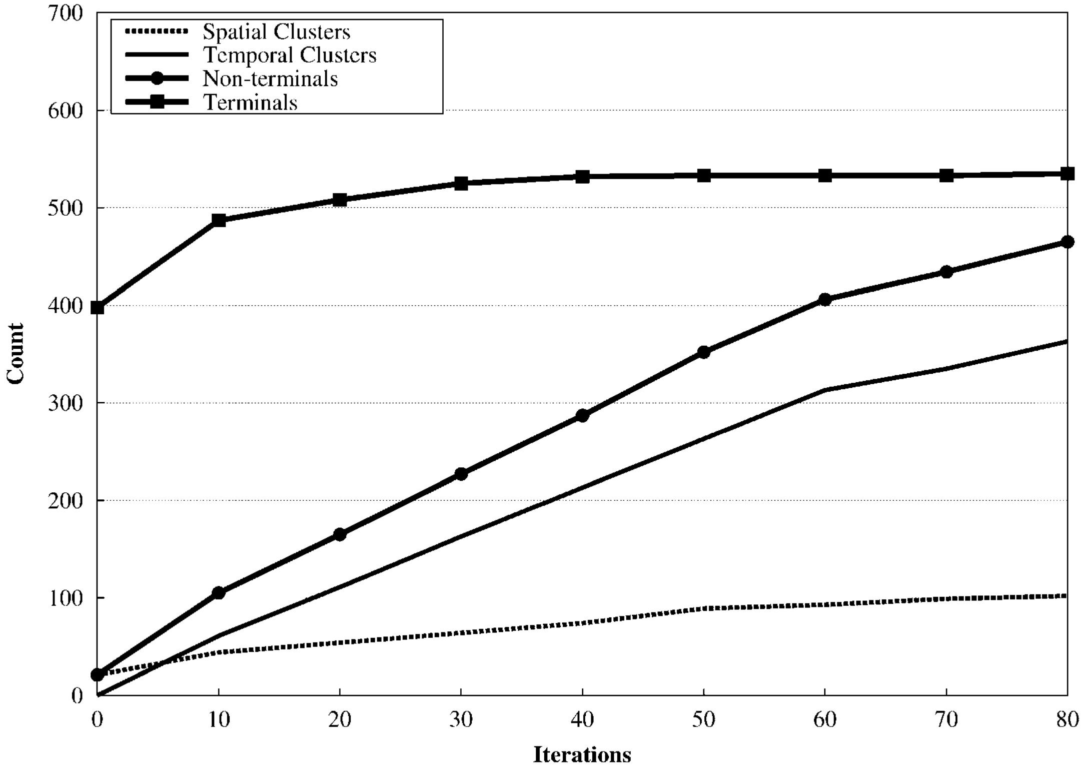


Fig. 2. Growth of grammar units alongside increasing iterations, for grammar induction seeded with prior knowledge.

*digit*, etc. We also observe rules that are close renditions of one another:

$$\begin{aligned} TC_{229} &\rightarrow \text{flights from } SC_7 \text{ to } SC_{12} \\ TC_{289} &\rightarrow \text{flights from } SC_7 \text{ to } SC_{29} \\ TC_{364} &\rightarrow \text{flights from } SC_7 \text{ to } SC_5. \end{aligned}$$

Since $SC_5$, $SC_7$, $SC_{12}$, and $SC_{29}$ are all city names, we manually collapsed $TC_{229}$, $TC_{289}$, and $TC_{364}$ into a single category. We compared our automatically derived clusters ($SCs$ and $TCs$) with the SQL annotations that accompany the training utterances. We found direct relevance between 20 of our clusters with the SQL attribute labels. These 20 categories have previously been automatically discovered and, thus, may not possess the complete set of terminals (e.g., $SC_7$). Alternatively, a single semantic category may be distributed over multiple categories (e.g., $SC_4$ and $SC_{17}$).

These observations prompted our idea of seeding the clustering algorithm with basic domain-specific knowledge. This should serve to jump-start our grammar induction process, and enable the algorithm to proceed further with even fewer iterations.

