2005

An investigation of the electrolytic plasma oxidation process for corrosion protection of pure magnesium and magnesium alloy AM50.

Yueyu Ma
University of Windsor

An Investigation of Grammar Design in Natural-Language Speech-Recognition

By

Shi, Yue

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the School of Computer Science
In Partial Fulfillment of the Requirements for
The Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2003

Your file Votre référence
ISBN: 0-612-92514-5
Our file Notre référence
ISBN: 0-612-92514-5

Abstract

With the growing interest and demand for human-machine interaction, much work concerning speech-recognition has been carried out over the past three decades. Although a variety of approaches have been proposed to address speech-recognition issues, such as stochastic (statistical) techniques, grammar-based techniques, techniques integrated with linguistic features, and other approaches, recognition accuracy and robustness remain among the major problems that need to be addressed.

At the state of the art, most commercial speech products are constructed using grammar-based speech-recognition technology. In this thesis, we investigate a number of features involved in grammar design in natural-language speech-recognition technology. We hypothesize that: with the same domain, a semantic grammar, which directly encodes some semantic constraints into the recognition grammar, achieves better accuracy, but less robustness; a syntactic grammar defines a language with a larger size, thereby it has better robustness, but less accuracy; a word-sequence grammar, which includes neither semantics nor syntax, defines the largest language, therefore, is the most robust, but has very poor recognition accuracy. In this Master's thesis, we claim that proper grammar design can achieve the appropriate compromise between recognition accuracy and robustness.

The thesis has been proven by experiments using the IBM Voice-Server SDK, which consists of a VoiceXML browser, IBM ViaVoice Speech Recognition and Text-To-Speech (TTS) engines, sample applications, and other tools for developing and testing VoiceXML applications. The experimental grammars are written in the Java Speech Grammar Format (JSGF), and the testing applications are written in VoiceXML. The tentative experimental results suggest that grammar design is a good area for further study.

Acknowledgements:

The work described in this report has been carried out at the School of Computer Science, University of Windsor, Canada. This thesis was completed under the guidance of the following committee:

Dr. Kai Hildebrandt – External Reader

Dr. Joan Morrissey – Internal Reader

Dr. Richard A. Frost – Advisor

Dr. Scott Goodwin - Chair

The author expresses the gratitude to her committee for their valuable support, guidance, and advice.

Also, the author would like to express her special sincere thankfulness to Dr. Richard A. Frost, who introduced her to this specific topic in speech-recognition, encouraged her from time to time, given her undivided attention, and various guidance that has gone beyond his supervisory role. Without his consideration and assistance, this thesis would still remain a challenge.

Also, the author would like to thank her husband, her parents, sisters, and brothers, the whole family, for their enduring love, encouragement, and support, along the long way, without which the author would be too weak to do anything.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

While speech recognition has been an active field for several decades, some newly-developing areas, e.g. computer-telephony integration, are demanding the speech solutions. In addition, the explosive growth in the use of wireless devices and the World Wide Web has created an urgency for better tools to manipulate speech-related operations, such as voice data entry and speech navigation of the web.

Although some new products have emerged recently, such as voice portal (McTear, 2002) (which provides a speech-based interface between a telephone user and web-based services), and VoiceXML (which is an XML-based markup language for creating distributed voice applications, much as HTML is a markup language for creating distributed visual applications (IBM, 2001)), the core is the speech-recognition technology, which still has a long way to go before the real value of the new tools can be harnessed.

Over the last three decades, a number of Artificial Intelligence (AI) researchers have been striving to build models to interact between humans and machines with natural-language speech. However, it is only in the past decade that speech technology has achieved advanced progress with the introduction of both research prototypes and commercial applications, such as SPHINX (the first accurate large-vocabulary continuous speaker-independent speech-recognition system developed at Carnegie Mellon University (Huang *et al.*, 1992) (Lee, 1988) (Kita and Ward, 1991)), ATIS (an actual spoken-language Air Travel Information System (Moore *et al.*, 1995)) , and the JUPITER weather-information system (developed at MIT, (Glass, 1999)).

Although speech-recognition technology has been addressed from various perspectives, a number of problems need to be solved, such as recognition accuracy, robustness, and flexibility. Speech is recognized correctly if and only if the recognition result returned from the system is correctly corresponding to the user's speech input. Robustness means the extent to which a system handles errors or "unexpected" input. A flexible spoken-dialogue system is able to accept a user's flexible utterances, allow the user to supply extra information and make reasonable responses (Milward, 1999).

In this thesis, we investigate the significance of grammar design in speech recognition from various aspects. This thesis is supported by an experiment with multi-direction comparisons over three types of grammar (semantic grammar, syntactic grammar, and word-sequence grammar, which are discussed in detail in chapter 5). We observe that the size of a language defined by a grammar has a significant influence on speech-recognition accuracy (and robustness, which is expected). The smaller language, which can be obtained by including semantic constraints in the syntax, has better accuracy and less robustness, with more complicated grammar design. (We explain what a "semantic constraint" is in sub-section 1.4). The syntactic grammar, using a less-complicated grammar, defines a larger language to obtain better robustness, but less accuracy. The trade-off between accuracy and robustness is adjusted by the grammar design. Therefore, grammar design is an extremely important topic in natural-language speech-recognition.

Although the preliminary experiments show no contradictive evidence to our claim, we have encountered some limitations that are discussed in sub-section 8.2.

1.1 The Need for Speech-Recognition

Looking back on human history, language has marked the evolution of humankind: words recorded the civilization of human society, and speech has been the most common, convenient, and preferred method of communication for human beings. For the majority of human beings, speech communication is the easiest way to convey information from human to human, for it can make hands free, can proceed in the dark, and can even reach very far distances through radio and telephone.

The question is: can machines make use of all of the advantages of human's natural-language speech? If a machine can understand natural language, one can easily interact with that machine (just like humans communicate with humans) in natural language to retrieve information, conduct transactions, or perform other problem-solving tasks. For example, people can direct the machine in spoken language to execute commands; with the assistance of external equipment (e.g. a telephone), activate remote controls or fulfill remote commercial transactions; visit the speech web with natural spoken-language input and voice output without text or graphic interfaces. Virtual-reality technology can be strengthened with more realistic natural-speech interactions. Machines can dictate what one says and save it as a text document; machines can automatically translate one language into other languages. People with vision disability will suffer less on account of the help of machines equipped with natural-language ability.

In addition, the World Wide Web has become an important tool in modern people's daily life to retrieve information and conduct e-business transactions. But the current popular structure is mostly based on visual interfaces, which means that information and services are delivered to users in graphical and textual formats via computers. Consequently, the web ignores a large number of people who have visual disabilities or do not have access to a computer due to time, location, and/or cost constraints. Therefore, we are looking forward to an alternative way to interact with the web, which provides such people with the chance to access the information and services by voice, i.e. a speech web.

1.2 Spoken-Dialogue Systems

A complete spoken-dialogue system involves integration of the following components (McTear, 2002) (Han, 2000) (Glass, 1999): a speech-recognition component, a language-understanding component, a dialogue-management component, a component for communication with an external system, a response-generation component, and a speech-output component. These components work in a sequential stream, in which the first component receives the user's input, then the output from that component feeds into the next component as the input, and so forth, until the consequent voice output is synthesized for the user. An overview of the interaction of the components in a spoken dialogue system is as follows (McTear, 2002):

The speech-recognition component receives the user's input utterance and converts the continuous-time signal into a sequence of discrete units for the use of the language-understanding component. As the language component receives the information from the previous speech-recognition component, it analyzes the discrete units and derives a meaning representation for the next dialogue control component. The dialogue-management component controls the dialogue flow by determining whether the user has provided sufficient information, also communicating with the external application and the user. Usually, it is a database that acts as the external system component for the requested information retrieval in the spoken-dialogue system. Finally, the response-generation component will construct the message retrieved from the external system component corresponding to the user's request and send it to the speech output component to synthesize the voice output for the user.

1.3 Voice Applications

Voice applications are applications in which the input and/or output are through a spoken, rather than a graphical, user interface (IBM, 2001). The voice application can be a stand-alone application, whose files reside on the local machine, or a distributed application, with application files residing in an intranet, or on the Internet.

Typically, voice applications can be categorized into "queries" and "transactions" (IBM, 2001). The purpose of user access to a "query application" system is to retrieve information. The system provides users with a series of instructions, such as prompts and menu choices, the user uses spoken commands to make menu selections and fill in form fields. Based on the user's input, the system locates the appropriate information from a back-end database, and presents the desired information to the user in voice output.

The "transaction" voice-application system provides users with the opportunity to execute specific transactions using voice. The user is guided to provide the data required for the transaction, and then responds to the system using spoken commands. Based on the collected data from the user's input, the system executes the transaction and updates the appropriate records in the corresponding back-end database. Also, the system reports back to the user by playing back prerecorded audio files or by synthesizing speech based on the information in the database records.

1.4 The Specific Problems to Be Addressed

Among the problems existing in speech recognition, accuracy and robustness are two important problems to be addressed. Although human beings seldom make mistakes in recognizing commonly-used spoken words in their own language, machines are susceptible to recognition ambiguities or errors owing to a noisy environment, speech disfluency, and inability to use contextual knowledge. Since it is impractical to expect the machine to recognize speech 100% correctly, to improve the recognition accuracy becomes one of the major goals.

Then, if the system cannot recognize the user's speech input, will it be stuck? Humans have the ability to tolerate the mistakes in human-human communication to some extent. For example, if a person asks "which moon *did* discovered by Hall?", we - human beings - can make the reasonable guess that he/she is asking "which moon *was* discovered by Hall", and give him/her a corresponding response. Therefore, we expect a spoken-language system to be robust to handle the user's errors or "unexpected" input to some extent, so that the system can provide a reasonable response to the user, and the human-machine interaction can proceed smoothly.

A grammar defines a language by specifying the legal utterances, i.e., the sequences of words that the user may say (Lucas, *et al.*, 1999)(VXML, 2000). Even with the same domain, different grammars can define different kinds of languages. For example, if some semantic constraints are encoded into the syntax, the semantic grammar defines a smaller language than the corresponding syntactic grammar. For example, a sentence can be defined as a noun phrase followed by a verbphrase, denoted as the following syntax:

<sentence> = <nounphrase> <verbphrase>

By this syntactic grammar, the sentences "a tree runs" and "a boy loses leaves" are legal, though they are not accepted in common sense. To avoid such funny sentences, some semantic constraints can be encoded into the syntax to further keep the correct semantics, as well as the correct syntax. The corresponding semantic grammar is as follows:

<sentence> = <animatenounphrase> <animateverbphrase>
| <inanimatenounphrase> <inanimateverbphrase>

Then, the semantic grammar requires that an animate noun phrase (e.g., a boy) should be followed by an animate verb phrase (e.g, runs), and an inanimate noun phrase (e.g., a tree) should be followed by an inanimate verb phrase (e.g., loses leaves). So, the sentences "a tree runs" and "a boy loses leaves" are not correct in semantic grammar, though they are correct in syntactic grammar. The accuracy is improved with the reduction of the defining language, but the robustness is lowered meanwhile. How to balance the accuracy and robustness is a great challenge for speech-recognition researchers.

1.5 Thesis Statement

This thesis is concerned with grammar design in natural-language speech-recognition. Several features are examined through initial experiments. In particular, we claim that:

(1) Encoding semantic constraints in a grammar can improve speech-recognition accuracy;

(2) Using a combination of grammars with different weights (probabilities) can help achieve good accuracy and good robustness.

1.6 The Structure of This Thesis Report

The rest of this thesis report is constructed as follows:

A review of some speech-recognition techniques, such as statistical techniques, grammar-based techniques, and techniques involving semantics, is presented in chapter two and chapter three; chapter four discusses the existing problems in the state-of-the-art speech-recognition technology; chapter five discusses the grammars used in the experiment; chapter six proposes the investigation of the grammar design from various aspects; the experiments, results, and analysis are described in chapter seven; finally, conclusions and future work are summarized in chapter eight.

Chapter 2

Overview of Speech-Recognition Techniques

Currently, many ways to construct language models for speech recognition exist. Roughly, the approaches can be categorized into stochastic (statistical) models (which require a large corpus of training data) and grammar-based models (which use grammars to specify the utterances) (Rayner *et al.*, 2000b). A language model consists of a vocabulary (a set of words that can be recognized by the system) and grammar (a set of rules by which sentences are parsed or constructed) (Souto *et al.*, 2002). The grammar can be a set of linguistic rules or a stochastic (statistical) model. Generally, if a substantial domain corpus is available, a stochastic (statistical) language model is better as it is more robust; otherwise, a Context-Free Grammar-based language model may be more appropriate.

Stochastic (statistical) techniques and grammar-based techniques are two main streams in language-model constructions. It was reported in (Knight *et al.*, 2001) that stochastic (statistical) language models were popular around 1995, while by 2001, grammar-based language models took the pre-eminent position in commercial products.

In this thesis report, we give only a brief overview of speech-recognition techniques; more details can be found in Appendix A, which contains a comprehensive survey of research and the use of natural-language features to improve speech-recognition accuracy.

2.1 Stochastic (Statistical) Techniques in Speech-Recognition

A Statistical Language Model (SLM) is simply a probability distribution P(*s*) over all possible sentences *s*, or spoken utterances, documents, or any other linguistic units (Rosenfeld, 2000a).

The typical architecture of the speech language-understanding system that uses a stochastic model is described in (Knight *et al.*, 2001) as follows: firstly, a domain corpus is collected and used to create the statistical language model; then the statistical language model is incorporated into the recognizer; after that, a robust phrase-spotting parser is built to analyze the text output of the recognizer and produce semantic representations in the form of slot/filler pairs.

Statistical Language Models (SLMs) have the advantages of simplicity, flexibility, adaptation, higher recognition accuracy, and robust performance. Meanwhile, SLMs suffer from the unavoidable disadvantage of the costly collection of huge amounts of training data. In addition, SLMs are not supported by commercial systems, such as VoiceXML browsers.

2.1.1 N-grams

The N-gram is the most frequently-used SLM technique in speech recognition. N-gram means: with enough amount of training data, each word can be predicted from the previous N-1 words (Souto *et al.*, 2002). The probability of a word's occurrence can be predicted by the preceding N-1 words, and one or more candidate words are output in some ranked "recognition-hypothesis list".

The type of training data to be collected is determined by the task of the model. For example, if it is a model for a medical application, the training data should be focused on medical reports, papers and other resource instead of sports or fashion. Usually, a trigram (N=3) is used with large training corpora (millions of words), whereas a bigram (N=2) is used with a smaller set of training data to create a less-accurate model (Rosenfeld, 2000a).

The primary advantage of the N-gram lies in its robustness.

2.1.2 Multi-class Composite N-gram (Class N-gram)

The sparseness (the infrequency of word sequences in a corpus (Magerman and Marcus, 1990)) is a common problem in the N-gram approach, even with large corpora. For example, in some training corpora, many triplets (in trigram) appear only once or a few times, thus, the straightforward estimation of N-gram probabilities from counts is not viable.

To address the problem of data sparseness, Rosenfeld (2000a) described an effective "class N-gram" technique, which is also proposed by Yamamoto *et al.* (2001), by using vocabulary clustering to battle the sparseness problem. Multiple words are assigned to one word class representing either syntactic categories (e.g., noun or verb) or semantic categories (e.g., days of the week, names or airports) (McTear, 2002) (Baggia *et al.*, 1999), thus, the transition probabilities from word to word are approximately changed to that from class to class. Consequently, with the decreased search space (the number of classes is much smaller than that of the original words), the perplexity is reduced and recognition accuracy increases.

The key point of this technique lies in the clustering, which determines the quality of the model. It works better with small domains by manual clustering of semantic categories, and it is not as effective in less-constrained domains (Rosenfeld, 2000a).

2.1.3 Decision-Tree Models and Semantic-Classification-tree Models

Decision-tree models (Rosenfeld, 2000a) as well as semantic classification-tree models (Noth *et al.*, 1996) take the advantage of a decision-tree structure. "A decision tree can arbitrarily partition the space of histories by asking arbitrary binary questions about the history at each of the internal nodes" (Rosenfeld, 2000a). The probability distribution of the next word is constructed, based on the training data at each leaf. Interpolating the leaf distribution with the internal-node distribution found along the path can contribute to reduce the variance of the estimate (Rosenfeld, 2000a).

This kind of model suffers from the huge search space. If the average vocabulary size is denoted as b (the branching factor of the tree); and the utterance length is denoted as d, (the depth of the tree), the decision tree model has space complexity of $O(b^d)$. Therefore, special techniques to prune the large trees are required.

