Headword Percolation in a Multi-Parser Architecture for Natural Language Understanding

Po Chui Luk, Kui Xu and Helen Meng
Human-Computer Communications Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong, Hong Kong SAR
{pcluk,kxu,hmmeng}@se.cuhk.edu.hk

Abstract

This paper describes an approach for identifying the information goal(s) of information-seeking queries, such as those in the ATIS domain (Price, 1990). The approach is based the Multi-Parser Architecture (MPA) extended with a *headword percolation* procedure. MPA can generate full and robust parses for input natural language queries. Headword percolation identifies the prime constituent(s) in the parse trees. These constituents are incorporated in goal identification rules derived from training parse trees using an automatic, data-driven technique. The rules are then evaluated with a test set in terms of in single goal identification, multiple goal identification and out-of-domain rejection. We compare the current approach with a previous one based on Belief Networks (BN) (Meng et al. 1999). Results show that the former gave statistically significant improvements. This is indicative of the merits of rich semantic/syntactic analysis in the parse structures for goal identification.

1 Introduction

Natural language understanding (NLU) systems (Potaminaos and Kuo, 2000; Seneff, 1992) often incorporate parsing to generate a rich semantic/syntactic analysis of the input sentence. The analysis may be a full parse with a single tree, or robust parse with multiple sub-trees. Robust parsing is particularly useful for handling disfluencies in understanding spontaneous utterances transcribed by speech recognizers. The semantic interpretation of the input sentence may be derived from its parse. Semantic interpretation of information-seeking queries need to extract the *informational goal* and related semantic attributes for database access, e.g. by transformation into an SQL query. An example can be found in the ATIS domain (Price, 1990). This work focuses on the automatic identification of the informational goal of information-seeking queries. We extend a parsing framework known as the Multi-Parser Architecture (MPA) (Xu et al., 2001) with a *headword percolation* procedure. MPA is designed for efficient, modularized parsing with compact, reusable subgrammars. Headword percolation aims to locate the prime constituent(s) among the syntactic/semantic structures identified in the parse tree for an input sentence. These constituents are incorporated in informational goal identification rules derived from training parse trees using an automatic, data-driven technique.

Previous work in goal identification include the use of neural networks in (Wutiwiwatchai et al. 2003); support vector machines in (Zhang and Lee, 2003); belief networks (Meng et al. 1999); combined statistical pattern recognition and rule-based approach in (Wang et al. 2002); and the use of chunk parsing with statistical classifiers in (Li and Roth, 2003). The current approach focuses on the use of parsed head constituents in a small set (~100) of automatically derived rules.

2 The ATIS Corpus

This experimental study is based on the natural language queries in the Air Travel Information

Service (ATIS) corpus (Price, 1990). We use Class A queries that are self-contained. The training set has 1564 queries, the 1993 test set has 448 queries, and the 1994 test set has 444 queries. Each query has a corresponding SQL query for database access (see Table 1). The main attribute of the SQL is treated as the informational goal of the query and the remaining attributes as semantic categories.

Query: *show me the flights from baltimore to oakland*
Simplified SQL: select distinct *flight.flight_id* from *flight* where *from_airport.city_name* = "baltimore" and *to_airport.city_name*="oakland"

Table 1: An ATIS utterance with its corresponding SQL query.

3 The Multi-Parser Architecture (MPA)

The use of GLR(*k*) parsers for natural language understanding is hampered by the problem of exponential increase in parsing table size as the number of grammar rules grows. Large parsing tables are difficult to manage and lead to inefficiencies in parsing. Xu et al. (2001) proposed the MPA that supports efficient parsing through partitioning a (large) context-free grammar into multiple, small, reusable sub-grammars. Each sub-grammar has its own specialized sub-parser. Sub-parsers can be used to output sub-trees that are later composed into an overall parse for the input sentence. Hence, MPA is highly modularized and is conducive towards distributed computing as well as portability to new application domains (Meng et al. 2002).