## 6 Injection of Prior Knowledge

As we referenced the grammar nonterminals formed from unsupervised clustering, we tagged 20 of them with semantic labels (to replace $SC_i$ and $TC_i$). These are basic semantic classes for our domain, which were automatically derived. Our tags include AIRLINE_NAME, AIRPORT_NAME, CITY_NAME, etc. We further compiled these categories to become seed categories ($SC_0$ to $SC_{19}$) for initializing our clustering algorithm. Compilation involves the consolidation of multiple $SCs$ into a single semantic class, as well as completion of the set of terminals belonging to a given nonterminal. Clustering was allowed to run through 80 iterations with $N = 5$. Again, we monitored the grammar inference process—Fig. 2 shows the clustering that was initialized with 398 terminals from the seed categories. These include vocabulary entries below the minimum count of five, as well as inflectional forms of words that have not occurred in the training data. The growth of terminal categories began to saturate within 20 iterations to 508. We manually found iteration 40 to be a suitable termination point, beyond which over-clustering aggravated and produced many heterogeneous groupings. At this point, we recorded 74 $SCs$, 213 $TCs$, and 532 terminals.

Inspection reveals that seed categories catalyzed the formation of longer phrasal structures with fewer iterations. We attempt to illustrate this with the following example rules:

$$\begin{aligned} TC_{153} &\rightarrow SC_3 \text{ flights between } SC_2 \text{ and } SC_2 \\ TC_{165} &\rightarrow SC_{19}\ SC_1 \text{ flights } SC_{16}\ SC_2\ SC_{17}\ SC_2 \\ TC_{186} &\rightarrow \text{flight } SC_{15}\ SC_{16}\ SC_2\ SC_{17}\ SC_2 \\ \text{where} & \\ SC_1 &\rightarrow \text{air canada} \mid \text{alaska airlines} \mid \text{america west} \ldots \\ SC_2 &\rightarrow \text{atlanta} \mid \text{baltimore} \mid \text{boston} \mid \ldots \\ SC_3 &\rightarrow \text{business class} \mid \text{economy} \mid \text{first class} \mid \ldots \\ SC_{15} &\rightarrow SC_{15}\ SC_{15} \mid \text{oh} \mid \text{zero} \mid \text{one} \mid \ldots \\ SC_{16} &\rightarrow \text{from} \\ SC_{17} &\rightarrow \text{to} \\ SC_{19} &\rightarrow \text{list} \mid \text{show} \mid \text{list the} \mid \ldots. \end{aligned}$$

## 7 Postprocessing

The grammar produced from 50 iterations of unsupervised clustering was postprocessed by hand-editing. The editing procedures include:

1. Replacing some of the $SC_i$ and $TC_i$ tags with meaningful labels, e.g., *city name, month*, etc.

   Before : $SC_i \rightarrow$ nashville | toronto | tampa | detroit | ...
   After : CITY_NAME $\rightarrow$ nashville | toronto | tampa | detroit | ....

2. Completing the set of terminals for some categories, e.g., *days of week*.

   Before : $SC_i \rightarrow$ monday | wednesday | thursday
   After : $SC_i \rightarrow$ monday | wednesday | thursday | tuesday | friday | saturday | sunday.

3. Consolidating grammar categories which belong to the same semantic class.

   Before : $SC_i \rightarrow$ december | february
   $SC_j \rightarrow$ june | march
   After : $SC_i \rightarrow$ december | february | june | march.

4. Pruning irrelevant nonterminals and terminals.[3] [4]

   Before : $SC_5 \rightarrow$ e_w_r | m_c_o | $SC_{15}$
   $SC_{15} \rightarrow$ f_f | h_p
   $SC_{21} \rightarrow$ number | tomorrow
   After : $SC_5 \rightarrow$ e_w_r | m_c_o
   $SC_{15} \rightarrow$ f_f | h_p

Such hand-editing produced 20 seed categories for the subsequent process of seeded grammar induction. This time the grammar produced from 40 iterations was also hand-edited using Procedures 1, 2, and 4 described above. Postprocessing took around five hours to produce a grammar with 36 nonterminals and 446 terminals.