2.1.4 Adaptive Models

Adaptive models in (Rosenfeld, 2000a) are addressed to alleviate the domain-restriction problem (discussed in sub-section 4.5). The Cross-Domain Adaptation model takes advantage of a cache to transfer test data to the language model without training. In the Within-Domain Adaptation model, the test data comes from the same source, but this particular source consists of many subsets of various topics, styles, or both. Then the adaptation can proceed among the subsets, and two different domains can be combined to construct a general model so that the language model can cover a wider domain.

A potential problem with adaptive models is that an increase in training data does not guarantee a corresponding improvement in the accuracy of the language model due to the fact that the data increases that occur in some domains might have little influence on the model in other domains.

2.1.5 N-best Filtering or Rescoring

N-best filtering or rescoring is a very simple search technique (Moore, 1999). Just as its name implies, this technique always chooses the best one in the sorted recognition hypothesis list according to certain criteria.

While simplicity represents the primary advantage of N-best filtering or rescoring approach, the high computational cost for large N is its disadvantage.

2.1.6 Learning Techniques

One of the big problems associated with SLMs is how to obtain the huge corpus of training data. Bootstrapping (Rayner *et al.*, 2000a)(McCandless and Glass, 1994)(Baggia *et al.*, 1999) and use of the World Wide Web (Zhu and Rosenfeld, 2000a) are two of the popular techniques to obtain the training data. Bootstrapping is the simplest and cheapest way to collect training data. Its basic mechanism is to build an initial version of the system using a hand-coded model, then put it into practice to collect more data. Recursively, the data is used to construct a new language model and that is used to collect new data. This cycle can be repeated until satisfactory accuracy is achieved. Also, the explosion of the information online makes the World Wide Web a good source for collecting training data.

2.2 Grammar-Based Speech-Recognition

As an alternative to Statistical Language Models (SLMs), which apply word probabilities (N-gram) as the only form of language knowledge (Rosenfeld, 2000a), grammar-based speech recognition describes the language features in a set of rules to generalize over a certain application domain.

According to Knight *et al.* (2001), the up-to-date grammar-based strategy (which is usually adopted by commercial organizations) is like this: use Nuance or Speechworks as a standard commercial platform; then hand-code a grammar in some subset of Context-Free Grammar (CFG), and extend the grammar with semantic annotations; later on, using a system-initiative dialogue strategy, code in Nuance's Speech Objects or Speechworks' Dialogue Models or VoiceXML.

Compared to statistical techniques, grammar-based speech recognition is more common and easier to use and has reasonable recognition accuracy for small domains. In addition, an important advantage over statistical approaches is that grammar-based approaches do not require a large amount of training data that is difficult and expensive to collect.

However, grammar-based techniques require experts to write high-quality grammars, and the grammar rules are difficult to maintain and extend. In addition, grammar-based recognition is not as robust as are statistical techniques. For example, it cannot handle the utterances that are not covered by the grammar.

2.2.1 CFGs

A Context-Free Grammar (CFG) is a crude, yet well-understood, model of natural language. A CFG consists of a vocabulary, a set of non-terminal symbols, and a set of production or transition rules. Usually, a CFG can be defined as a set of rules that have a single atomic grammatical category on the left-hand side, and a sequence of atomic categories and words on the right-hand side (Moore, 1999)(Amaya *et al.,* 1999). Based on the fact that all context-free rules can contain only one symbol on the left-hand side, and it is free to be replaced by the right-side rules, comes the name "Context-Free Grammar" (Blackburn and Striegnitz, 2002).

A sample CFG grammar that defines a sentence, such as "a boy opened the door", is shown in figure 2.2.1:

<S> = <NP> <VP>;
<NP> = <Det> <N>;
<VP> = <V> <NP>;
<Det> = the | a;
<N> = boy | door;
<V> = opened | closed;

Figure 2.2.1: a sample CFG grammar

2.1.2 Statistical or Probabilistic Grammars

Probabilistic Context-Free Grammars (PCFG) and Probabilistic Dependency Grammars (PDG) are two probabilistic (statistical) grammars. PCFGs are CFGs with a probability distribution defined over all productions that share their left-hand side (Rosenfeld, 2000b) (Moore, 1999) (Weber and Gőrz, 1999). For the example, the conditional probability of the rule S -> NP VP might be 0.5, that means: if there is a sentence S, there is 0.5 chance that it consists of a NP (noun phrase) followed by VP (verb phrase).

PDGs have some similarity to regular N-grams in that each word is predicted based on a number of other words. The difference is that, in a conventional N-gram, each word is predicated from the N-1 words immediately before it; whereas in a PDG, the words acting as the predictors depend on a hidden variable, the dependency graph (Rosenfeld, 2000a). Typically, a sentence s is parsed to generate the most likely dependency graphs Gi (with attendant probabilities P(Gi)); then compute each generation probability P(s|Gi) (either N-gram style or an Maximum Entropy (ME) model); finally, the complete sentence probability is given by $P(s) \approx \Sigma_i P(Gi)*P(s|Gi)$ (the reason for the approximation is that the P(Gi) themselves were derived from the sentence s). Sometimes P(s) is further approximated as P(s|G*), where G* is the single best scoring parse (Rosenfeld, 2000a).

2.2.3 Discourse Grammar

The idea of Discourse Grammar that was proposed by Churcher *et al.* (1996) is to break the large syntax into smaller syntaxes to improve the performance of the language models with lower perplexity and ambiguity. The supporting idea is that, generally, the smaller syntax contains fewer words and less complicated structure than the original one, hence is potentially less ambiguous. A discourse segment can be a set of utterances with some properties in common, e.g., a certain topic, or even the discourse between a set of speakers, i.e., a dialogue.

2.2.4 Semantic Grammars

According to Demetriou and Atwell (1994a), semantic grammars are usually represented as transition networks, and provide stronger constraints than pure syntax by integrating semantic conditions closely with the syntactic rules of the grammar. A syntactic grammar is effective in describing the structure of phrases and sentences, whereas semantic constraints are particularly useful for languages whose phrase orders are not very constrained, such as Japanese (Takezawa *et al.*, 1991).

2.3 Combined Stochastic (Statistical) and Grammar-Based Techniques

As we have seen, both stochastic (statistical) and grammar-based techniques have their advantages and disadvantages. A question is whether it is feasible to take their respective advantages and overcome the disadvantages by integrating the stochastic techniques and grammar-based techniques.

There are some successful cases that combine these two techniques. The ATIS, Air Travel Information System (Moore, *et al.* 1995) uses a CFG in parsing and produces a sequence of grammatical fragments, then, applies a trigram (N=3) to obtain a 15% reduction in a speech-recognition-error rate. Knight *et al.* (2001) first set up a CFG grammar-based system, then used it to collect the training corpus for a SLM. The results show the effectiveness of grammar-based language for in-coverage sentences, and the SLM for out-of-coverage examples. Also, Rayner and Carter (1997), Geutner (1996), and Jones *et al.* (1993) achieved robust and efficient performance within a linguistically motivated framework by combining the rule-based and statistical methods.

More detailed discussion of the research described in this chapter is given in Appendix A.

Chapter 3

Overview of Existing Techniques of Using Semantics in Speech-Recognition

Language features are very effective in any system for reducing the number of possible utterances and for prioritizing utterance hypotheses (Hermannsdottir, 1996). Takezawa *et al.* (1991) say that "the accuracy of speech recognition heavily depends on what kinds of linguistic knowledge are used". At the current state of the art, to achieve high accuracy in speech recognition with moderate to large vocabularies (hundreds to tens of thousands of words), language models are necessary (Moore, 1999)(Harper *et al.*, 2000)(Takezawa *et al.*, 1991)(Seneff *et al.*, 1995) as discussed earlier, and in Appendix A.

Semantics is that part of linguistic knowledge which is concerned with meaning. Semantic rules can be used to restrict the expressions of a language defined by a grammar. For example, the question "which man orbits a blue man" is syntactically correct but not semantically correct.

3.1 Use of Large N, N-grams to Try and Capture Semantic Information

A traditional N-gram predicts the current word by the immediately previous N-1 words (discussed in sub-section 2.1), which assumes that the relevant information lies in the immediate past. However, the fact is that some syntactic or semantic information does exist farther back in the utterance. On the other hand, if a larger N in an N-gram model is used, the free parameters will increase exponentially, which is too difficult to analyze.

Supported by an experiment using long-distance bigrams with reduced number of free parameters, Huang *et al.* (1992) concludes that there is some relevant information, which is thinly spread across the history, in the distant past.

Considering the fact that in many languages (e.g. English) multiple words can be unified together and be treated as a single unit (phrase) in communication, Riccardi and Bangalore (1996) and Riccardi and Gorin (1998) proposed "phrase-based language models" to better (compared to word-based language models) capture long-spanning dependencies between words, without the exponential increase in the number of parameters.

3.2 Semantic Post-Processing of Output from Statistical Recognizer

Since the goal of completely eradicating speech-recognition errors at the front-end of the recognizer is impractical at the state of the art, many approaches using semantic post-processing for error correction have been investigated to further improve the recognition accuracy.

3.2.1 Post-processing to Choose Best Hypothesis

On account of its simplicity and efficiency, N-best search can be used in a post-processing stage in speech recognition to get better performance. Tran *et al.*, (1996) first constructed a recognition-hypothesis word graph, and then extracted the N-best word sequences from the word graph. Combined with language features, such as syntactic and/or semantic analysis, the N candidates can be re-scored with highly-reduced computational cost (Rayner *et al.*, 1994), and even many of the top-N sentence hypotheses can be eliminated before reaching the end with this type of syntactic and semantic analyses (Seneff *et al.*, 1995).

3.2.2 Post-processing to Correct Errors

Loken-kim (1988) developed the Automatic Error Detection and Correction System (AutoDac), which is able to parse ill-formed sentences with a combination of left-to-right and right-to-left parsing; learn the history of recognition errors and utilize this information to subsequently recover from similar recognition errors in future tasks; and allow a user to manually correct any part of the recognized sentence. Combining automatic and manual error correction, a total of 142 out of 192 testing sentences were recovered correctly (Loken-kim, 1988).

3.2.3 Post-processing to Modify System for Future Use

In the voice-interactive natural language system, Fink (1984) added a special module, called an expectation system, to aid the speech-recognition process. The basic idea is that the expectation system accepts the user's utterances and studies repetition and patterns in the dialogues to create a more general dialogue, then uses this generalized dialogue to correct errors in future sentences by prediction. The results showed that the average sentence error rate was decreased from 53% to less than 8%. Furthermore, it was concluded that the expectation system is capable of predicting what might happen in any situation that tends to be repeated.

3.3 Integrating Semantics into the Grammar to Better Direct the Recognizer – Unification Grammars

Belonging to the augmented or annotated Context-Free Grammars, a Unification Grammar is more expressive and more concise than a traditional CFG in "representing" semantics in a syntactic notation. A Unification Grammar is a higher-level formalism than a Context-Free Grammar, and is obtained by applying some restriction properties to a CFG. With constraints unified to the grammar, Unification Grammars help reduce the system's perplexity. To better understand the Unification Grammar, consider the following example from (Moore, 1999):

S: [tensed=yes] → NP: [person=P, num= N] VP: [tensed=yes, person=P, num=N]

The difference to a traditional Context-Free Grammar (CFG) is the notion of the feature constraints (e.g. person=P, num=N). The consequent power lies in the fact that the Unification Grammar constrains the features to variable matching instead of listing all matching constraint value pairs. The subsequent advantage can be seen from the above example that Unification Grammar guarantees that the person and num features of Noun Phrase (NP) and Verb Phrase (VP) must agree with each other, avoiding enumerating their respective features (person = first, num = singular, and so on).

3.4 Integrating Semantics into the Grammar to Better Direct the Recognizer - Direct Encoding of Semantics as Syntax Rules

Observing that some syntactically correct utterances may be semantically wrong, Frost (2002) proposed an approach for encoding semantic rules directly in the syntax of the grammar to reduce the size of the language and therefore improve the recognition accuracy. Frost (2002) presented an example in which the sentence "which man orbits kuiper" may be accepted by a simple grammar for its correct syntax, but in the domain used in the example, people cannot orbit other people, thus it is semantically incorrect. The simple syntax that accepts the above example sentence might be as follows:

question ::= "which" nounphrase verbphrase

If we replace it with the following:

question ::= "which" animatenounphrase animateverbphrase
| "which" inanimatenounphrase inanimateverbphrase

then the semantically incorrect utterance above is not accepted, the perplexity is reduced, and hence the speech-recognition accuracy should be improved.

The primary advantage of this technique is an improvement in speech recognition accuracy without unnaturally restricting the input utterances. However, this technique has the disadvantage that there is an increase in the size of the grammar by encoding semantic rules in the syntax, and this makes the system difficult to maintain. This can be overcome to some extent by combining this technique with the use of hyperlinks to create a Speech Web of speech-accessible objects, and further improve recognition accuracy by allowing the user to move between domain-dependent grammars (Frost, 2002).

The method investigated in this thesis is based on Frost's idea of encoding semantic constraints in the syntax of the recognition grammars.

3.5 Speech Webs

It is not easy to construct speech interfaces to large knowledge bases for the reason that large knowledge sources require large and complicated grammars, which are not trivial to implement and which have high perplexity and therefore low accuracy (Frost and Chitte, 1999). Instead, Frost and Chitte (1999) propose a new approach of dividing large knowledge sources into several smaller domain-based knowledge bases, called "sihlos", and using relatively narrow grammars in each individual sihlo. Only when the sihlo is visited are its grammar and other related properties downloaded to respond to the user. With the decrease of the scope of the knowledge source, the query language is shrunk, which can significantly improve speech-recognition accuracy.

The user can move from sihlo to sihlo by "speaking" hyperlinks. In this approach, semantic constraints that are coded in the syntax of each sihlo are chosen to reflect the fact that some semantic constraints are appropriate in one context and are inappropriate in others (Frost, 2002). For example, the constraint "people cannot orbit anything" might be appropriate in a sihlo which only answers questions about moons orbiting planets, while not appropriate in a sihlo about astronauts.

Chapter 4

Problems in Speech-Recognition

It seems that "speech-communication ability" is an instinct of human beings, for most human beings will be able to speak naturally at a certain age. But it is quite different for machines. Since countless human conversations proceed every day without any trouble, people do not realize that they have overcome many problems. In addition, many utterances can be understood only in particular context within some domains. However, all the above challenges and others, such as noise of the background and speaker variation, are very difficult for machines to tackle. Due to the large variability and flexibility of human speech and the speciality of machines (compared to human beings), there are many problems in the speech-recognition process.

4.1 Recognition Accuracy

Speech is recognized correctly if and only if the recognition result returned from the system is correctly corresponding to the user's speech input. There are two types of recognition errors: (1) utterance is not recognized at all; (2) utterance is mis-recognized. Since the first type of error (i.e., not recognized) might prompt the user to repeat, and the second type of error (i.e., mis-recognition) is likely to direct the user to the wrong results, it is very important to minimize the mis-recognitions. In general, statistical models have better recognition accuracy than grammar-based models. Good recognition accuracy is definitely one of the goals that numerous AI researchers have been pursuing.

4.2 Robustness

Robustness means the extent to which a system handles errors or "unexpected" input. Robustness is crucial in language systems for the reason that the inability or low performance in processing incorrect utterances will cause unacceptable degradation of the overall system (Ballim and Pallotta, 2000). Like human beings, the ideal spoken-language models should tolerate disfluencies, out of vocabulary words, incomplete or ungrammatical utterances, to some extent in speech communication. In reality, various uncertain and flexible factors of the spontaneous dialogue add more difficulties to speech recognition. Generally, statistical models outperform grammar-based models in the sense of robustness. However, there is still a lot left to be desired in state-of-the-art language models toward the goal of robustness.

4.3 Flexibility

An ideal spoken-dialogue system should be able to accept a user's flexible utterances, allow the user to supply extra information and make reasonable responses (Milward, 1999). While the fact is that the user may not realize the bounds of the domain, they may ask queries that are beyond the capability of the system. For example, the JUPITER weather-information system (developed in MIT) can only forecast short-term weather (Glass, 1999). So, if the user asks for "What is the weather in two months?" the JUPITER weather-information system cannot give an answer. Under such circumstance, the system is expected to give the user appropriate help to direct him/her to formulate an acceptable query. Since statistical models are based on huge training data and grammar-based models depend on the defining grammar, the former is more flexible than the latter.

4.4 Large vocabulary.

Many spoken-language systems are supported by a large vocabulary so that they can cover as many of the spontaneous utterances as possible. On the other hand, a large vocabulary can make the language system intractable; especially, the large number of categories due to the huge number of unrelated entries (Rosenfeld, 2000a) is a great challenge for speech recognition. For example, in a large vocabulary, there is no closer relation between BANK and LOAN than that with COUNTRY. The relative independence in a vocabulary leads to the huge intractable parameters, which is a problem existing in both statistical and grammar-based models.