MPA involves a *virtual terminal technique* to interface among sub-grammars and sub-parsers. A virtual terminal is essentially a non-terminal, but acts as if it were a terminal. The input set to a sub-grammar is a set of virtual terminals that were previously parsed by other sub-grammars. The output set of a sub-grammar is a non-terminal that is parsed with the current sub-grammar and may be used by other sub-grammars as their input sets. Hence a grammar partition (or sub-grammar) is a subset of production rules that can be viewed as a function – it takes the virtual terminals in the input set as input, and returns a non-terminal in the output set as output. Table 2 shows an example from the ATIS domain where the non-terminal LOC is the output of sub-grammar G_{LOC} and belongs to the input set of sub-grammar G_{DEPART}.

G_{DEPART}:	G_{LOC}:
OUTPUT: DEPARTURE INPUT: LOC, TIME_NP DEPARTURE →FROM LOC [TIME_NP]	…	OUTPUT: LOC INPUT: CITY_NAME, STATE_NAME… LOC→CITY_NAME [STATE_NAME]	…
FROM → from …	…

Table 2. Example sub-grammars (i.e. grammar partitions) G_{DEPART} and G_{LOC} in the ATIS domain.

MPA also involves a directed acyclic graph known as the *lattice with multiple granularity* (LMG) to record the input and output of all sub-parsers. The LMG has edges that may either be terminals or virtual terminals; and the nodes of the LMG indicate word positions in the parsed sentence. (see Figure 1). LMG forms a book-keeping representation in the *parser composition procedure* (Meng et al. 2002) that combines the output of sub-parsers. One method for parser composition is cascading. This is a bottom-up parsing procedure that first converts the input sentence into an LMG, invokes sub-parsers at each (virtual) terminal edge and migrates to the sentence level. Each sub-parser can start or end a parse tree at any edge and if it can parse for its output, a virtual terminal is added to dynamically to the LMG. Full/robust parses are generated in this way.

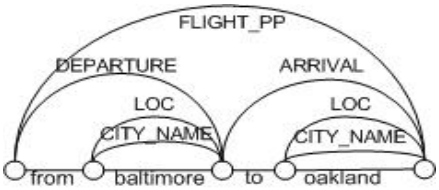

Figure 1: An example LMG in the ATIS domain.

3.1 Grammar Development

We wrote a set of context-free grammar rules for the ATIS domain that involves both semantic and syntactic categories. Low-level grammar rules are mainly semantic concepts, e.g. CITY-NAME, CLASS-TYPE, DAY-NUMBER, etc. High-level grammar rules mainly describe phrases, such as a time phrase, flight prepositional phrase (FLIGHT_PP), etc. SENTENCE-level grammar rules are generated using a data-driven approach – each training sentence is parsed into an LMG which is then trav-

ersed by the shortest path algorithm to generate a SENTENCE -level rule. (see Table 3).

S → ASK FLIGHT_NP | QUEST TIME_NP |…
ASK → show me | list | tell me| give me |…
FLIGHT_NP → FLIGHT FLIGHT_PP |…
FLIGHT_PP → DEPARTURE | ARRIVAL |…
ARRIVAL → TO LOC [TIME_NP] |…
CITY_NAME → atlanta | chicago | seattle |…

Table 3. Example grammar rules in ATIS.

We discarded SENTENCE-level rules that are lengthy or overly specific. Parse coverage for the training set, test set (93 and 94) are 97.1%, 62.5% and 66.0% respectively. We also partitioned the grammar into 67 sub-grammars and created virtual terminals such as S, FLIGHT_NP, AIRPORT_NAME, STATE_NAME, etc. Most of these correspond to semantic concepts or syntactic structures.

3.2 Semantic Interpretation

Upon completion of parser composition, the LMG may generate a full parse with a single tree (see Figure 2) or a robust parse with multiple sub-trees. In the latter case, we apply a heuristic scoring algorithm (see Equation 1) to find the "best" path linking sub-trees in the LMG. Higher level sub-trees are preferred over lower level ones.

$$s=(s_1+s_2+\ldots+s_k)\,\lambda^{n} \quad (1)$$

where

s	denotes the score of a sub-tree
$s_1,s_2,\ldots,s_k$	denote the score of the sub-tree's children if it contains k children (score of terminal is 1)
λ	denotes a constant factor (set at 1.1)
n	denotes the number of terminals covered by the subtree

The semantic interpreter walks through the full parse or best robust parse in top-down and left-to-right order. The interpreter extracts key-value pairs where keys are designed with reference to the schema of the SQL queries and values are the terminals corresponding to the keys (see Table 4).