## 8 Evaluation

The grammar inferred from 40 iterations was post-processed manually, and this semiautomatically generated grammar ($G_{SA}$) was compared with a handcrafted grammar ($G_H$). $G_{SA}$ and $G_H$ were developed *independently* by two different individuals. $G_H$ took two months to develop, while the development of $G_{SA}$ took only three days. $G_H$ was manually designed to capture the key semantic categories from the training set. $G_{SA}$ has 36 nonterminals and 446 terminals. $G_H$ has 66 nonterminals and 483 terminals. Out of the 36 nonterminals in $G_{SA}$, 20 were inferred seed categories used for semiautomatic grammar induction. This set of nonterminals was also found to be present in a similar form in the handcrafted grammar $G_H$. The remaining 16 nonterminals in $G_{SA}$ were obtained from automatic clustering, and survived pruning during the hand-revision, e.g.,

FLIGHT_NUMBER $\rightarrow$ FLIGHT DIGIT
(e.g.,"*flight four seventeen*")
TIME_VALUE $\rightarrow$ PRE_TIME DIGIT
(e.g.,"*after ten twenty six*").

TABLE 2
The Size of Semiautomatically Generated and Handcrafted Grammars

| Grammar | Non-terminals | Terminals |
|---|---|---|
| Semiautomatically generated ($G_{SA}$) | 36 | 446 |
| Handcrafted ($G_H$) | 66 | 483 |

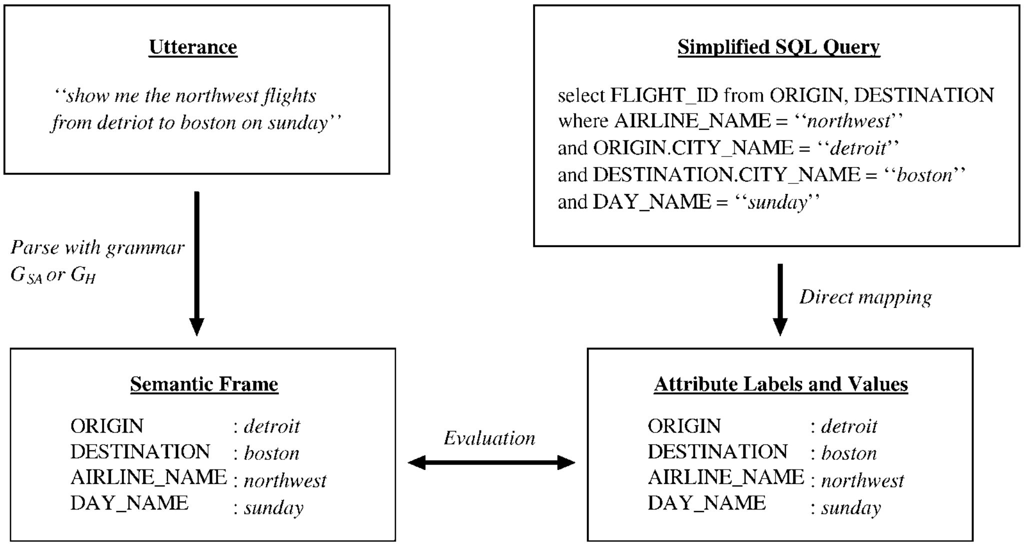
Using each of these grammars, we parsed our data sets to retrieve the key semantic concepts of each query. The $SCs$ in our grammars specify the key semantic categories to be extracted from the utterance and entered into a case frame. These are compared with the reference set of semantic categories from the SQL query (see Table 2).