In communication, human beings use knowledge about word relationships to help them recognize utterances. For example, if someone hears " the interest rate on bank loa… is 5%". They can fill in the missing letters and recognize "loa." as "loan". In this way, humans can recognize utterances involving huge vocabularies. However, computer-recognition systems that are based on simple syntax rules or statistical relationships between word occurances cannot handle huge vocabularies as well as human beings.

4.5 Brittleness across domains.

The efficiency of current language models depends much on the domains on which they are trained (Rosenfeld, 2000a). For example, a language model trained on business is not appropriate to recognize utterances about sports. Training of language models refers only to statistical models. Grammar-based models are totally brittle across a domain in the sense that a recognizer based on a grammar for one domain will not work at all in another domain unless they share common vocabulary and syntax rules.

4.6 False independence assumption.

While building a tractable language model, the state-of-the-art technology assumes some independence among different portions of the same document (Rosenfeld, 2000a). For example, the N-gram model (statistical model) determines the probability of the current word in a sentence only by the identity of the last N-1 words, which loses the long-term dependency. In particular, semantic constraints cannot be modeled with small N.

4.7 The Challenge

As mentioned in sub-section 1.4, accuracy and robustness are among the most important problems existing in speech-recognition technology. Usually, good accuracy is likely to lead to poor robustness; and vice versa. For example, the experiments of chapter seven show that the semantic grammar defines the smallest size of language and the best recognition accuracy but the poorest robustness; while the syntactic grammar defines a larger language size, better robustness, but lower accuracy than the semantic grammar; meanwhile, the word-sequence grammar, defining the largest language, is the most robust, but the least accurate among these three grammars (i.e., semantic, syntactic, and word-sequence grammar). The challenge is, how to achieve a good balance between accuracy and robustness.

Chapter 5

Grammars Used in the Experiment

Three types of grammar and their extensions are involved in the experiments. The experimental grammars are constructed based on the grammars created by Frost (2002), which define a language consisting of questions about the solar system, such as "who discovered phobos". The three unextended grammars are defined over the same vocabulary, but define different sets of expressions on account of the different ways of combining the words. Furthermore, a set of words is added to each vocabulary of these grammars for extension purpose, so that each extended grammar covers a larger language than the original grammar.

The experimental grammars are defined in the Java Speech Grammar Format (JSGF), which is a platform-independent, vendor-independent textual representation of grammars for use in speech recognition (Sun, 2000). A summary of JSFG features is listed in table 5 (Sun, 2000).

Table 5: summary of JSGF features

Feature	Purpose
Word or "word"	Words (terminals, tokens) need not be quoted
<rule>	Rule names (non-terminals) are enclosed in <>
[x]	Optionally x

(...)	Grouping
x y z ...	A sequence of x then y then z then ...
x | y | z ...	A set of alternatives of x or y or z or ...
<rule> = x; public <rule> = x;	A private and a public rule definition

Table 5: summary of JSGF features (Cont'd)

5.1 Simple Word-Sequence Grammars

A simple word-sequence grammar defines any sequence of words from the dictionary of some length, including neither semantics nor syntax. The unextended word-sequence grammar used in the experiment is given in figure 5.1.

```
1. /* 10-word word-sequence grammar
2. wordSequence_gram_ext1.gram */
3. grammar wordSequence_gram_ext1;
4. public <s> = <word>
    |<word> <word>
    |<word> <word> <word>
    |<word> <word> <word> <word>
    |<word> <word><word> <word><word>
    |<word> <word><word> <word><word> <word>
    |<word> <word><word> <word><word> <word><word>
    |<word> <word><word> <word><word> <word><word> <word>
    |<word> <word><word> <word><word> <word><word> <word><word>
    |<word> <word><word> <word><word> <word><word> <word><word> <word>
    |<simple>;
```

Figure 5.1: word-sequence grammar

Thereafter, the leftmost numbers in figures (figure 5.1, 5.2., 5.3) are line numbers. Line 1 and 2 are comments. Line 1 says that this simple word-sequence grammar defines any 10-word sequence. Line 2 tells the name of the grammar file. Line 3 marks the beginning of the JSGF grammar, defining the grammar name. Line 4 is a public rule, also the root rule of the grammar (the rule name is *s*), which consists of 10 alternatives of word sequence, i.e., 1-word sequence (*<word>*), or (denoted by "|") 2-word sequence (*<word><word>*), or 3-word sequence (*<word><word><word>*), and so on. Line 5 specifies some sentences that can be used as condition to direct the voice application. For example, in our testing applications, if the user says "goodbye", the voice application terminates. Line 6, defines the dictionary (vocabulary) of the language by listing all possible alternatives of the non-terminal *<word>*.

5.2 Syntactic Grammar

The syntactic grammar in the experiment only includes syntax, which defines the rules governing the structure of a language. The complete syntactic grammars used in the experiment are given in Appendix B. Figure 5.2 shows an extract. A brief explaination is provided later.

```
1.  /* syntax_gram_ext1.gram */
2.  grammar syntax_gram_ext1 ;
3.  public <s> = <linkingvb> <termph> [<transvb> by ] <termph>
              | <linkingvb> <termph> [<transvb> <preposition>] <termph>
              | <quest1> <sent>
              | ( who |what) <verbph>
              | ( which | how many ) <nouncla><verbph>;
4.  <sent> = <termph> <verbph>;
5.  <termph> = <stermph> | <stermph> (and | or) <stermph>;
6.  <stermph> = <pnoun> | <detph>;
7.  <verbph> = <transvbph> | <intransvb>;
8.  <transvbph> = ( <transvb> | <linkingvb> <transvb> by ) <termph> |
        ( <transvb> | <linkingvb> <transvb> <preposition> ) <termph>;
9.  <nouncla> = <adj> <cnoun> | <cnoun>;
10. <cnoun> = man | men | person | people | planet | planets |  moon | moons | mountain | mountains |
        crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal|
              metals| nonmetal | nonmetals | country | countries | capital | capitals | city |cities|
              continent| continents | river | rivers | lake | lakes ;
11. <intransvb> = spin | spins | orbit | orbits| orbited | exist | exists ;
12. <pnoun> = <pnoun_planet_moon_human>
```

```
            |<nonhuman_pnoun_chemical>
            |<space_program>
            |<earth_geography_domain>;
13. <transvb> = orbit | orbits | discover | discovered | neighbour | neighbours | neighboured | worship |
            worshiped | contain | contains | contained | find | finds | found;
```

Figure 5.2: extract of syntactic grammar

Line 3 is the root rule, which defines five kinds of questions by five alternatives. The first (/second) kind of question is started by a linking verb, then a term phrase, then a transitive verb and *by* (second kind of question uses preposition like "*in*" or "*on*", instead of *by*) (which is optional), then a term phrase. A term phrase is defined in line 5, which could use nouns in any category of planet, moon, human, geography, and so on. Sample sentence of this question type could be: "*is mars discovered by hall*" or "*is mars a moon*". The sample second kind of question could be: "*is hydrogen found on earth*". The third kind of question starts with a question word (*do|does| did*), then a term phrase, followed by a verb phrase (which uses transitive or intransitive verbs). The sample questions could be: "*Does phobos orbit mars*" or "*Does phobos spin*". The fourth and fifth kinds of question define questions such as "*who discovered phobos*" and "*how many moons orbit mars*".

5.3 Semantic Grammars

Semantics defines the relationships between symbols and their meanings. A semantic grammar directly encodes semantic constraints into the syntax of the grammar. The complete semantic grammars are given in Appendix B. An extract is shown in figure 5.3, and explained later.

```
1.  /* semantics_gram_ext1.gram */
2.  grammar semantics_gram_ext1;
3.  public <s> = <linkingvb> <termphrase_verbphrase>
            | is <pnoun> <pnoun>
```

| is <pnoun> (a|an) <nouncla>
| is <pnoun> (a|an) <nouncla> or (a|an) <nouncla>
| <quest1> <sent>
| (who) <animate_verbph>
| (what) <inanimate_verbph>
| (which | how many) <nouncla_verbph>
| (which | how many) <nouncla_verbph_other>;

4. <termphrase_verbphrase> = <nonhuman_termph_planet> <transvb_by_termph>
| <nonhuman_termph_moon> <animate_transvb> by <human_termph>
| <nonhuman_termph_other> <animate_transvb> by <human_termph>
| <nonhuman_termph_other> <animate_transvb> <preposition>
<nonhuman_termph_planet>
| <nonhuman_termph_other> <animate_transvb> <preposition>
<nonhuman_termph_moon>;

5. <transvb_by_termph> = <animate_transvb> by <human_termph>
| <inanimate_transvb> by <nonhuman_termph_moon>
| <inanimate_transvb_other> by <nonhuman_termph_other>;

6. <sent> = <human_termph> <animate_verbph>
| <nonhuman_termph_moon> <inanimate_verbph_active>
| <nonhuman_termph_planet> <inanimate_verbph_passive>
| <nonhuman_termph_moon> <inanimate_verbph_active_other>;

7. <nouncla_verbph> = <human_nouncla> <animate_verbph>
| <nonhuman_nouncla_moon> <animate_verbph_passive>
| <nonhuman_nouncla_planet> <animate_verbph_passive>
| <nonhuman_nouncla_moon> <inanimate_verbph_active>
| <nonhuman_nouncla_planet> <inanimate_verbph_passive> ;

8. <nouncla_verbph_other> = <nonhuman_nouncla_other> <animate_verbph_passive>
| <nonhuman_nouncla_other> <inanimate_verbph_passive_other>;

9. <inanimate_verbph> = <inanimate_verbph_active>
| <inanimate_verbph_passive>
| <inanimate_verbph_active_other>
| <inanimate_verbph_passive_other>;

10. <human_stermph> = <human_pnoun> | <human_detph> ;

11. <nonhuman_stermph_planet> = <nonhuman_pnoun_planet> | <nonhuman_detph_planet>;

12. <nonhuman_stermph_moon> = <nonhuman_pnoun_moon> | <nonhuman_detph_moon> ;

13. <nonhuman_stermph_other> = <nonhuman_pnoun_other> | <nonhuman_detph_other>;

14. <human_termph> = <human_stermph> | <human_stermph> (and | or) <human_stermph> ;

15. <nonhuman_termph_planet> = <nonhuman_stermph_planet>
| <nonhuman_stermph_planet> (and | or) <nonhuman_stermph_planet> ;

16. <nonhuman_termph_moon> = <nonhuman_stermph_moon>
| <nonhuman_stermph_moon> (and | or) <nonhuman_stermph_moon>;
17. <nonhuman_termph_other> = <nonhuman_stermph_other>
| <nonhuman_stermph_other> (and | or) <nonhuman_stermph_other>;
18. <animate_verbph> = <animate_transvbph>;
19. <inanimate_verbph_active> = <inanimate_transvbph_active> | <intransvb> ;
20. <inanimate_verbph_passive> = <inanimate_transvbph_passive>
| <intransvb>
| <inanimate_transvb> sun ;
21. <inanimate_verbph_active_other> = <inanimate_transvbph_active_other> | <intransvb_other>;
22. <inanimate_verbph_passive_other> = <inanimate_transvbph_passive_other> | <intransvb_other>;
23. <animate_transvb> = discover | discovers | discovered | find | finds | found ;
24. <animate_transvb_other> = worship | worshiped;
25. <inanimate_transvb> = orbit | orbits | orbited | neighbour | neighbours | neighboured;
26. <inanimate_transvb_other> = contain | contains | contained ;

Figure 5.3: extract of semantic grammar

Similar to the syntactic grammar in figure 5.2, the semantic grammar in figure 5.2 defines nine kinds of question by specifying nine alternatives in line 3. The primary difference between the semantic grammar and the syntactic grammar is that the former encodes some semantic constraints into the syntax of the grammar to ensure the correct semantics besides the correct syntax. In the semantic grammar, nouns are classifed into groups based on semantics, such as *human*, *moon*, *planet*, and *other* category; and verbs are gouped into *animate* and *inanimate*, so that it is possible to make the nouns and verbs match in semantics. For example, *hall* and *bond* are people, so, they belong to *human* group; *phobos* and *tritan* are *moons*; *earth* and *mars* are *planets*; *hydrogen* and *water* go to *other* category; *discover* is an *animate* verb; *orbit* and *spin* are *inanimate* verbs. So, if take a look at the first type of question, it can be traced down the first alternative in line 3, then the first alternative in line 4 to expand the non-terminal in line 3, then the first alternative in line 5 to expand the non-terminal in line 4, finally, we can have the sample question like: "*is mars discovered by hall*". In this way, the question like: "*is mars discovered by earth*" would never be generated by the semantic grammar in figure 5.3, though it is possible by the syntactic grammar in figure 5.2 (e.g. from the first alternative of line 3 in figure 5.2). In other word, the semantic grammar improves the recognition accuracy by including semantic constraints in syntax to reduce the language size.

5.4 Extending the grammars

To further investigate the features of different grammars, the three types of grammars discussed in sub-sections 5.1, 5.2, and 5.3 are extended. To simplify the expansion, we just add a set of words to each vocabulary of these grammars, so that each extended grammar covers a larger language than the original grammar. For example, in each original grammar, the *<country>* and *<capital>* rules both have size 6 (i.e., each language covers 6 countries and 6 capitals), and in the extended grammars, we add 181 countries and 92 capitals to the vocabulary (now, each language covers 187 countries and 98 capitals), so that the extended grammars cover larger languages than the original grammars.

Chapter 6

Grammar Design in Speech-Recognition

Since most commercial speech products are constructed using grammar-based technology (Knight *et al.*, 2001), grammar design becomes a crucial issue in speech recognition. A grammar specifies the legal utterances, i.e., the sequences of words that the user may say (Lucas, *et al.*, 1999)(VXML, 2000). Good grammar can achieve an appropriate compromise between accuracy and robustness. In our investigation, we observe that the size of the language defined by the grammar has a significant influence on speech recognition. For example, the direct encoding of semantic constraints into a syntactic grammar can reduce the language size, and the experiments show that this causes the speech-recognition accuracy to improve. However, constraining the language in this way leads to a reduction in robustness. Therefore, the grammar design is an extremely important topic in natural-language speech-recognition.

6.1 Grammar and Language Size

As a grammar defines a language, the size of the language is defined at the same time. Language size means how many possible utterances can be generated by the specific defining grammar.

A CFG can be defined as a set of rules that have a single atomic grammatical category on the left-hand side, and a sequence of atomic categories and words on the right-hand side (described in sub-section 2.2.1). To make it simple, we consider "word" or "category" as "symbol", and "expression" to consist of one or more "symbols". Then, the size of the defined language can be calculated in the following way:

1) The language size is the size of the root rule;
2) The size of right-hand expression is assigned to the size of left-hand expression;
3) If an expression is constructed by one symbol, the size of this expression is equal to the size of the symbol;
4) If an expression is composed by a group of symbols (a phrase), the size of the expression is the product of the size of each symbol in this group;
5) If an expression consists of alternate symbols (disjunction), the size of the expression can be obtained by summing of the sizes of all the alternative symbols;
6) Each single word has the size 1;

Consider the sample CFG grammar in figure 2.2.1, language size is calculated as shown in figure 6.1:

$<S>^{32} = <NP>^{4}\ <VP>^{8};$ // 32 = 4*8
$<NP>^{4} = <Det>^{2}\ <N>^{2};$ // 4 = 2*2
$<VP>^{8} = <V>^{2}\ <NP>^{4};$ // 8 = 2*4
$<Det>^{2} = \text{the} \mid \text{a};$ // 2
$<N>^{2} = \text{boy} \mid \text{door};$ // 2
$<V>^{2} = \text{opened} \mid \text{closed};$ // 2

Figure 6.1: language-size computation

Note: superscripts are used to denote the obtained size of the sub-language defined by the expressions; the following comments (starting with "//") denote the computation used to calculate the size.

The calculation process is explained as follows:

To calculate the language size defined by the grammar in figure 6.1, we start from the first rule of the grammar (also the root rule), which is composed of a complete phrase (grouping) with 2 symbols (i.e., <NP>, <VP>), so we need multiply these two symbols' sizes which need further computation. Then, we trace the symbol <NP> first, which is defined in the second rule. We can find that <NP> requires <Det> and <N>. As for <Det>, from the fourth rule of the above grammar, we know it has 2 alternative words (disjunction), which means the size of <Det> is 2 (i.e. 2=1+1); also, we can get the size of <N> by 2. Then, we come back to the second rule to calculate the size of <NP> by multiply the sizes of <Det> and <N> (i.e., 4=2*2), i.e., the size of <NP> is 4. Similarly, we can get the size of <VP> by multiply the sizes of <V> and <NP> (i.e., 8 = 2*4). Finally, the root rule size is obtained by multiplying the sizes of <NP> and <VP> (i.e., 32 = 4*8). Therefore, the size of the language defined by the above sample grammar is 32, which means it can accept 32 utterances, such as "A boy opened the door."