Semantic Frame:
DEPARTURE_CITY_NAME=baltimore
ARRIVAL_CITY_NAME=oakland

Table 4: Semantic frame corresponding to parse tree in Figure 2.

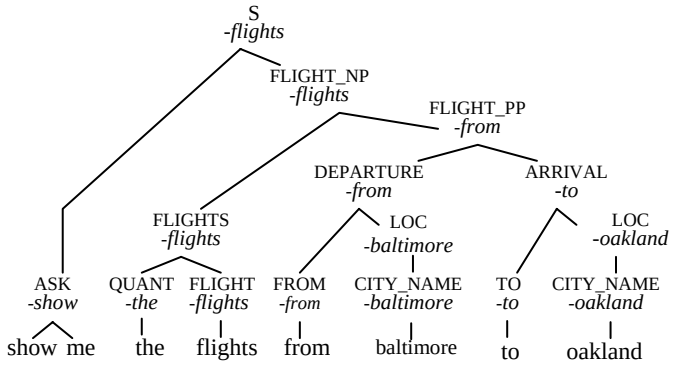

Figure 2: Parse tree of the sentence in Table 1. Italicized words are *headwords* (see Section 4).

4 Goal Identification with Head Constituents

4.1. Headword Percolation

A headword is defined to be the prime constituent of a phrase or sentence (Bloomfield, 1983). For example, the headword of a noun phrase is the noun; that of a verb phrase is the verb; a prepositional phrase is a preposition, etc. Remaining words in the phrase/sentence are regarded as the left or right modifiers of the headword. Headwords are percolated in parse trees through *head constituents* in context free rules. The rule $p \rightarrow l_n \ldots l_1\ h\ r_1 \ldots r_m$, has the non-terminal p with head child is h and l_i and r_i are the left and right modifiers of h respectively. We adapted the guidelines in (Collins 1999) in determining the head constituents for the ATIS grammar. Figure 2 shows headwords (in italics) that are percolated up the parse tree. An example in here is the non-terminal ARRIVAL (with rule ARRIVAL → TO LOC) that acts like a prepositional phrase. Its head is the first preposition residing in the non-terminal TO. The headword that reached the top is "*flights*". The *head label* is the non-terminal parent label of the headword, i.e. FLIGHT. We believe that the headword and head label are good indicators of the informational goal of the sentence. In this example, the informational goal is flight.flight_id (see Table 1), i.e. flight identification.

4.2. Goal Identification Rules

We have developed a set of goal identification rules that incorporate head constituents in the format of:

(head_label, headword, question_type) → *goal*

We define five question types by rule in ATIS – HOW_MUCH, HOW_MANY, HOW_LONG, WHERE and WHEN. Sentences that do not belong to any of these five are classified with type NULL. Hence goal identification rules can be derived from training sentences and their parses (see Table 5).

Query: *how many flights arrive at general Mitchell international* **Goal Identification Rule**: (FLIGHT, *flights*, HOW_MANY)→*count_flight* 5
Query: *what does fare code y mean* **Goal Identification Rule**: (with multiple goals) (BOOKING_CLASS, *y*, NULL) → *class_of_service.class_description*, *fare_basis.fare_basis_code* 3

Table 5: Examples of ATIS queries and their goal identification rules, each with a frequency of occurrence.

Deriving goal identification rules from fully parsed training sentences is straightforward. These full parses also enable us to automatically determine a set of 22 *prominent* virtual terminals (out of 67 in total) that contain the sentence head, e.g. S, FLIGHT_NP, FLIGHT, etc. In the case of robust parses in the training set, we refer to the "best" path, identified with Equation (1) above. This path links *multiple* parse trees which gives *multiple* headwords and head labels. To select a *single* subtree for goal identification rule derivation, we devised a *virtual terminal selection* procedure – subtrees are examined in order from left to right in the robust parse and the sub-tree whose root node contains a prominent virtual terminal will be selected. Its headword and head label will be incorporated into a goal identification rule. Next the algorithm searches for a question type by traversing the LMG from right to left. Within each node (or sub-tree) in the LMG, traversal proceeds in a top-down, left-to-right manner. A late-binding approach is used for selection, i.e. the last *question_type* discovered during the traversal is incorporated in the goal identification rule. Figure 3 illustrates a robust parse with three sub-trees. Virtual terminal selection chooses the second sub-tree since its root node contains virtual terminal S. The question type of this sentence is HOW MUCH.