Fig. 3 illustrates our experimental procedure. In the ATIS-3 corpora, each informational query from the user is accompanied by its reference SQL query for database retrieval. The SQL query provides a list of attribute label-value pairs for our reference and evaluation. For each grammar, we parsed the natural language query to obtain a semantic frame from the parse tree. This is the semantic representation of what was understood from the natural language query. During this experimental process, we can evaluate each grammar in turn based on its coverage of the test set. We can also compare the semantic frame with the reference attribute label-value list, to see how many of the concepts have been correctly extracted.

Results on parse coverage are shown in Table 3. "Full Understanding" refers to utterances with exact matches between the semantic categories in the case frame and those in the SQL. "Partial Understanding" refers to partial matches. "No Understanding" occurs when no semantic categories were extracted, due to out-of-domain words/word sequences. Our results show that $G_H$ has extremely high coverage and accuracy in understanding. Coverage of the $G_{SA}$ is slightly lower and, generally, has a lower rate than $G_H$ in full understanding. This is somewhat compensated by a higher rate in partial understanding. The error rate for test set 1994 is higher, in general, since the training set was collected in 1993 and bears greater resemblance to the test set for 1993.

We also compared the grammars based on concept sequence evaluation. This is based on the evaluation method used in the SUNSTAR program [8] and also similar systems [28]. Concepts that are missing from the semantic frame are regarded as deletions. Additional concepts that appear in the semantic frame, but did not appear in the reference frame are regarded as insertions. The remaining differences are substitutions. The rates of substitution, deletion and insertion are summed to form the overall

[3] $SC_5$ : *airport name.*
[4] $SC_{15}$ : *airline name.*



```
select FLIGHT_ID from ORIGIN, DESTINATION
where AIRLINE_NAME = "northwest"
and ORIGIN.CITY_NAME = "detroit"
and DESTINATION.CITY_NAME = "boston"
and DAY_NAME = "sunday"
```

Fig. 3. Process of evaluating the performance of understanding ATIS queries—each semantic frame generated is compared against the attribute labels from the corresponding SQL query.

error rate. Table 4 shows that the two grammars have comparable training set performance. $G_{SA}$ suffers from degradation in test set performance when compared to $G_H$—this is a sign of overfitting to the training data due to our statistical approach.

Inspection reveals that the main cause of the inferior performance of $G_{SA}$ (lower coverage, higher error rate), when compared to $G_H$, was the absence of a number of semantic grammar rules which can contribute to understanding. These rules were not generated during the automatic grammar induction process because they involved entities whose occurrences were fewer than our minimum count threshold (parameter $M = 5$ in the grammar induction algorithm), e.g.,[5]

| | | |
|---|---|---|
| AIRCRAFT_CODE | → | d ten \| seven fifty seven \| m eighty \|... |
| MANUFACTURER | → | boeing \| mcdonell douglas |
| TRANSPORT_TYPE | → | limousine \| train \| rental car \| ... |

The effect of missing grammar rules propagated and affected higher levels in the grammar structure such as:

AIRCRAFT_INFORMATION→MANFACTURER AIRCRAFT_CODE

e.g.,*"show me all flights from orlando to san diego on a* **boeing seven thirty***"*

AIRCRAFT_INFORMATION→AIRCRAFT AIRCRAFT_CODE

e.g.,*"how many canadian airlines iternational flights use* **aircraft three twenty**."

A second reason for a lower performance in the $G_{SA}$ was due to ambiguities in semantics within the domain. For example, *"washington"* and *"new york"* can either be in category CITY_NAME or STATE_NAME. Our grammar induction algorithm was able to extract temporal clusters such as *"tacoma washington"* and *"westchester county new york"* from utterances such as: However, similar occurrences were too few to allow the induction of the rule:

CITY_STATE_PAIR → CITY_NAME STATE_NAME

*"what flights do you have from burbank to* **tacoma washington***"*

*"flights from denver to* **westchester county new york** *weekdays"*

Additionally, a third reason for lower performance for the $G_{SA}$ was due to the inability to infer more complex rules such as:

| | | |
|---|---|---|
| TIME_RANGE | → | DIGIT CONNECTIVE TIME_VALUE |
| where | | |
| TIME_VALUE | → | DIGIT TIME_UNIT |
| DIGIT | → | DIGIT DIGIT\| zero \| one \|... |

which covers sentences, e.g.,

*"tell me about flights from toronto to salt lake city leaving toronto between* **five thirty and seven pm**."