The details of language-size computation of the grammars in our experiment are given in Appendix B.

6.2 Interpretation of Language Size

Since the left-hand side symbol in CFG rule can be freely replaced by the right-hand side rules (refer to sub-section 2.2.1), we can obtain the following equivalent in figure 6.2 (1) to the sample grammar in figure 6.1:

$$<S>^{32} = <Det>^{2} <N>^{2} <V>^{2} <Det>^{2} <N>^{2}$$

Figure 6.2 (1): variation of the grammar in figure 6.1

So, from figure 6.2 (1), the grammar in figure 6.1 actually states that a valid sentence is composed of a determiner (<Det>), a noun (<N>), a verb (<V>), a determiner (<Det>), and a noun (<N>). In this specific example, each word has 2 valid alternatives. So, there are 2*2*2*2*2 = 32 possible valid sentences in the language defined by the sample grammar. Furthermore, the language size is the size of the root rule, which is the product of each word-candidate size (word-candidate sizes means how many possible alternatives for this word candidate, e.g., size of <N> is 2).

Then, if we take d as the average depth, i.e., the average length of a sentence in the language defined by a grammar, take b as the average branching factor, i.e., the average number of word candidates. In the above example, the average depth (average length of a sentence) d is 5, the average word-candidate size (branching factor) b is 2, and the language size s is equal to 2^5 (i.e., 32).

In this specific example in figure 6.2 (1), each non-terminal in the grammar rule has the same number of word candidates, and the grammar is equivalent to one rule. So, the average branching factor and the sentence length are obvious. In general, it is hard to know the precise sentence length and the branching factor. Since the language size can be precisely calculated using the method discussed in sub-section 6.1, if either branching factor or language length is available, the other is able to be obtained using the formula $s = b^d$. Assuming all terminals and non-terminals in the grammar which has been assigned weight in a particular context will all occur with equal probability, the general average branching factor can be estimated in the following way:

1) The branching factor for an expression is the number of its successors;
2) The left-hand side expression takes the first alternatives on the right-hand side expression as successors.

3) If an expression has successors, it will be taken into account for average branching factor;
4) Each terminal (word) has the size 1;
5) Average branching factor can be obtained by summing up all the branching factors, then divided by the number of expressions for which branching factors have been calculated.

Consider a general grammar in figure 6.2 (2), where the leftmost numbers are line numbers, superscripts are used to denote the branching factors of the preceding expressions (in the left-hand side, superscripts directly denote the branching factor of non-terminals); the underlined superscripts are used for average branching-factor computation.

1. <sent>$^{\underline{3}}$ = <ex>^{3} <w>$^{\underline{9}}$;
2. <ex>^{3}= <t1>^{3} and$^{\underline{1}}$ <t2>$^{\underline{2}}$;
3. <w>^{9}= <t2>^{2} or$^{\underline{1}}$ <t3>$^{\underline{5}}$
   | <t4>^{7} ;
4. <t1>^{3} = w1 | w2 | w3 ;
5. <t2>^{2} = x1 | x2 ;
6. <t3>^{5} = n1 | n2 | n3 | n4 | n5 ;
7. <t4>^{7} = a1 | a2 | a3 | a4 | a5 | a6 | a7 ;

Figure 6.2 (2): sample grammar with branching factors

The first rule in line 1 is the root rule of this grammar. The expression *sent* is composed of *ex* followed by *w*. The branching factor (number of successors) for *sent* is the number of possible alternatives of *ex*, which can be obtained by calculate the number of words in *t1* according to the rule in line 2. *t1* has 3 alternatives (line 4), so *ex* has the branching factor 3 in line 2, and *sent* has the branching factor 3 in line 1. In the right-hand side of line 1, the successor of *ex* is *w*, which is defined in line 3 with *t2* and *t4* as successors. Since *t2* has 2 alternatives defined in line 5 and *t4* has 7 alternatives defined in line 7, *w* has 9 successors in line 3(i.e.,9=2+7), which will be passed to *ex* in line 1. In this way, the other branching factors can be obtained shown in figure 6.2 (2). The average branching factor is calculated based on the expression with successors (numbers underlined in figure 6.2):

$$b = (3+9+1+2+1+5) / 6 = 3.5$$

Note that, this method is not suitable for the word-sequence grammar which consists of word sequences. The branching factor for the word-sequence grammar is always equal to the number of words in the dictionary.

Since the grammar has defined b^d possible valid choices for speech input, we consider the following two cases: (1) If the branching factor (b) is a constant, which means the average number of word candidate are supposed a constant, the language size will be in exponential increase with d, the average length of an utterance in the language. (2) If the average length of a sentence d is fixed, then the increase of the branching factor b, i.e., the word-candidate size, will induce a polynomial increase in the language size (b^d).

In practice, natural-language-database queries have a stable average utterance length (d), so the number of word candidates (i.e., the branching factor b) plays a prominent role in language size. In other words, increasing the vocabulary in a database query system can increase the language size considerably. For example, assuming an average utterance length (d) of 5 and an average branching factor (b) of 2, we have 32 (i.e., 2^5) utterances. If we keep the same utterance length ($d=5$), and double the branching factor (vocabulary size, $b=4$), the language size increases to 1024 (i.e., 4^5).

Note that this assumes that the average branching factor is directly proportional to the vocabulary size. Our experiment shows that this assumption is not valid in all applications.

6.3 The Significance of Language Size

6.3.1 Influence on Speech-Recognition Accuracy:

We hypothesize that, in the same domain, the smaller the size of the defined language, the higher is the speech-recognition accuracy. This hypothesis is examined from both horizontal- and vertical- direction comparisons. Consider the following three general types of grammar (discussed in chapter 5): one is the semantic grammar, which directly encodes semantic constraints into syntax rules of the CFG; the second is the syntactic grammar, which contains only syntax rules; the third one is a simple word-sequence grammar, which includes neither semantics nor syntax. With the same vocabulary, the semantic grammar defines a language with the smallest size, the syntactic grammar defines a larger-size language, and word-sequence grammar covers the largest language. The horizontal-direction comparison occurs between these three different types of grammar. We found that recognition accuracy increases with the decrease of the language size, which means that the semantic grammar is the most accurate, the second accurate grammar is the syntactic grammar, and the word-sequence grammer has the worst recognition accuracy.

In a second experiment, these three types of grammar are each extended to enlarge the language size by adding more words to the vocabulary. Then the vertical comparison is available between the original grammar (e.g. syntactic grammar) and the later extended grammar (e.g. extended syntactic grammar). The result was that the extended grammar has lower recognition accuracy than the original one, for it has increased the language size. This result was expected. The interesting part of this result is that recognition accuracy remained quite good for the semantic grammar despite significant increase in the language size.

We consider the speech recognition to be correct only when the recognition result returned by the speech-recognition system is in accordance with the user's speech input. On the other hand, if the speech recognition is not correct, there may occur two possible cases: (1) the system mis-recognizes the user's speech input into something else. For example, the user says "Who discovered m*ars*?", the system returns with "Who discovered m*imas*?"; (2) the system cannot recognize the user's speech input. For example, the system responds to the user with "Sorry, I didn't understand" .

While designing a grammar in speech recognition, we expect a good accuracy. However, it is impractical to require a speech-recognition system to have 100% recognition accuracy with current technology. We would like the system to report the information of "not recognized" (such as the response to user "Sorry, I didn't understand" in our experiment) rather than the mis-recognition (incorrect recognition), if the speech recognition is not correct. The reason is that "Sorry, I didn't understand" may prompt the user to repeat and get the correct speech input; while a mis-recognition is likely to pass the system with wrong information and lead the user to some wrong results.

Our experiments show that, with the semantic grammar, the speech-recognition system makes fewer mis-recognitions than the syntactic grammar with both semantically and syntactically correct utterances, and the word-sequence grammar has the most mis-recognitions among these three types of grammar, which proves that the semantic grammar outperforms the syntactic grammar and word-sequence grammar in recognition accuracy with both semantically and syntactically correct utterance inputs.

Therefore, if speech-recognition accuracy is the most important feature for a speech-recognition system, the grammar designer would try to restrict the language grammar to a size as small as possible, which, for instance, can be implemented by directly encoding semantics into the syntax of the grammar.

6.3.2 Influence on Robustness

Although it is effective to get good recognition accuracy by adding more constraints to shrink a language in size, it is likely to make the speech-recognition system lose robustness. Intuitively, when we are reducing a language in size, we are adding more constraints to the language, which implies that more utterances (that are valid in the original grammar) are discarded due to their invalidity in the shrunk language. If the discarded utterances are indeed not correct in some sense (e.g. semantics), the shrunk language is achieving a more accurate performance.

However, not all users might be clear about the domain of the speech-recognition system. It is possible they may ask some out-of-range questions. If the system just discards such input, it may confuse the users if they indeed don't realize what's wrong with their inputs. For example, if in a solar system with the semantic grammar, the user asks the system "Does mars orbit phobos?", which is absolutely syntactically correct, but semantically incorrect for a planet cannot orbit a moon in common sense. Then the solar system (with the semantic grammar) refuses such speech input due to its semantic incorrectness. But the user has not realized the problem and keeps asking such questions on account of their syntactic correctness. At such time, if the user cannot get any help from the system, communication may get stuck.

Based on the above reason, sometimes, we expect the system to be able to accept some "incorrect" input, and provide the users with proper guidance to direct them back to the correct track on the speech-recognition system. That's where the robustness lies. At this point, the speech-recognition system with the larger language size (e.g., defined by syntactic grammar) outperforms that with a smaller language size (e.g., defined by semantic grammar).

In grammar design, if the application requires more robustness than accuracy, a syntactic grammar, instead of a semantic grammar, can be considered, for the reason that the syntactic grammar is capable of accepting the utterances that are syntactically correct but semantically incorrect, which are rejected by a semantic grammar.

Generally, a trade-off exists between recognition accuracy and robustness, and how to balance the speech-recognition accuracy and robustness is a significant future task.

Chapter 7

Experiment and Result

7.1 Overview of the Experiment

Our preliminary experiment was carried out to investigate the significance of grammar design in speech-recognition. Six grammars, i.e., semantic grammar, syntactic grammar, word-sequence grammar, extended semantic grammar, extended syntactic grammar, and extended word-sequence grammar (which are discussed in chapter 5), and two people, one English male and one non-English female, were involved in the experiment. The experimental subjects (people) speak to the experimantal system at a normal speed, pronouncing every word as clearly as possible, like a normal user to a speech-recognition system. They adjust their pronunciation by experience. All experiments are conducted consistently in the same experimental location, with the same background.

A summary of the language sizes is given later in table 7.4.2, and a detailed computation process of language sizes is given in Appendix B. The horizontal comparison is made among the semantic grammar, syntactic grammar, and word-sequence grammar, also among the extended semantic grammar, extended syntactic grammar, and extended word-sequence grammar. The vertical comparison is conducted between semantic grammar and extended semantic grammar, syntactic grammar and extended syntactic grammar, word-sequence grammar and extended word-sequence grammar.

At the beginning of the experiment, each subject (person) is trained by a set of utterances, in order that they can get used to the testing system and make their pronunciation acceptable to the system. Generally, people will adjust their pronunciation during the practice, so that it is gradually accepted more and more by the system. Therefore, we include the training part in the experiment to minimize the order effect, which means that the order that the grammar is tested will not affect its recognition accuracy. The training set is customized as a set of syntactically correct questions. Each person is trained by going through this set ten times using the syntax grammar.

The testing utterance inputs are categorized into the following three categories: a semantics set, which is composed of the questions that are both semantically and syntactically correct (e.g., "Is titania a mountain"); a syntax set, which consists of the questions that are only syntactically correct, but semantically incorrect (e.g., "Does a mountain contain a moon"); and a word-sequence set, which covers the utterances that are neither semantically nor syntactically correct, they are only word sequences (e.g., "Is mountain contain moon"). All three types of testing utterances are checked by text-mode testing to ensure they are categorized correctly.

To further minimize the order affect, the user will go through the three sets of questions for each grammar twice in different sequences, for example, in the order like this: (extended) semantic grammar, (extended) syntactic grammar, and (extended) word-sequence grammar.

7.2 Experiment Environment

Our experiments were carried out using IBM WebSphere Voice Server SDK which can be freely downloaded from IBM (2002) on Windows XP platform. The grammars were written in JSGF, which can be embedded in VoiceXML pages as in-line grammar segments, or stored in separate files locally or remotely. The testing applications were written in VoiceXML (Voice eXtensible Markup Language), which is a programming language for building interactive voice applications (Tellme, 2002). VocieXML is an XML-based markup language for creating distributed voice applications, much as HTML is a markup language for creating distributed visual applications (IBM, 2001).

The IBM WebSphere Voice Server SDK provides a spoken equivalent to visual browsing, such as supporting VoiceXML to web application development activities (IBM, 2001). It can be used to create and test Web-based voice applications based on the workstation's speakers to play audio output. Also, the developers can input data using the workstations' microphones, prerecorded audio files, or the IBM WebSphere Voice Server SDK's DTMF Simulator (to simulate any telephone key input) (IBM, 2001). The SDK also supports text-mode and automated testing.

The IBM WebSphere Voice Server SDK consists of a speech browser that interprets VoiceXML markup, IBM ViaVoice Speech Recognition and Text-To-Speech (TTS) engines for accepting voice input and generating synthesized speech output, sample applications, and other tools for developing and testing VoiceXML applications (IBM, 2001).

The hardware configuration is as follows:

- 256 MB RAM;

- 30GB hard drive;
- A display adapter with a setting of greater than 256 colors;
- A Microsoft Windows 2000 compatible, 16-bit, full-duplex sound card (with a microphone input jack) with good recording quality;
- An average microphone.

7.3 Experiment Results

The experiment results are given with respect to subject (people), grammar, testing utterance set, and recognition result. The experiment result is denoted as follows: C: Correctly recognized, I: Incorrectly recognized, N: Not recognized at all. The testing order is considered in the experiment to ensure that the results are not unduly affected by the testing order.

Note that, in the experiments, person #1 went though all the semantic grammars and syntactic grammars using the semantics set and the syntactic set, and some of the experiments using the word-sequence grammars and the word-sequence testing utterance set; person #2 went through all the experiments using all types of testing utterances and grammars. The experiment results are represented by two formats: a table and a graph. Partial experiment results in detail are given in Appendix D.uo

In the experiment, the recognizer was tailored with a grammar. The subject read the queries (utterances), and the recognition results were recorded. For example, given a small set of three queries as follows:

1. Was phobos discovered by a person?
2. Is titania a mountain?
3. Does Saturn contain a crater?

Supposing person #2 uses the extended semantic grammar to test the above testing utterance set, the testing voice application is called "semantics_test_ext2.vxml", the command to run this application is: vsaudio_en_US.bat semantics_test_ext2.vxml. The screen shot is shown in figure 7.3 (1).