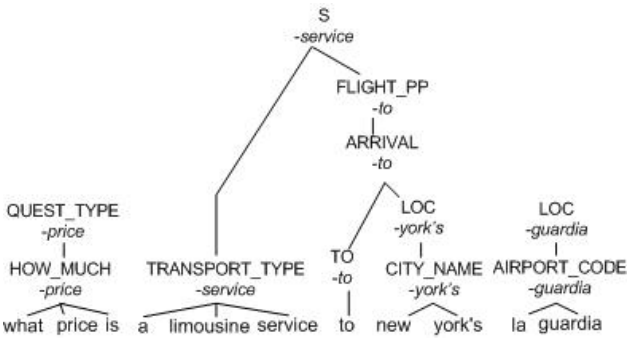

Figure 3: A robust parse with three sub-trees.

During testing, each sentence in the test set is parsed with MPA. If a full parse is obtained, the 3-tuple (*head label, headword, question type*) is extracted directly. If a robust parse is obtained, we invoke the virtual terminal selection procedure as well as search for the question type to extract the 3-tuple. The appropriate goal identification rule is then applied to map the 3-tupe to an informational goal. If no rule is applicable, the test sentence is classified as out-of-domain (OOD).

5 Experiments

The ATIS training set is parsed with MPA and from the parse trees we derive a set of 104 goal identification rules. In case of ambiguous rules whose 3-tuples are the same but map to different goal(s), we preserve the most frequently occurring rule. Only the 11 goals (out of 32 in all) that have been instantiated ten times or more in the training set are considered, e.g. *flight.flight_id, fare.fare_id*, etc. All remaining goals are rejected as out-of-domain.

We compared the goal identification performance of the headword percolation approach with an alternative approach based on belief networks (BN) (Meng et al., 1999). *N*(=11) BNs were developed, one for each informational goal. The input to the network is a set of semantic concepts parsed from the input query. Each belief network makes a binary decision regarding the presence or absence of its corresponding goal. Hence the

framework can also handle single goal queries, multiple goal queries (i.e. when multiple belief networks vote positive for their corresponding goals) and OOD queries (i.e. when all networks vote negative for their corresponding goals). However, the BN approach does not consider the rich semantic and syntactic analysis in the parse structures for goal identification. Results are shown in Table 6. The headword percolation approach outperforms the belief network approach on single goal identification and OOD rejection.

	1993 Test		1994 Test	
Approach	**BN**	**Head-word**	**BN**	**Head-word**
Single goal correct	381/405	388/405	373/401	388/401
# rejection	39	45	35	44
Rejection correct	23/35	27/35	13/37	31/37
Multiple goals correct	5/8	6/8	4/6	1/6
Correctly handled	409/448 (91.3%)	421/448 (94.0%)	390/444 (87.8%)	420/444 (94.6%)

Table 6: Comparing the goal identification performances between the BN approach and the headword percolation approach based on test sets 1993 and 1994.

We also compared goal identification performance between queries with full parses and queries with robust parses. Results are shown in Table 7. Goal identification from full parses is significantly better than from robust parses. This suggests that we should strive to improve parse coverage as well as seek better strategies in handling robust parses.

	Correct Goal Identification	
	1993 Test	**1994 Test**
Queries with full parses	275/280 (98.2%)	291/293 (99.3%)
Queries with robust parses	146/168 (86.9%)	129/151 (85.4%)

Table 7: Goal identification accuracies based on the headword percolation approach – comparison between full parses and robust parses.

6 Conclusions and Future Directions

This paper presents an approach for deducing the informational goal of a natural language query from its parse tree. A parsing framework known as the multi-parser architecture (MPA) has been designed to generate a full parse or robust parse from an input natural language query. The parse tree(s) present a detailed syntactic/semantic analysis of the input query. The MPA supports modular parsing, where a large grammar can be decomposed into multiple sub-grammars by a process known as *grammar partitioning*. Each sub-grammar has its own specialized sub-parser (GLR parser) that outputs a *virtual terminal* if it can parse an input sentence (fragment) successfully. Grammar partitioning offers the advantage of parsing efficiency through the use of multiple sub-grammars and sub-parsers with (small) parsing tables. This circumvents the problem of generating parsing tables that may become unmanageably large as the grammar grows in size. The second process in the MPA is *parser composition*, which combines the outputs of the sub-parsers to generate an overall parse for the input sentence.