These factors all contribute towards concept extraction errors in the semantic frame. Such error analyses provide valuable information for grammar postprocessing and the necessary rules can be easily inserted to augment our automatically induced grammar.

We do not know of other work that utilizes a semiautomatic grammar induction approach in similar tasks. Minker [28] has reported on the use of a rule-based approach to produce annotated corpora to train a Hidden Markov Model (HMM) for language understanding. Performance on the 1994 test set gave a concepts error rate of 14.4 percent. A semiautomatic approach was also devised where an HMM is trained on a small amount of data, and used to annotate the entire training data set. The annotation was hand-corrected and the HMM was retrained with the refined annotation. This produced a concept error rate of 13.7 percent on the 1994 test set. Hence, we believe that our

[5] This is a type of aircraft, (see below aircraft code).

TABLE 3
Result of Semantic Parsing Based on the Semiautomatically Generated Grammar ($G_{SA}$) and Handcrafted Grammar ($G_H$)

| | **Training Set** | | **1993 Test Set** | | **1994 Test Set** | |
|---|---|---|---|---|---|---|
| **Understanding** | $G_{SA}$ | $G_H$ | $G_{SA}$ | $G_H$ | $G_{SA}$ | $G_H$ |
| **Full** | 86.9 % | 87.5 % | 80.4 % | 85.5 % | 76.8 % | 78.6 % |
| **Partial** | 13.0 % | 12.5 % | 16.5 % | 14.5 % | 21.8 % | 20.2 % |
| **No** | 0.1 % | 0.0 % | 3.1 % | 0.0 % | 1.4 % | 1.1 % |

semiautomatic grammar induction approach achieves a performance comparable to other similar work.

## 9 Conclusions and Future Work

In this paper, we have presented our semiautomatic approach for grammar induction to achieve language understanding in restricted domains. Our agglomerative clustering technique strives to construct a grammar from unannotated corpora since annotation with semantic tags is both time-consuming and expensive. Our approach is semiautomatic because the inferred grammar is intended to be hand-revised for quality improvement. The clustering algorithm is amenable to initialization with prior domain-specific knowledge to catalyze grammar induction. This initialization produced noticeable improvement in performance.

A handcrafted grammar $G_H$ was designed independently based on the same training corpus. $G_H$ took two months to develop and $G_{SA}$ took three days. These three days include the unsupervised grammar induction run, the seeded grammar induction run, as well as the hand refinement postprocess.

Comparison between $G_{SA}$ with $G_H$ shows that the semiautomatically acquired grammar has good coverage for our two (disjoint) test sets in parsing for understanding. $G_{SA}$ achieved full understanding for 80.4 percent for the 1993 test set and 76.8 percent for the 1994 test set. Corresponding values for $G_H$ are 85.5 percent and 78.6 percent, respectively. Evaluation based on the semantic sequences (concept accuracies) shows comparable training set performance. Test set performances of $G_{SA}$ suffer some degradation, giving 14.0 percent for the 1993 test set and 12.2 percent for the 1994 test set. Corresponding values for $G_H$ are 7.0 percent and 11.3 percent, respectively. Considering the trade-off between development time and grammar performance, these results are encouraging. Future work will be devoted towards improving the grammar quality and examining portability across languages and domains.

TABLE 4
Semantic Sequence Evaluation Based on Parsing with the Semiautomatically Generated Grammar $G_{SA}$ and the Handcrafted Grammar $G_H$

| | **Error Rate of $G_{SA}$** | **Error Rate of $G_H$** |
|---|---|---|
| **Traning Set** | 5.5 % | 5.2 % |
| **1993 Test Set** | 14.0 % | 7.0 % |
| **1994 Test Set** | 12.2 % | 11.3 % |

## Acknowledgments

The authors would like to thank the three anonymous reviewers for their valuable comments which improved this paper.