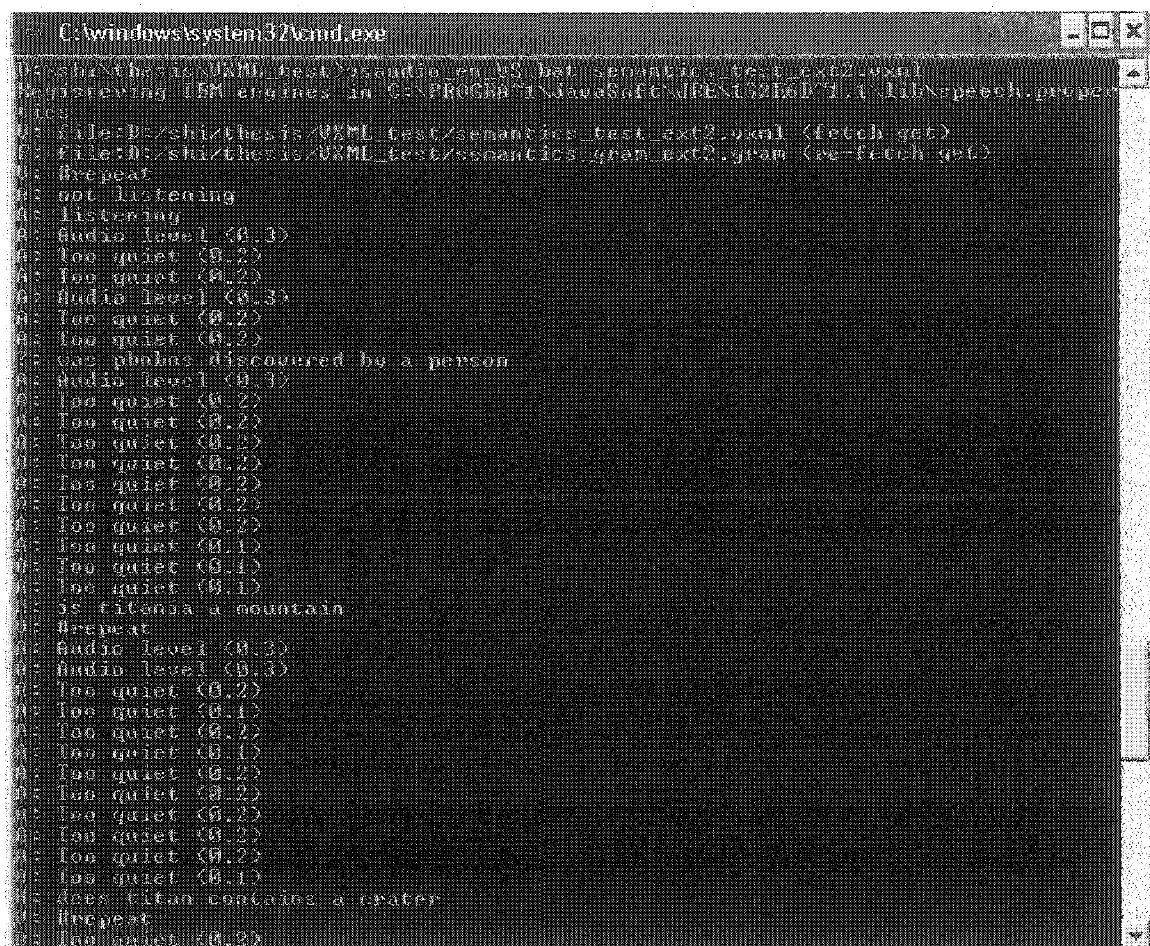

Figure 7.3 (1): sample screen shown of the experiment

The format of trace entries in the IBM Voice Server SDK is defined with "Code: Message" as shown in table 7.3 (1) (IBM, 2001) :

Table 7.3 (1): Trace code in IBM Voice Server SDK

Code	Message
A	Logged when the VoiceXML browser detects audio input, but the speech

	recognition engine does not return a recognized phrase; this may be due to breath or background noise. The message column contains audio level messages.
F	Logged when the VoiceXML browser fetches a resource such as a grammar file, an audio file, or a script. The message column contains the URI of the file, and whether it was fetched from the server or was in the cache.
H	Logged when the user responds using voice input. The message column displays the word or phrase that was recognized by the speech recognition engine.
V	Logged when the VoiceXML browser fetches a .vxml file. The message column contains the URI of the file, and whether it was fetched from the server or was in the cache.
?	Logged when the speech recognition engine determines that the user said something, but the confidence level is not high enough to justify using the results. In response, the VoiceXML browser throws a nomathc event. The message column contains the word or phrase that was recognized.

Table 7.3 (1): Trace code in IBM Voice Server SDK (Cont'd)

Refer to figure 7.3 (1), the "?: Was phobos discovered by a person" on the screen shot (i.e., in trace log) means that the user's speech input "Was phobos discovered by a person" could not be recognized by the speech engine due to an insufficiently high confidence level, which is denoted by "N" in our experiment result record. The "H: Is titania a mountain" is the recognition result returned by the speech recognition engine, also that's exactly what the user has said. Under such circumstance, we consider this recognition result to be correct, and denote it with "C" in our experiment result record. The third utterance asked by the user was "Does *saturn* contain a crater", but the speech recognizer recognized it as "Does *titan* contains a crater". Actually, the speech engine mis-recognized the user's utterance input, we record it with "I" in the recognition result.

7.3.1. Table Representation

The following tables contain summaries of the raw results of the experiments which are given in Appendix D. To remind the reader what the rows and columns stand for, we summarize some of the discussion so far in this chapter:

1. Three initial grammars were used to configure the speech recognizer: a semantic grammar that defines the smallest language, a syntactic grammar that defines a larger language consisting of syntactically correct utterances, and a word-sequence grammar.
2. The three grammars were all extended to include a larger vocabulary and the experiment was repeated.
3. Three sets of utterances were used. A semantics set, which includes testing utterances that are both semantically and syntactically corret; a syntax set, which contains testing utterances that are only syntactically correct but semantically incorrect; and a word-sequence set, which covers word sequences that are neither semantically correct nor syntactically correct.

In addition, these tables show the experiment results after we changed the grammars to accommodate the person-specific problem (which is discussed later in sub-section 7.4.4).

Table 7.3 (2): Experiment result using grammars BEFORE extension – Person #1

Person #1 (English male):

Utterance	Grammar	Testing Order	Total Test Utterances (#)	Correctly Recognized (#)	Incorrectly Recognized (#)	Not Recognized (#)	Correct Percentage (%)	Incorrect Percentage (%)	Not Recog' Percentage (%)
Semantics Set	Semantic Grammar	# 1	73	60	3	10	82.2	4.1	13.7
		# 3	73	60	4	9	82.2	5.5	12.3
	Average		73	60	3.5	9.5	82.2	4.8	13.0
	Syntactic Grammar	# 2	73	60	11	2	82.2	15.1	2.7
		# 4	73	57	11	5	78.0	15.1	6.9
	Average		73	58.5	11	3.5	80.1	15.1	4.8
Syntax Set	Semantic Grammar	# 1	25	0	5	20	0	20	80
		# 3	25	0	8	17	0	32	68
	Average		25	0	6.5	18.5	0	26	74
	Syntactic Grammar	# 2	25	22	0	3	88	0	12
		# 4	25	22	0	3	88	0	12
	Average		25	22	0	3	88	0	12

Table 7.3 (3): Experiment result using grammars BEFORE extension – Person #2

Person #2 (non-English female):

Utterance	Grammar	Testing Order	Total Test Utterances (#)	Correctly Recognized (#)	Incorrectly Recognized (#)	Not Recognized (#)	Correct Percentage (%)	Incorrect Percentage (%)	Not Recog' Percentage (%)
Semantics Set	Semantic Grammar	# 1	73	48	2	23	65.8	2.7	31.5
		# 4	73	52	2	19	71.2	2.7	26.0
	Average		73	50	2	21	68.5	2.7	28.8
	Syntactic Grammar	# 2	73	36	10	27	49.3	13.7	37.0
		# 5	73	41	10	22	56.2	13.7	30.1
	Average		73	38.5	10	24.5	52.7	13.7	33.6
	Word Sequence	# 3	73	9	46	18	12.3	63.0	24.7
		# 6	73	9	41	23	12.3	56.2	31.5
	Average		73	9	43.5	20.5	12.3	59.6	28.1

Syntax Set	Semantic	# 1	25	0	4	21	0	16	84
	Grammar	# 4	25	0	5	20	0	20	80
	Average		25	0	4.5	20.5	0	18	82
	Syntactic	# 2	25	9	4	12	36	16	48
	Grammar	# 5	25	12	1	12	48	4	48
	Average		25	10.5	2.5	12	42	10	48
	Word	# 3	25	2	10	13	8	40	52
	Sequence	# 6	25	2	12	11	8	48	44
	Average		25	2	11	12	8	44	48
Word-Sequence Set	Semantic	# 1	24	0	3	21	0	12.5	87.5
	Grammar	# 4	24	0	2	22	0	8.3	91.7
	Average		24	0	2.5	21.5	0	10.4	89.6
	Syntactic	# 2	24	0	7	17	0	29.2	70.8
	Grammar	# 5	24	0	7	17	0	29.2	70.8
	Average		24	0	7	17	0	29.2	70.8
	Word	# 3	24	4	12	8	16.7	50.0	33.3
	Sequence	# 6	24	3	15	6	12.5	62.5	25.0
	Average		24	3.5	13.5	7	14.6	56.3	29.1

Table 7.3 (3): Experiment result using grammars BEFORE extension – Person #2 (Cont'd)

Table 7.3 (4): Experiment result using grammars AFTER extension – Person #1

Person #1 (English male):

Utterance	Extended Grammar	Testing Order	Total Test Utterances (#)	Correctly Recognized (#)	Incorrectly Recognized (#)	Not Recognized (#)	Correct Percentage (%)	Incorrect Percentage (%)	Not Recog' Percentage (%)
Semantics Set	Semantic Grammar	# 1	73	59	6	8	80.8	8.2	11.0
		# 3	73	56	7	10	76.7	9.6	13.7
	Average		73	57.5	6.5	9	78.8	8.9	12.3
	Syntactic Grammar	# 2	73	55	9	9	75.4	12.3	12.3
		# 4	73	53	11	9	72.6	15.1	12.3
	Average		73	54	10	9	74.0	13.7	12.3
	Wd Seq	# 5	73	14	46	13	19.2	63.0	17.8
	Average		73	14	46	13	19.2	63.0	17.8
Syntax Set	Semantic Grammar	# 1	25	0	4	21	0	16.0	84.0
		# 3	25	0	4	21	0	16.0	84.0
	Average		25	0	4	21	0	16.0	84.0
	Syntactic Grammar	# 2	25	20	1	4	80	4	16
		# 4	25	21	0	4	84	0	16
	Average		25	20.5	0.5	4	82	2	16

Table 7.3 (5): Experiment result using grammars AFTER extension – Person #2

Person #2 (non-English female):

Utterance	Extended Grammar	Testing Order	Total Test Utterances (#)	Correctly Recognized (#)	Incorrectly Recognized (#)	Not Recognized (#)	Correct Percentage (%)	Incorrect Percentage (%)	Not Recog' Percentage (%)
Semantics Set	Semantic Grammar	# 1	73	43	5	25	58.9	6.8	34.3
		# 3	73	46	4	23	63.0	5.5	31.5
		# 5	73	45	6	22	61.6	8.2	30.1
		# 7	73	44	3	26	60.3	4.1	35.6
	Average		73	44.5	4.5	24	61.0	6.2	32.8
	Syntactic Grammar	# 2	73	29	9	35	39.7	12.3	48.0
		# 4	73	32	14	27	43.8	19.2	37.0
		# 6	73	38	10	25	52.1	13.7	34.2
	Syn Gram	# 8	73	37	15	21	50.7	20.5	28.8
	Average		73	34	12	27	46.6	16.4	37.0
	Word Sequence	# 9	73	4	49	20	5.5	67.1	27.4
		# 10	73	3	53	17	4.1	72.6	23.3
	Average		73	3.5	51	18.5	4.8	69.9	25.3

Syntax Set	Semantic	# 1	25	0	5	20	0	20.0	80.0
	Grammar	# 3	25	0	3	22	0	12.0	88.0
	Average		25	0	4	21	0	16.0	84.0
	Syntactic	# 2	25	12	3	10	48.0	12.0	40.0
	Grammar	# 4	25	12	1	12	48.0	4.0	48.0
	Average		25	12	2	11	48.0	8.0	44.0
	Word	# 5	25	1	15	9	4.0	60.0	36.0
	Sequence	# 6	25	0	14	11	0	56.0	44.0
	Average		25	0.5	14.5	10	2.0	58.0	40.0
Word-Sequence Set	Semantic	# 1	24	0	1	23	0	4.2	95.8
	Grammar	# 4	24	0	1	23	0	4.2	95.8
	Average		24	0	1	23	0	4.2	95.8
	Syntactic	# 2	24	0	6	18	0	25.0	75.0
	Grammar	# 5	24	0	7	17	0	29.2	70.8
	Average		24	0	6.5	17.5	0	27.1	72.9
	Word	# 3	24	1	18	5	4.2	75.0	20.8
	Sequence	# 6	24	2	14	8	8.3	58.4	33.3
	Average		24	1.5	16	6.5	6.3	66.7	27.0

Table 7.3 (5): Experiment result using grammars AFTER extension – Person #2 (Cont'd)

7.3.2 Further Summary and Graphical Representation of the Results

To study the general trend of the experimental results, we take the average recognition results of each subject using each grammar under each testing utterance set.

Table 7.3.2 (1): the "Correct" feature using the semantics set

Grammars	Person #1	Person #2	Average
Semantic	82.2	68.5	75.35
Syntactic	80.1	52.7	66.4
Word Sequence		12.3	12.3
Ext. Semantic	78.8	61	69.9
Ext. Syntactic	74	46.6	60.3
Ext. Word Sequence	19.2	4.8	12

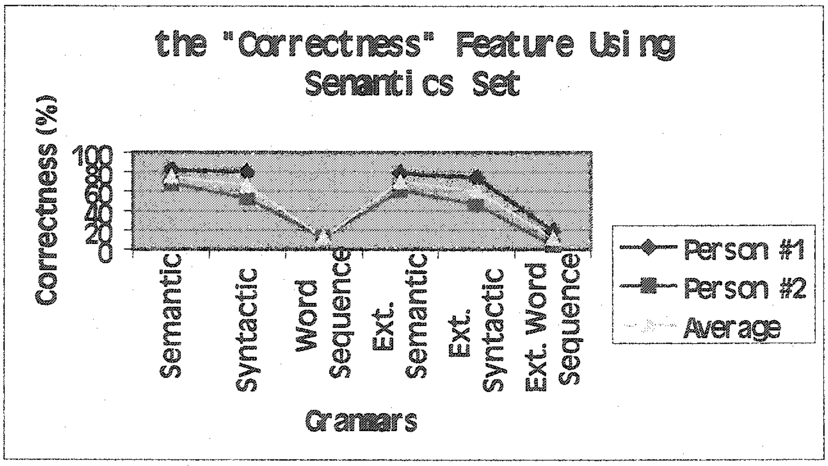

Figure 7.3.2 (1): the "Correct" feature using the semantics set

The data above shows that: for both subjects, and for the original and extended grammars, the semantic grammar has higher accuracy than the other grammars; the word-sequence grammar has much lower accuracy than the other grammars, for queries that are semantically as well as syntactically correct.

Table 7.3.2 (2): the "Incorrect" feature using the semantics set

Grammars	Person #1	Person #2	Average
Semantic	4.8	2.7	3.75
Syntactic	15.1	13.7	14.4
Word Sequence		59.6	59.6
Ext. Semantic	8.9	6.2	7.55
Ext. Syntactic	13.7	16.4	15.05
Ext. Word Sequence	63	69.9	66.45

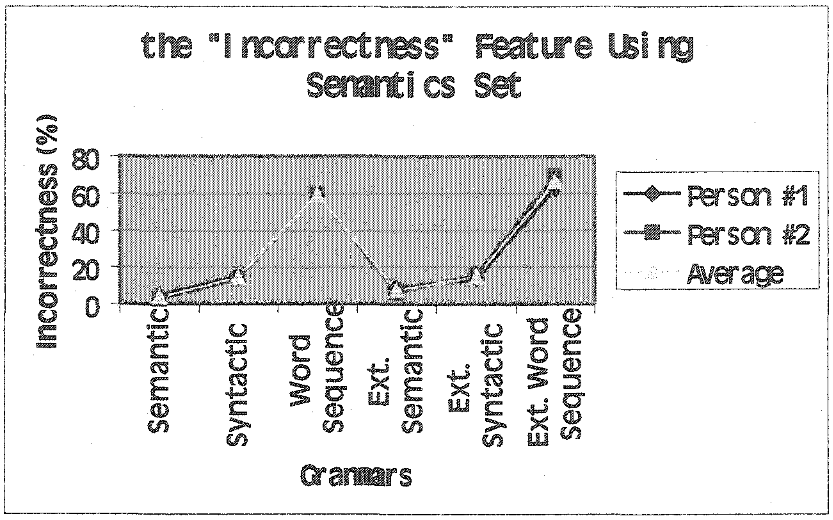

Figure 7.3.2 (2): the "Incorrect" feature using the semantics set

The data above shows that: for both subjects, and for the original and extended grammars, the semantic grammar has the lowest mis-recognition rate, and the word-sequence grammar has the highest mis-recognition rate, for queries that are semantically and syntactically correct.