This work enriches the MPA with a *headword percolation* procedure. The headword is deemed the prime constituent of a phrase or sentence. Various headwords are selected and percolated up the parse tree according to the guidelines outlined in (Collins 1999) and adapted for our domain-specific experimental corpus, i.e. the ATIS corpus. Headword percolation aims to locate the most important constituent among the syntactic/semantic structures identified in the full parse of the input sentence. In the case of a robust parse where a sentence is analyzed in terms of multiple sub-trees, a *virtual terminal selection* procedure is designed to select the key sub-tree for deriving the headword. We believe that such head constituents are most indicative of the informational goal of the query. Hence, we parse all training sentences and extract the headwords, head labels (i.e. non-terminals of the head words) and question types (also a non-terminal) from the parse trees. These are used to generate a set of goal identification rules of the format:

(head label, headword, question type) → *goal*

Each goal identification rule is associated with a frequency of occurrence in the training set.

We performed goal identification experiments on the ATIS test sets. Each testing sentence is parsed and the 3-tuple *(head label, headword, question type)* is extracted from the parse tree. Goal identification rules derived from the training data are applied to map the 3-tuple into one or more goals. Should none of the rules be applicable, the testing sentence is rejected as out-of-domain (OOD). The headword percolation approach correctly handles over 94% of the ATIS test sets in terms of single/multiple goal identification as well as OOD rejection. This is a statistically significant improvement over an alternative, belief network-based approach that lacks a rich semantic/syntactic analysis such as that presented in a parse tree. Furthermore, while the generation of goal identification rules is a data-driven process, the resulting performance does not seem to be affected severely by sparseness in training data. Additionally, analysis of results shows that goal identification performance of queries with full parses is significantly better than queries with robust parses. This suggests the possible future directions of improving the ATIS grammar to achieve better parse coverage, or devising better methods for handling robust parses in goal identification. We will also explore the simultaneous identification of informational goals (i.e. task goals) and dialog acts using the head percolation approach.

7 Acknowledgments

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong SAR (Project No. CUHK4326/02E). We thank Fuliang Weng for his contributions in this work.

References

L. Bloomfield. 1983. *An Introduction to the Study of Language*, Henry Holt and Company, New York.

E. Charniak. 2001. "Immediate-Head Parsing for Language Models," Proceedings of ACL.

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing, PhD thesis.

X. Li and D. Roth, 2002. "Learning Question Classifiers," Proceedings of COLING.

H. Meng, W. Lam and C. Wai. 1999. "To Believe is to Understand," Proceedings of Eurospeech.

H. Meng, P.C. Luk, and K. Xu. 2002. "GLR Parsing with Multiple Grammars for Natural Language Queries," ACM Transactions on Asian Language Information Processing, Vol. 1, No. 2, Pages 21-42.

A. Potamianos and H.J. Kuo. 2000. "Statistical Recursive State Machine Parsing for Speech Understanding," Proceedings of ICSLP.

P. Price. 1990. "Evaluation of spoken Language Systems: The ATIS domain," Proceedings of the ARPA Human Language Technology Workshop.

S. Seneff. 1992. "TINA: A Natural Language System for Spoken Language Applications," Computational Linguistics, Vol. 18, No. 1.

Y.Y. Wang et al. 2002. "Combination of Statistical and Rule-based Approaches for spoken Language Understanding," Proceedings of ICSLP.

C. Wutiwiwatchai and S. Furui. 2003. "Combination of Finite State Automata and Neural Network for Spoken Language Understanding," Proceedings of Eurospeech.

K. Xu, F. Weng, P. C. Luk and H. Meng. 2001. "A Multi-Parser Architecture for Processing Natural Language Queries," Proceedings of Eurospeech.

D. Zhang and W.S. Lee 2003. "Question Classification using Support Vector Machines," Proceedings of the SIGIR Conference.