## References

[1] V. Zue, "Conversational Interfaces: Advances and Challenges," *Proc. Fifth European Conf. Speech Comm. and Technology,* pp. 9-18, Sept. 1997.

[2] *Proc. Fourth Message Understanding Conf.,* 1992.

[3] J.T. Kim and D. Moldovan, "Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction," *IEEE Trans. Knowledge and Data Eng.,* vol. 7, no. 5, Oct. 1995.

[4] K. Arai, J. Wright, G. Riccardi, and A. Gorin, "Grammar Fragment Acquisition Using Syntactic and Semantic Clustering," *Proc. Fifth Int'l Conf. Spoken Language Processing,* Nov. 1998.

[5] S. Chen, "Bayesian Grammar Induction for Language Modeling," *Proc. 33rd Ann. Meeting of the Assoc. Computational Linguistics,* pp. 228-235, June 1995.

[6] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics,* vol. 19, no. 2, pp. 313-330, 1993.

[7] W. Francis and H. Kucera, *Frequency Analysis of English Usage: Lexicon and Grammar.* Houghton Mifflin Co., 1982.

[8] N. Fraser and P. Dalsgaard, "Spoken Dialogue Systems: A European Perspective," *Proc. Int'l Symp. Spoken Dialogue,* pp. 25-36, Oct. 1996.

[9] P. Price, "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. ARPA Human Language Technology Workshop,* pp. 91-95, Mar. 1990.

[10] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel, "Dialog in the RailTel Telephone-Based System," *Proc. Int'l Conf. Speech and Language Processing,* pp. 550-553, Oct. 1996.

[11] L. Lamel, S. Bennacef, J. Gauvain, H. Dartigues, and J. Temem, "User Evaluation of the MASK Kiosk," *Proc. Fifth Int'l Conf. Spoken Language Processing,* Nov. 1998.

[12] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, and N. Morgan, "The Berkeley Restaurant Project," *Proc. First Int'l Conf. Spoken Language Processing,* pp. 2139-2142, Sept. 1994.

[13] M. Blomberg, R. Carlson, K. Elenius, B. Granstrom, J. Gustafson, S. Hunnicutt, R. Lindell, and L. Neovius, "An Experimental Dialogue System: WAXHOLM," *Proc. Third European Conf. Speech Comm. and Technology,* Sept. 1993.

[14] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C Pao, J. Polifroni, R. Schloming, and P. Schmid, "From Interface to Content: Translingual Access and Delivery of On-Line Information," *Proc. Fifth European Conf. Speech Comm. and Technology,* pp. 2227-2230, Sept. 1997.

[15] H. Meng, S. Busayapongchai, J. Glass, D. Goddeau, L. Hetherington, E. Hurley, C. Pao, J. Polifroni, S. Seneff, and V. Zue, "WHEELS: Conversational System in Automobile Classifieds Domain," *Proc. Fourth Int'l Conf. Spoken Language Processing,* pp. 542-545, Oct. 1996.

[16] C. Wang, J. Glass, H. Meng, J. Polifroni, S. Seneff, and V. Zue, "YINHE: A Mandarin Chinese Version of the GALAXY System," *Proc. Fifth European Conf. Speech Comm. and Technology,* pp. 351-354, Sept. 1997.

[17] L. Lee, "Spoken Language Processing for Mandarin Chinese—-Present and Future," *Proc. 1998 Symp. Image, Speech, Signal Processing, and Robotics,* pp. 229-234, Sept. 1998.

[18] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics,* vol. 18, no. 1, pp. 61-86, 1992.

[19] Y.C. Lin, T.H. Chiang, H.M. Wang, C.M. Peng, and C.H. Chang, "The Design of a Multi-Domain Mandarin Chinese Spoken Dialogue System," *Proc. Fifth Int'l Conf. Spoken Language Processing* Nov. 1998.

[20] S. Seneff, H. Meng, and V. Zue, "Language Modeling for Recognition and Understanding Using Layered Bigrams," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 317-320, Mar. 1992.