Table 7.3.2 (3): the "Not recognized" feature using the semantics set

Grammars	Person #1	Person #2	Average
Semantic	13	28.8	20.9
Syntactic	4.8	33.6	19.2
Word Sequence		28.1	28.1
Ext. Semantic	12.3	32.8	22.55
Ext. Syntactic	12.3	37	24.65
Ext. Word Sequence	17.8	25.3	21.55

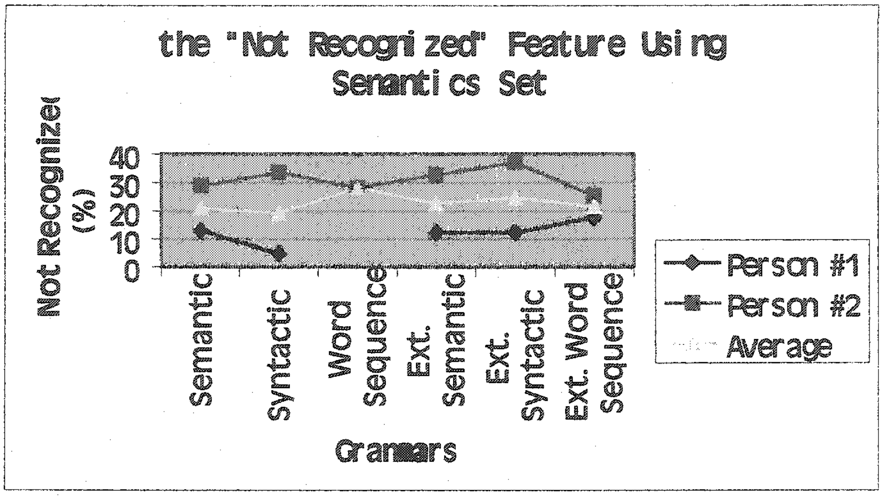

Figure 7.3.2 (3): the "Not recognized" feature using the semantics set

Though it seems that there is not an obvious trend for the "not recognized" feature using semantics set, we can see the "not recognized" rates of semantic grammar and syntactic grammar are approximately the same. The figure also shows that the person #2 has encountered more "not recognized" than person #1, which may be due to their experience with English.

The above data shows that if the user asks the queries that are both semantically correct and syntactically correct, for both subjects, and for the original and extended grammars, the experiment results can be summarized as follows:

- The semantic grammars have the highest correct recognition rate and the fewest incorrect recognition (mis-recognition) rate.
- The word-sequence grammar has significantly less accuracy and higher mis-recognition rate than the other grammars;
- The semantic grammar has approximately the same percentage of "not recognized" as the syntactic grammar.

Table 7.3.2 (4): the "Correct" feature using the syntax set

Grammars	Person #1	Person #2	Average
Semantic	0	0	0
Syntactic	88	42	65
Word Sequence		8	8
Ext. Semantic	0	0	0
Ext. Syntactic	82	48	65
Ext. Word Sequence		2	2

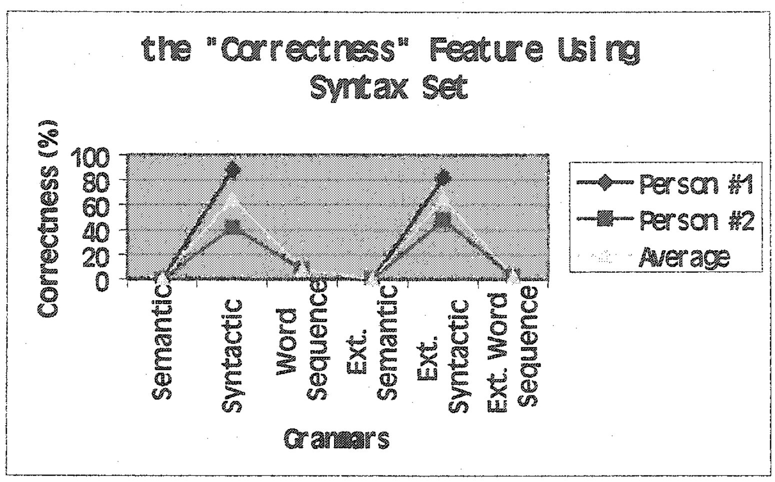

Figure 7.3.2 (4): the "Correct" feature using the syntax set

The above data shows that: if the user asks the queries in syntax set, which are only syntactically correct but semantically incorrect, the syntactic grammar, as well as its extension, has higher recognition accuracy than the other grammars. The semantic grammars cannot recognize any query in the syntax set, and the word-sequence grammars have very low-recognition accuracy.

Table 7.3.2 (5): the "Incorrect" feature using the syntax set

Grammars	Person #1	Person #2	Average
Semantic	26	18	22
Syntactic	0	10	5
Word Sequence		44	44
Ext. Semantic	16	16	16
Ext. Syntactic	2	8	5
Ext. Word Sequence		58	58

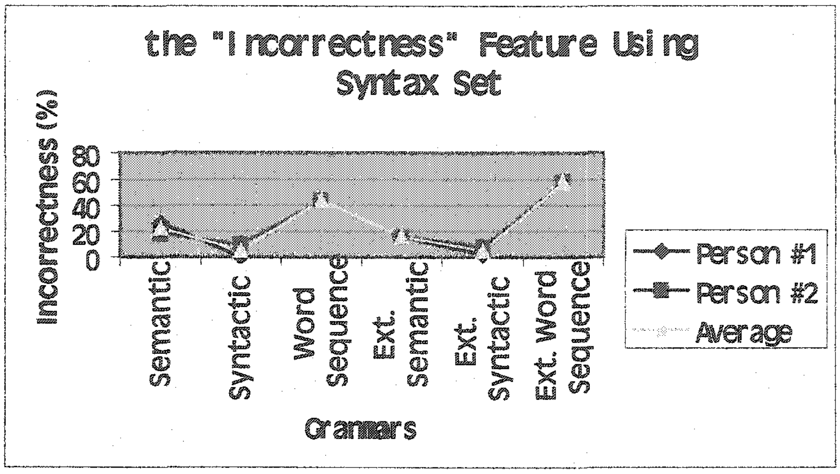

Figure 7.3.2 (5): the "Incorrect" feature using the syntax set

The above data shows that: for the queries in syntax set, which are only syntactically correct but semantically incorrect, for both subjects, and for the original and extended grammars, the syntactic grammar has the lowest mis-recognition rate. The word-sequence grammar has the highest mis-recognition rate.

Table 7.3.2 (6): the "Not recognized" feature using the syntax set

Grammars	Person #1	Person #2	Average
Semantic	74	82	78
Syntactic	12	48	30
Word Sequence		48	48
Ext. Semantic	84	84	84
Ext. Syntactic	16	44	30
Ext. Word Sequence		40	40

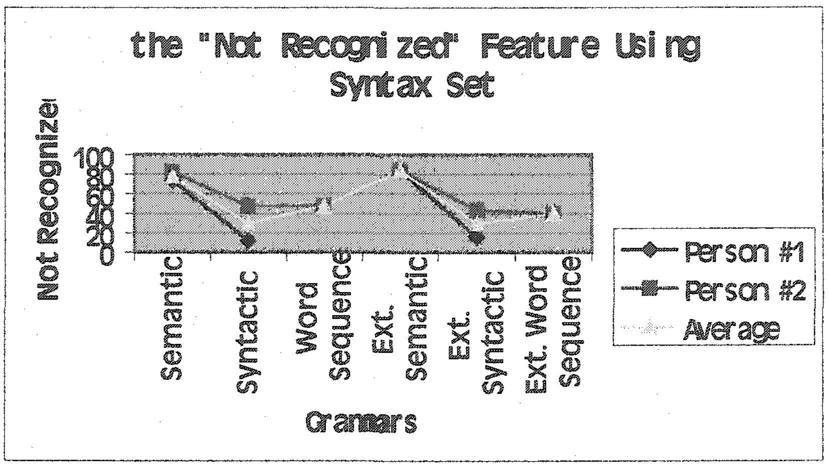

Figure 7.3.2 (6): the "Not recognized" feature using the syntax set

The above data shows that: if the queries are only syntactically correct but semantically incorrect, the semantic grammar is more likely to report "not recognized" information to the user than the other grammars. The syntactic grammar has the lowest "not recognized" rate.

Figures 7.3.2 (4), 7.3.2 (5), and 7.3.2 (6), as well as tables 7.3.2 (4), 7.3.2 (5), and 7.3.2 (6) show that if the queries are only syntactically correct, but semantically incorrect, the experiment results can be summarized as follows:

- The syntactic grammars have the highest accuracy, the lowest mis-recognition rate, and the lowest "not recognized" rate.
- The semantic grammar cannot recognize any such kind of queries, and has the highest "not recognized" rate.
- The word-sequence grammar has the most mis-recognitions;

Table 7.3.2 (7): the "Correct" feature using the word-sequence set

Grammars	Person #2
Semantic	0
Syntactic	0
Word Sequence	14.6
Ext. Semantic	0
Ext. Syntactic	0
Ext. Word Sequence	6.3

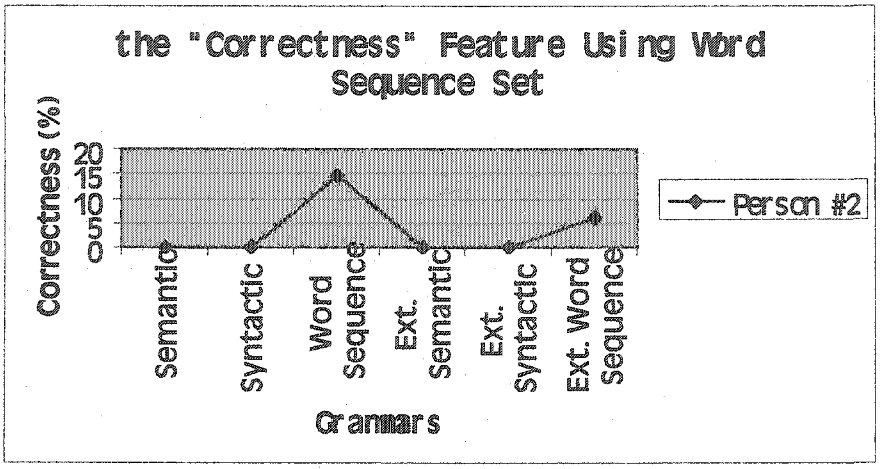

Figure 7.3.2 (7): the "Correct" feature using the word-sequence set

It can be seen from the above data that: if the user's queries are only word sequences that are neither semantically correct nor syntactically correct, only the word-sequence grammar can recognize some, though the accuracy (14.6%) is much lower than that of the semantic grammar using the semantics set (75.35%) or that of syntactic grammar using syntax set (65%). Neither semantic grammar nor syntactic grammar can recognize any query in word-sequence set.

Table 7.3.2 (8): the "Incorrectness" feature using the word-sequence set

Grammars	Person #2
Semantic	10.4
Syntactic	29.2
Word Sequence	56.3
Ext. Semantic	4.2
Ext. Syntactic	27.1
Ext. Word Sequence	66.7

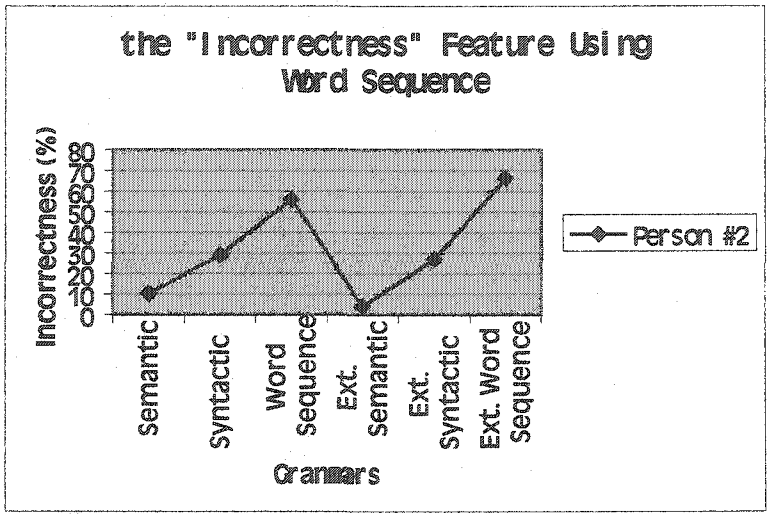

Figure 7.3.2 (8): the "Incorrectness" feature using the word-sequence set

The above data shows that: if the input queries are only word sequences, the trend with respect to the mis-recognition rate is similar to that shown in figure 7.3.2 (4), which represents the "incorrectness" feature using the semantics set. The word-sequence grammar has the highest mis-recognition rate, and the semantic grammar has the lowest mis-recognition rate.

Table 7.3.2 (9): the "Not recognized" feature using the word-sequence set

Grammars	Person #2
Semantic	89.6
Syntactic	70.8
Word Sequence	29.1
Ext. Semantic	95.8
Ext. Syntactic	72.9
Ext. Word Sequence	27

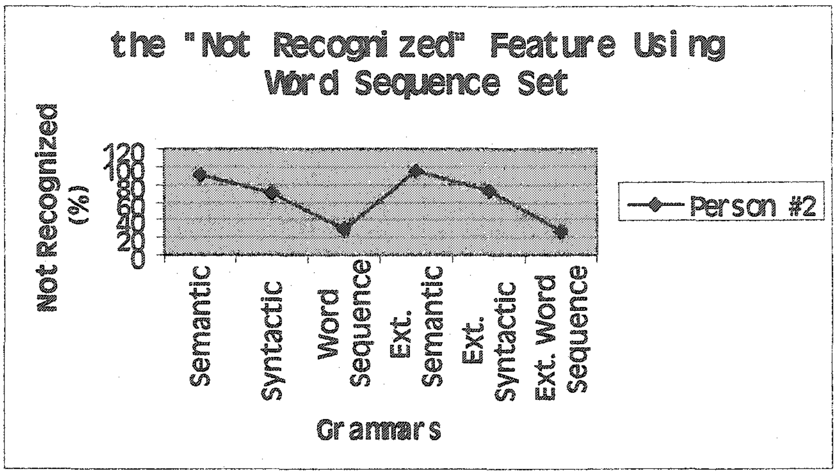

Figure 7.3.2 (9): the "Not recognized" feature using the word-sequence set

The above data shows that: if the user asks only word sequences that are neither semantically correct nor syntactically correct, the semantic grammar has the highest "not recognized" rate, and that for word-sequence grammar is the lowest.

Figures 7.3.2 (7), 7.3.2 (8), and 7.3.2 (9), and tables 7.3.2 (7), 7.3.2 (8), and 7.3.2 (9) show that, if the queries are only word sequences that are neither semantically correct, nor syntactically correct, the experiment results can be summarized as follows:

- Only the word-sequence grammar can recognize some queries. Neither the semantic grammar nor the syntactic grammars can recognize any such kind of queries.
- The word-sequence grammars have the most mis-recognitions, and the semantic grammar has the lowest mis-recognition rate.
- The word-sequence grammars have the lowest percentage for "not recognized" among the three kinds of grammars, and the semantic grammar has the highest "not recognition" rate.

7.4 Detailed Analysis of the Results

The tentative experiment is examined and analyzed from the following aspects:

7.4.1 Review the Nature of the Testing Utterances (Queries)

The grammars in the experiment define the language that can accept users' questions within the domain of a solar system. The testing utterances are customized with the goal of being representatives of the language. However, the language in the experiment is too large to be able to choose a sample size that can be shown to be truly representative from a phonetic perspective. For example, the smallest language in the experiment is defined by the semantic grammar with a language size of $2.70 * 10^{12}$. What we have done is to pick representatives from each type of question in each alternative in the root rule of the grammar, which is subdivided further for subtypes of utterances. While selecting the words in the same category, we apply different words in different testing utterances in order to have a broad testing coverage. In addition, we did not include very long queries for testing utterances, such as "Is a red crater or an atmospheric mountain contained by a planet or a moon", in the experiments in order to avoid speech errors from the person speaking that result from misreading the query. As a matter of fact, we would say that the testing utterances are enough in number rather than in the sense of being provably representative. In a future, more intensive experiment, it might be possible to identify a more 'provably-representative' set of utterances.

7.4.2 Calculation of Language Size

Using the method described in sub-section 6.1, the sizes of the languages defined by the experimental grammars can be calculated precisely, using the method in sub-section 6.2, the average branching factors can be estimated, furthermore, the estimated-average-query lengths are also available using the formula $s=b^d$. The detailed calculation process of language sizes and branching factors can be found in Appendix B and Appendix C respectively, the results are summarized in table 7.4.2.

Table 7.4.2:language sizes and branching factors

Grammar	**Language size (*s*)**	**Branching factor (*b*)**
semantic grammar	$2.70 * 10^{12}$	39.6
syntactic grammar	$3.05 * 10^{15}$	95.5
word-sequence grammar	$2.31 * 10^{24}$	273
extended semantic grammar	$5.55 * 10^{12}$	95.6
extended syntactic grammar	$8.17 * 10^{15}$	267.3
extended word-sequence grammar	$2.40 * 10^{27}$	547

7.4.3 An Analysis of Individual User

Since the default voice in the experimental environment is an American male voice, it is not surprising that the person #1, English male (although not North American), in the experiment has an overall higher recognition accuracy than person #2 who is a non-English female. In addition, person #2 is more likely to be affected by the training, which means that she is being accepted by the system better with more practice and adjustment. That's also the reason that we vary the testing order in the experiments.