[21] A. Lavie, "GLR*: A Robust Grammar Focused Parser for Spontaneously Spoken Language," PhD thesis, Carnegie Mellon Univ., 1995.

[22] W. Ward, "The CMU Air Travel Information Service: Understanding Spontaneous Speech" *Proc. DARPA Speech and Natural Language Workshop,* pp. 127-129, June 1990.

[23] M. Gavaldà and A. Waibel, "Growing Semantic Grammars," *Proc. 36th Ann. Meeting of the Assoc. Computational Linguistics,* pp. 451-456, Aug. 1998.

[24] E. Kaiser, M. Johnston, and P. Heeman, "PROFER: Predictive, Robust Finite-State Parsing for Spoken Language," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing,* Mar. 1999.

[25] S. Seneff, "The Use of Linguistic Hierarchies in Speech Understanding," *Proc. Int'l Conf. Spoken Language Processing,* Nov. 1998.

[26] R. Pieraccini, E. Levin, and C. Lee, "Stochastic Representation of Conceptual Structure in the ATIS Task," *Proc. DARPA Speech and Natural Language Workshop,* pp. 121-124, Feb. 1992.

[27] R. Schwartz, S. Miller, D. Stallard, and J. Makhoul, "Hidden Understanding Models for Statistical Sentence Understanding," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing,* pp. 1479-1482, Apr. 1997.

[28] W. Minker, "Stochastically-Based Natural Language Understanding Across Tasks and Languages," *Proc. European Conf. Speech Comm. and Technology,* pp. 1423-1426, Sept. 1997.

[29] M. Epstein, K. Papineni, S. Roukos, T. Ward, and S. Della Pietra, "Statistical Natural Language Understanding Using Hidden Clumpings," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing,* pp. 176-179, May 1996.

[30] R. Kuhn and R. Mori, "The Application of Semantic Classification Trees to Natural Language Understanding," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 5, pp. 449-460, May 1995.

[31] B. Carpenter and J. Chu-Carroll, "Natural Language Call Routing: A Robust, Self-Organizing Approach," *Proc. Fifth Int'l Conf. Spoken Language Processing,* Nov. 1998.

[32] Y.J. Yang and L.S. Lee, "A Syllable-based Chinese Spoken Dialogue System for Telephone Directory Services Primarily Trained with a Corpus," *Proc. Fifth Int'l Conf. Spoken Language Processing,* Nov. 1998.

[33] M. McCandless and J. Glass, "Empirical Acquistion of Word and Phrases Classes in the ATIS Domain," *Proc. Third European Conf Speech Comm. and Technology,* Sept. 1993.

[34] S. Kullback, "Information Theory and Statistics." New York: John Wiley & Sons, 1959.

[35] D. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the Scope of the ATIS Task: The ATIS-3 Corpus," *Proc. ARPA Spoken Language Technology Workshop,* pp. 3-8, Mar. 1994.

**Helen M. Meng** received the SB, SM, and PhD degrees in 1989, 1991, and 1995, respectively, all in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA. In 1995 she was a research scientist at the MIT Spoken Language Systems Group, where she worked on multilingual conversational systems. She is currently an assistant professor in the Department of Systems Engineering and Engineering Management at the Chinese University of Hong Kong, which she joined in 1998. She set up the Human-Computer Communications Laboratory there in 1999. Her research interest is in the area of human-computer interaction via multilingual spoken language systems, which integrate a plethora of speech and language technologies, including speech recognition, natural language understanding, discourse and dialog modeling, as well as, language generation, and speech synthesis. She is also working on translingual speech retrieval technologies. She is a member of the IEEE Signal Processing Society, the IEEE, and a member of Sigma Xi.

**Kai-Chung Siu** received the BS and MPhil degrees from the Department of Systems Engineering and Engineering Management from the Chinese University of Hong Kong, in 1998 and 2000, respectively. His reseach interests are in natural language understanding for spoken dialog systems.