Despite the differences between the experiment subjects, they provide the same trend, with only one exception (the "not-recognized" result when using the semantics set of questions), with the same grammar and testing-utterance set in the experiment, which is clearly shown by the figures in sub-section 7.3.2. This fact indicates that the performances of different languages (defined by different grammars) are comparable though various subjects may be involved in the experiment. Furthermore, it proves the generality of the observations set up in this thesis.

7.4.4 An Analysis of the Person-Specific Problem

The VXML browser has the ability to convert text to speech (TTS), but it requires the text be represented in its pronunciation format. For example, "OK" need to be written in "Okay", and "etc." in "et cetera". During the experiment, we observed that the speech-recognition system may not be able to recognize some of the user's specific words. Person #1, for instance, the word "earth" maybe recognized as "paris" or something else, and for person #2, the word "earth" may be acceptable, but the word "Jupiter" may be mis-recognized as "Jupitereighth". To these specific words, we make the modification in all the grammars using "urth" to replace word "earth" that may have the pronunciation "ear th", and using "Jupiter eighth" that are divided into two separate words, instead of the one word "Jupitereighth" in order to avoid the mis-recognition of word "Jupiter".

The experiments do show the effectiveness of these modifications. The figures in sub-section 7.3.2 show the results after we made such changes to all the experiemental grammars. The semantic grammars correct those words successfully, but the problems still exist in the syntactic grammars, which also proves our statement that the semantic grammar has better recognition accuracy than the syntactic grammar. The drawback is that this correction is person-specific.

7.4.5 An Analysis of the "Correctness" Feature

From the figures in sub-section 7.3.2, we can state that: if the user is very clear about the system, and inputs both semantically and syntactically correct utterances (in semantics set) to the speech-recognition system, the semantic grammar provides the best recognition accuracy, the syntactic grammar has the second best accuracy, and the accuracy of word-sequence grammar is the lowest. Meanwhile, in the vertical comparison, the extended grammar has lower recognition accuracy than its original one (before its extension).

However, if the user is not familiar with the speech-recognition system and its current domain, therefore asks out-of-range utterances, the semantic grammar is not superior to the syntactic grammar or word-sequence grammar any more. Refer to figure 7.3.2 (2) and figure 7.3.2 (3), as for only syntactically correct but not semantically correct utterances (syntax set), the syntactic grammar has better recognition accuracy than semantic grammar and word-sequence grammar; only word-sequence grammar, among the three types of grammar, has any recognition ability (i.e., recognize some utterances correctly) to the word-sequence set.

7.4.6 An Analysis of the "Incorrectness" Feature

As discussed in sub-section 6.3.1, mis-recognitions are unwelcome in a speech-recognition system. Refer to figure 7.3.2 (4), in the semantics testing utterance set, the semantic grammars are least likely to have mis-recognitions, which means it performs better than the syntactic grammar and the word-sequence grammar. Meanwhile, the extended grammars have relatively more mis-recognitions than their original ones. These phenomena are in coincidence with the finding that semantic grammars have the best recognition accuracy among these three types of grammar, and the accuracy drops down with the extension of the grammar (discussed in sub-section 7.4.4). But if the user's inputs belong to the syntax testing utterance set (only syntactically correct, but semantically incorrect), the semantic grammar is inferior to the syntactic grammars with respect to the mis-recognition feature. The word-sequence grammar always has the most mis-recognitions with any type of input utterances.

7.4.7 An Analysis of the "Not recognized" Feature

Within the semantics set, the semantic grammar is the one that is most likely to respond the user with "not recognized" information. The percentage of "not recognized" of the extended grammars is usually higher than the original grammars. Since the testing utterances in the syntax set are semantically wrong, the semantic grammar often responds the user with "not recognized" here. The word-sequence grammars seldom respond the "not recognized" information in any testing utterance set.

7.4.8 Examine the Detail of Incorrect Recognition (Mis-recognition)

The tentative experiment has explicitly shown that the overall recognition accuracy of the word-sequence grammar is pretty low. So, does that mean the word-sequence grammar is useless any way? If we take a careful examination of Table Appendix D (8) in Appendix D, we can find that, with proper analysis, the word-sequence grammar is also able to provide some useful information in the speech-recognition system.

The mis-recognitions can be classified into two types by the extent of the incorrectness in the recognition. The first type of mis-recognition is: the system recognizes most of the words (e.g., greater than 70%, this threshold is set up depending on specific system and requirement). For example, the user says "Was phobos discovered by a person", the system does not recognizes the determiner "a", and the recognition result is "Was phobos discovered by person" (number of correct words/ total number=5/6 = 83.3% correctness). In this case, the system has caught the correct meaning of the user's input, the communication between the user and the system can proceed smoothly.

In the second type of mis-recognition, the system may only recognize a small part of the input utterance (e.g. less than 70%). For example, if the user asks "Which mountain is found on Jupiter", the system recognizes as "which mountain is *Yaounde* Jupiter", the correct recognition rate is 4/ 6 (66.7%). Though the recognition result seems funny, we can guess from that the user is interested in "which mountain" and some relation to "Jupiter". In this case, if the system is robust enough, it could further confirm the user's question by prompting the user with "Are you interested in the mountain and Jupiter?" If the user answers "Yes", the system may direct the user to the site with the information of "Jupiter and mountains", and the communication continues correctly.

Furthermore, if we examine some sample recognition results, we may find some "probable" mis-recognitions. Again take a look at the Table Appendix D (8), we find out that the user's input "found on" is likely to be recognized with "*Yaounde*". So, if there are not many utterances about the real "*Yaounde*", we can replace the "*Yaounde*" with "found on" in the recognition results and obtain the more reasonable recognition. Then, the above example mis-recognition "which mountain is *Yaounde* Jupiter" is restored to "which mountain is *found on* Jupiter", which is the correct result corresponding to the user's input.

7.4.9 An Analysis of the "Robustness" Feature

As discussed in sub-section 4.2, robustness means the extent to which a system handles errors or "unexpected" inputs. From the figures in sub-section 7.3.2, we can see that the semantic grammar is most likely to refuse incorrect inputs, since it cannot accept any utterance that is not semantically correct; the next one is syntactic grammar, which can accept the syntax set, but refuse the word-sequence set; while the word-sequence grammar seems to be able to accept any kind of utterances and word sequences. Therefore, the robustness of semantic grammar, syntactic grammar, and word-sequence grammar is increasing in this order.

7.4.10 Issues on Grammar Combination

Since we cannot anticipate 100% accuracy in the state of the art, we have to try to overcome the drawback of non-recognition. For mis-recognition, it seems we cannot do anything to it, because during the communition, the user doesn't realize the speech system has mis-recognized his/ her voice input unless the system returns what it gets after the user's input, which is certainly annoying to the user.

If the system returns the information of "Not recognized", we may have the following two choices to improve it: (1) prompt the user to repeat his/ her utterances. For example, refer to table Appendix D (9), the second utterance ("Is titania a mountain") was not recognized in the first round test (sem #1), but in the next round test (sem #3), it was recognized correctly. (2) Transfer the speech input to a grammar defining a larger language with the same domain. For example, we could combine the semantic grammar, syntactic grammar, and word-sequence grammar into one "combined grammar", and assign them with the different probabilities in the descending order. When the system receives a voice input, the grammar with the highest probability (i.e., semantic grammar) is tried first. If it cannot recognize the input, the input is transferred to the grammar with lower probability (i.e., syntactic grammar), and so on. For example, the user asks an utterance in the syntax set, the semantic grammar definitely refuses it, then the syntactic grammar (with lower probability) could be used, and may accept the input. The results of our experiments suggest that speech-recognition systems which use combined grammars will be able to achieve a flexible combination of good accuracy and good robustness. This part of our hypothesis requires further investation.

7.4.11 An Analysis of the Results for Design Issues

Through the above analysis, the advantages and disadvantages of each grammar (semantic grammar, syntactic grammar, and word-sequence grammar) are summarized as follows:

- The semantic grammar has the best recognition accuracy for semantically and syntactically correct utterances, but lowest robustness for other types of utterances. It defines the smallest language, with the most complicated grammar design that is most difficult to maintain and extend;
- The syntactic grammar has median performances in accuracy, robustness, language size, and grammar complicity, among these three types of grammar;
- The recognition accuracy of the word-sequence grammar is very low, but it is the most robust grammar, and may provide some useful information when the user inputs an 'unexpected' utterance. The grammar of word sequence is the simplest one, which covers the largest language.
- If these three grammars are integrated into one combined grammar, using probability values, the speech-recognition system may achieve flexible combination of accuracy and robustness.

So, what kind of grammar should be applied in a specific speech-recognition system which can only use one grammar? If the system requires high recognition accuracy, the semantic grammar should be the first selection; otherwise, if the system emphasizes more robustness than accuracy, the syntactic grammar could be considered. The word-sequence grammar as the most robust grammar may be useful in some specific application. To balance the robustness and accuracy, we suggest integrating these three grammars, and assigning them different probability values.

Furthermore, the language size defined by the grammar in the speech-recognition system needs to be considered. Refer to sub-section 7.4.2, the smallest language in the experiment has the size of $2.70 * 10^{12}$. So what has been proven in the experiment may be applicable to grammars that define a language size less than $2.70 * 10^{12}$. To better imagine how large the language is, figure 7.4.11 (1) and figure 7.4.11 (2) show a sample grammar and a language in tree structure.

<Sent>^{360000} = <Quest>^{3} <Det>^{2}<Noun>^{100} <Verb>^{3} <Det>^{2} <Noun>^{100};
//3*100*3*2*100=1.8 * 10^5
<Quest>^{3} = was | does | did;
<Det>^{2} = a | an;
<Noun>^{100} = planet | moon | mountain | gas | chemical | earth | mars |... // 100 words
<Verb>^{3} = find | found | contain;

Figure 7.4.11 (1): sample grammar showing language size

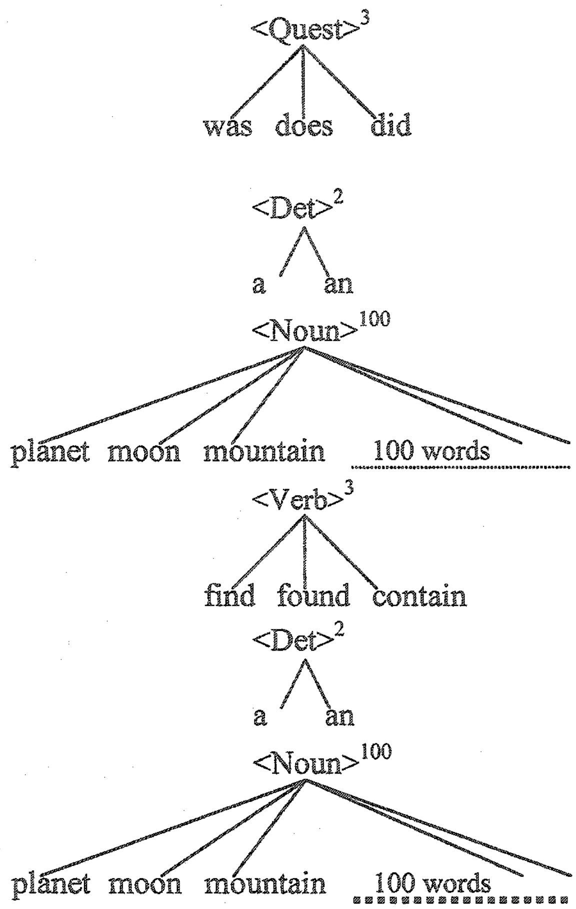

Figure 7.4.11 (2): Tree structure of the language defined by the sample grammar

It is shown that the smallest language in our experiment is almost amillion times larger than the above sample language. It is reasonable to believe that the results of the experiment identify a not-worse performance in speech-recognition systems, such as command systems, covering a small vocabulary of commands, such as "open the door" and "turn on the light", but the experiment cannot guarantee the same accuracy for larger systems such as a university-management system with a larger vocabulary, for example, thousands of students.

Chapter 8

Conclusion

8.1 Summary of Work Done

To investigate the features of grammars in speech recognition, experiments were carried out and results were analyzed. Based on the grammars created by Frost (2002), we constructed three types of grammar, semantic grammar, syntactic grammar, and word-sequence grammar, as well as their extensions. The examination of different grammars in speech recognition is conducted from two directions: horizontal and vertical comparisons. Semantic grammar, syntactic grammar, and word-sequence grammar are compared in horizontal direction. The original (unextended) grammar is compared with its extension (e.g. syntactic grammar vs. extended syntactic grammar) for vertical comparison. Two subjects (people) are involved in this experiment, an English male and a non-English female. Three customized testing-utterance sets are included in the experiments: one is the semantics set, in which the utterances are both semantically and syntactically correct; second set is the syntax set, which covers the utterances that are only syntactically correct but semantically incorrect; the third set is the word-sequence set, which includes only word sequences that are neither semantically nor syntactically correct.

The experiments indicate that: in the same domain, the smaller the size of the defined language, the higher is the recognition accuracy, but the less is the robustness. Meanwhile, the experiments show the unexpected result that the semantic grammar is less likely to lead to mis-recognition than the syntactic grammar and word-sequence grammar with the utterances that are both semantically and syntactically correct. If the utterances are only syntactically correct and semantically incorrect, the syntactic grammar outperforms the semantic grammar. In addition, the word-sequence grammar would be useful with 'unexpected' utterance inputs.

Furthermore, the experimental results suggest that the integration of semantics, syntax, and word-sequence grammar, using probability values, into speech-recognition grammar, would achieve a flexible combination of robustness and accuracy.

There are many factors involved in grammar design and speech-recognition issues. Though what we have shown are very crude experiments, they are sufficient to indicate that grammar design in speech-recognition technology is a good area for further study.

8.2 Limitations of the Experiment

Although there is no contradictive evidence in the experiment to the thesis statement, it is really a quite crude and preliminary experiment. There are a number of limitations that need further improvement.

- **Insufficient knowledge of the recognition mechanism used in the VXML tool**

During the experiment, we did not study the recognition mechanism of the experiment tool. We don't know the threshold of a phoneme to be accepted (recognized) by the system. We don't know whether the speech speed will influence the speech recognition. We have no idea whether large spaces between words will be helpful or hindering to speech recognition. We also don't know whether the loudness level will affect the recognition result, and to what extent background noise will affect the recognition.

- **Representative utterances**

In sub-section 7.4.1, we look into the testing utterances, and have to accept that it is hard to say that these utterances are the exact representatives of the language. For example, we did not include very long testing utterances in the experiments, such as "Is a red crater or an atmospheric mountain contained by a planet or a moon", in order to avoid speech errors that result from the person misreading the query.

- **Subject-specific problems**

As analyzed in sub-section 7.4.4, there are subject-specific problems in our experiment which may affect the generic application of the speech-recognition system. If more subjects (people) are involved in the experiments, someone may have some problems that are all right to others, but the others may have other new problems. So, it will be very hard to handle the subject-specific problems in generic meaning.

- **Crude experiment**

This experiment is only a crude and preliminary experiment. Only two people (subjects) and three types of grammar: semantic grammar, syntactic grammar, and word-sequence grammars, are involved. In the future, more people (subjects) and more experiments will be involved.

8.3 Future Work

As we have seen through this thesis report, there is a lot to be desired in the speech-recognition technology. Moreover, every step along the long road is open to philosophical debate. As much as we understand that a 100% accuracy and robustness in speech-recognition is impractical, we do respect the sufficiency of any trivial observation and improvement. Since the extensive investigation shows the significance of grammar design in speech-recognition technology, it deserves further attention in the future work.

Besides the above limitations discussed in sub-section 8.2, we will consider another critical issue existing in speech-recognition technology: how to balance the recognition accuracy and robustness. In future work, we would combine the semantics, syntax, and word sequence, using probability values, into speech-recognition grammar to achieve a flexible combination of accuracy and robustness.

Furthermore, we also expect to construct a speech-recognition system with good flexibility in the future. As we showed in sub-section 4.3, flexibility is one of the problems in speech-recognition technology. The ideal system is able to accept the user's out-of-range utterances, and provide him/her with some reasonable guidance to direct him/her to the correct place to continue using the system. We'd like to set the flexible navigation through a speech-web as our future work.

8.4 Summary of Conclusions

Over the past decades, a lot of work has been carried out on speech-recognition technology, a variety of approaches have been proposed, and numerous commercial or laboratorial speech-related products have emerged. However, there are a number of unsolved problems in speech-recognition technology. In this thesis report, we have investigated the significance of grammar design in natural-language speech-recognition.

Supported by the experiments, we conclude as follows:

- Adding syntactic rules does improve recognition accuracy.
- Adding semantic constraints further improves accuracy.
- All of the grammars have advantages and disadvantages, so the application characteristics need to be carefully examined to select the proper grammar. Table 8.4 summarizes the relation between the application characteristics and grammars.

Table 8.4: application characteristics and grammars

Application Characteristics	**Grammars**	**Accuracy (%)**	**Incorrectness (%)**
High accuracy for semantic	✓ *Semantic grammar*	*High (75)*	*Low (4)*
	Syntactic grammar	Median (66)	M (14)

	Word-sequence grammar	Low (12)	High (60)
High robustness for syntactic queries	Semantic grammar	Low (0)	Median (22)
	✓ *Syntactic grammar*	*High (65)*	*Low (5)*
	Word-sequence grammar	Median (8)	High (44)
Highest accuracy for word sequences	Semantic grammar	Low (0)	Low (10)
	Syntactic grammar	Low (0)	M (30)
	✓ *Word-sequence*	*Median (15)*	*High (56)*

Table 8.4: application characteristics and grammars (Cont'd)

Table 8.4 shows that if the application requires high recognition accuracy for semantic queries, the semantic grammar should be the best choice with highest recognition accuracy and lowest mis-recognition rate; if the application asks for high robustness with syntactic queries, the syntactic grammar should be the candidate; if the application need highest robustness for word sequences, the word-sequence grammar could be considered.

- If the grammar could be combined, using probability values, it would result in a flexible combination between accuracy and robustness.

Bibliography

Note: This bibliography is for both the thesis report and the survey given in appendix A.

1. Ahlrichs, U., Fischer, J., Denzler, J., Drexler, C., Niemann, H., Noth, E., and Paulus, D. (1999) Knowledge Based Image and Speech Analysis for Service Robots. In *Proceedings of the Conference on Integration of Speech and Image Understanding.* Korfu, Greek, 1999, 21-47.
2. Amaya, F., Benedf, J. M. and Sãnchez, J.A. (1999) Learning of Stochastic Context-Free Grammars From Bracketed Corpora By Means of Reestimation Algorithms. *VIII Simposium on Pattern Recognition and Image Analysis*. Vol. 1, Bilbao, 1999, 119-126.
3. Atwell, E., Arnfield, S., Demetrious, G., Hanlon, S., Hughes, J., Jost, U., Pocock, R., Souter, C., and Ueberia, J. (1993) Multi-Level Disambiguation Grammar Inferred From English Corpus, Treebank, and Dictionary. In *[Lucas 93]: 81-98.*
4. Atwell, E., and Kevitt, P. M. (1993) Pragmatic Linguistic Constraint Models for Large-Vocabulary Speech Processing. *Knowledge at Work in Universities: Proceedings of the second annual conference of the Higher Education Funding Councils' Knowledge Based Systems Initialtive.* Leeds University Press, 1993, 147.
5. Baggia, P., Kellner, A., Perennou, G., Popovici, C., and Sturm, J. (1999) Language Modelling and Spoken Dialogue Systems the ARISE Experience. In *Proceedings of the Sixth European Conference on Speech Communication and Technology*, Budapest, Hungary, September 1999, 1459-1462.
6. Ballim, A. and Pallotta, V. (2000) Semantic Filtering by Inference on Domain Knowledge in Spoken Language Dialogue Systems. In *Proceedings of From spoken dialogue to full natural interactive dialogue. Theory, empirical analysis and evaluation LREC 2000 Workshop,* May 29th, 2000 Athen, Greece.
7. Bellegarda, J.R. (1998) Exploiting Both Local and Global Constraints for Multi-Span Statistical Language Modelling. In *Acoustics, Speech and Signal Processing, 1998. proceedings of the 1998 IEEE International Conference on.* vol.2, May 1998, Seattle, WA, USA, 677-680.

8. Benedi, J. and Sanchez, J. (2000) Combination of N-grams and Stochastic Context-Free Grammars. In *Proc. of International Conference on Computational Linguists, COLING 2000*, Saarbrücken, July 31st to Aug 4th, 2000, 196-206.

9. Black, E., Jelinek, F., Lafferly, J., Magerman, D. M., Mereer, R. and Roukos, S. (1992) Towards History-Based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the February 1992 DARPA Speech and Natural Language Workshop*. Arden House, NY.

10. Blackburn, P. and Striegnitz, K. (2002) http://www.coli.uni-sb.de/~kris/nlp-with-prolog/html/

11. Bonafonte, A., Marino, J.B., and Nogueiras, A. (1996) Sethos: The Upc Speech Understanding System, in *Proceedings (ICSLP) '96*, vol. 4, Philadelphia, PA, October 1996

12. Bunt, H. C. (1995) Semantics and Pragmatics in the DELTA System. *In: L. Dybkjaer (ed.)* Topics. *Proc. of the Second Dialogue and Discourse Workshop*, Dublin, April 1995. CCI, Roskilde University

13. Buo, F. D. and Waibel, A. (1996) FeasPar - A Feature Structure Parser Learning to Parse Spoken Language. In *Proceedings. COLING-96*, Copenhagen, Denmark 1996.

14. Carbonell, J.G. and Hayes, P.J. (1984) Recovery Strategies for Parsing Extra-grammatical Language. In *Technical Report CMU-CS-84-107*, Carnegie Mellon University, Pittsburgh, PA, 1984.

15. Casacuberta, F., Llorens, D., Martinez, C., Molau, S., Nevado, F., Ney, H., Pastor, M., Pico, D., Sanchis, A., Vidal, E., and Vilar, J. M. (2001) Speech-To-Speech Translation Based on Finite-State Transducers. In *Proceedings of http://citeseer.nj.nec.com/casacuberta01speechtospeech.html*.

16. Chappelier, J., Rajman, M., Aragues, R. and Rozenknop, A. (1999) Lattice Parsing for Speech Recognition. In *Proceedings of 6me conference sur le Traitement Automatique du Language Naturel (TALN'99),* July 1999, 95-104.

17. Chelba, C.L. (2000) Exploiting Syntactic Structure for Natural Language Modelling. *The Johns Hopkins University Ph.D's Dissertation, 2000.*

18. Chelba, C., Engle, D., Jelinek, F., Jimenaz, V., Khudanpur, S., Mangu, L, Printz, H., Ristad, E., Rosenfeld, R., Stolcke, A., and Wu, D (1997) Structure and Performance of A Dependency Language Model. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 1997, volume 5, 2775-2778.

19. Chung, G. and Seneff, S. (1998) Improvements in Speech Understanding Accuracy Through the Integration of Hierarchical Linguistic, Prosodic, and Phonological Constraints in the Jupiter Domain. *Proc. ICSLP 98, Sydney,* Australia, December 1998, 935-939.

20. Churcher, G.E., Souter, C. and Atwell, E.S. (1996) Dialogues In Air Traffic Control. In *Proceedings of 11th Twente Workshop on Language Technology*, Twente, Netherlands.

21. Coccaro, N., and Jurafsky, D. (1998) Towards Better Integration of Semantic Predictors in Statistical Language Modeling. In *Proceedings of ICSLP '98*, Sydney, Australia, November 1998, 6.2403-2406.

22. Collins, M. (1996) A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th annual meeting of the Association for Computational Linguistics*, May 1996, 184-191.

23. Demetriou, G.C, Atwell, E., and S7outer, C. (2000) Using Lexical Semantic Knowledge From Machine Readable Dictionaries for Domain Independent Language Modelling. In *LREC 2000 2nd International Conference on Language Resources & Evaluation.*

24. Demetriou, G.C. and Atwell, E.S. (1994a) Semantics in Speech Recognition and Understanding: A Survey. In *Evett, L. & Rose, T. (editors) Computational Linguistics for Speech and Handwriting Recognition. AISB'94 Workshop*. University of Leeds/AISB.

25. Demetriou, G.C. and Atwell, E.S. (1994b) A Large Vocabulary Semantic Network for Computerised Speech Recognition. In *Evett, L & Rose,T (editors) Computational Linguistics for Speech and Handwriting Recognition AISB'94 Workshop*, University of Leeds/AISB. 1994, 21-28.

26. Dorre, J. (1997) Efficient Construction of Underspecified Semantics Under Massive Ambiguity. In *meeting of the Association for Computational Linguistics*, 386-393.

27. Dowding, J., Bratt, E.O. and Goldwater, S. (1999) Interpreting Language in Context in CommandTalk. In *Communicative Agents: The Use of Natural Language in Embodied Systems*, 63-67.

28. Dowding, J., Moore, R., Andry, F., and Moran, D. (1994) Interleaving Syntax and Semantics in an Efficient Bottom-Up Parser. In *Proceedings of the 32nd Annual meeting of the Association for Computational Linguistics*, New Mexico State University, las Cruces, New Mexico, 27 June – 1 July, 1994, 110-116.

29. Dowding, J., Gawron, J. M., Appelt, D., Bear, J., Cherny, L., Moore, R., and Moran, D. (1993) GEMINI: A Natural Language System For Spoken-Language Understanding. In *Proceedings of the ARPA Workshop on Human Language Technology,* 1993.

30. Dupont, P. (1993) Dynamic Use of Syntactical Knowledge in Continuous Speech Recognition. In *Proceedings of Eurospeech 1993*, 1959-1962.

31. Edmiston, R.D. (1982) The Puff Speech Recognition System. *Ph.D's dissertation.* The University of Texas at Austin, 1982.

32. Fink, P. (1984) The Acquisition and Use of Dialogue Expectation in Speech Recognition (Natural Language Processing, Artificial Intelligence). *DAI-B 45/01*, Jul 1984. Computer Science, Duke University, 263.

33. Fischer, J., Noth, E. and Niemann, H. (1999) Combining Statistics with Semantic Networks in A Real-Time Dialogue System. In *Proceedings of International Workshop Speech and Computer (SPECOM '99)*. Moskau, Russia, 1999, 75-78.

34. Frost, R.A. (2002) Improving Speech-Recognition Accuracy by Coding Semantic Constraints in the Syntax of the Recognition Grammar. *Technical Note.* School of Computer Science, University of Windsor.

35. Frost, R.A and Chitte, S. (1999) A New Approach For Providing Natural-Language Speech Access to Large Knowledge Bases. *Proceedings of the Pacific Association of Computational Linguistics Conference PACLING '99*, University of Waterloo, August 1999, 82-89.

36. Geistert, B. (1998) Learning Unification-Based Phrase-Structure-Rules with a Grammar Inference Tool. In *Proceedings of the European Summer School in Logic,*

Language and Information (ESSLLI-98) Workshop on Automated Acquisition of Syntax and Parsing, Saarbrücken, 1998.

37. Geutner, P. (1996) Introducing Linguistic Constraints Into Statistical Language Modeling. In *Spoken Language, 1996. ICSLP '96*. Proceeding Fourth International Conference on. vol. 1, 3-6 Oct. 1996, Philadelphia, PA, USA, 402-405.
38. Glass, J.R. (1999) Challenges for Spoken Dialogue Systems. In *Proceedings of the 1999 IEEE ASRU Workshop*. http://www.sls.lcs.mit.edu
39. Goddeau, D.M. (1993) An LR Parser-Based Probabilistic language Model for Spoken language Systems (LR Parser). *DAI-B 54/03*, Sep 1993, Massachusetts Institute of Technology, 1494.
40. Goldwater, S. J., Bratt, E. O., Gawron, J. M., and Dowding, J. (2000) Building a Robust Dialogue System with Limited Data. In *Workshop on Conversational Systems*, Somerset, New Jersey, 2000, Publisher: Association for Computational Linguistics, 61-65.
41. Good, I.J. (1995) The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40 (3 and 4): 237-264.
42. Haas, J., Hornegger, J, Noth, E. and Niemann, H. (1998) A Probabilistic Approach for the Semantic Analysis. In *Proceedings of SQEL '98 Workshop on Multi-Lingual Information Retrieval Dialogs, Stara lesna (Slovakia), 1998*, 422-430.
43. Haas, J., Hornegger, J. Huber, R., and Niemann, H. (1997) Probabilistic Semantic Analysis of Speech. In E. Paulus and F.M. Wahl, editors, *Mustererkennung 1997, DAGM Symposium*. 270-277.
44. Halber, A. (1998) Grammatical Factor and Spoken Sentence recognition. In *Conference on Text Speech and Dialogue TSD'98*, 1998.
45. Han, B. (2000) Improving Spoken Dialog Systems, *Language Technology Institue, Carnegie Mellon University*, December 14, 2000.
46. Harper, M., White, C., Wang, W. and Johnson, M. (2000) The Effectiveness of Corpus-Induced Dependency Grammars for Post-processing Speech. In *Proceedings of the First Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, April 2000.

47. Harper, M. P., Hockema, S. A., and White, C. M. (1999a) Enhanced Constraint Dependency Grammar Parsers. In *Proceedings of the IASTED International Conference for Artificial Intelligence and Soft Computing*, Honolulu, HI, August 1999, 294-128

48. Harper, M.P., Johnson, M.T., Jamieson, L.H., Hockema, S.A., and White, C.M. (1999b) Interfacing a CDG Parser with An HMM Word Recognize Word Graphs. In *Acoustics, Speech, and Signal Processing, 1999 Proceedings, 1999 IEEE International Conference on*, vol. 2, 15-19 March 1999, Phoenix, AZ, USA, 733-736.

49. Harper, M. P., Helzerman, A. (1995) Extensions To Constraint Dependency Parsing For Spoken Language Processing. *Computer Speech and Language*, 9:187-234.

50. Harper, M.P., Jamieson, L.H., Mitchell, C.D., Ying, G., Potisuk, S., Srinivasan, P.N., Chen, R., Zoltowski, C.B., McPheters, L.L., Pellom, B. and Helzerman, R.A. (1994) Integrating Language Models with Speech Recognition. In *the Proceedings of the AAAI-94 Workshop on Integration of Natural Language and Speech Processing,* Seattle, Washington, July 1994.

51. Harper, M.P., Jamieson, L.H., Zoltowski, C.B., and Helzerman, R.A. (1992) Semantics and Constraint Parsing of Word Graphs. In *Proceedings of the International Conference on Acoustics, Speech, and signal processing,* Volume II, April 1992, 63-66.

52. Hauptmann, A.G., Jones, R.E., Seymore, K., Slattery, S.T., and Witbrock, M.J. (1998) Experiments in Information Retrieval From Spoken Documents. In *Proceedings of the Broadcast News Transcription and Understanding Workshop,* 175-181.

53. Hazen, T. J., Burianek, T., Polifroni, J. andSeneff, S. (2000) Integrating Recognition Confidence Scoring with Language Understanding and Dialogue Modeling. In *theses Proceedings*.

54. Helzerman, R. A. and Harper, M. P. (1996) MUSE CSP: An Extension To The Constraint Satisfaction Problem. *Journal of Artificial Intelligence Research*, 5: 239-288.

55. Hermannsdottir, S.H. (1996) A Dialogue Manager For A Spoken Dialogue System. *McGill University (Canada) MSC thesis, MAI 34/04*, Aug 1996, 1616.

56. Huang, X., Alleva, F., Hon, H., Hwang, M., and Rosenfeld, R. (1992) The SPHINX-II Speech Recognition System: An Overview. *Computer Speech and Language,* vol. 7(2), 137-148.

57. Hunt, A.J. (1994) Improving Speech understanding Through Integration of Prosody And Syntax. In *Proceedings of the Artificial Intelligence Conference, Australia.* Armidale, Australia, 442-449.

58. IBM (2002) http://www-3.ibm.com/software/pervasive/products/voice/voice_server_sdk.shtml.

59. IBM (2001) *WebSphere Voice Server Software Developers Kit (SDK) Programmer's Guide*, Version 3.11.

60. Jelinek, F. and Mercer, R.L (1980) Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, the Netherlands: North-Holland, May 1980, 381-397.

61. Johnson, M.T. (2001) Incorporating Prosodic Information and Language Structure into Speech Recognition Systems. *Purdue University Ph. D's Dissertation, DAI-B 62/06*, Dec 2001.

62. Jones, G.J.F., Lloyd-Thomas, H., and Wright, J.H. (1993) Adaptive Statistical And Grammar Models of Language Application to Speech recognition. In *Grammatical Inference: Theory, Applications Alternatives, IEEE Colloquium On*, Colchester, UK, 1993, 25/1-25/8.

63. Jurafsky, D., Wooters, C., Segal, J., Stolcke, A., Fosler, E., Tajchman, G., and Morgan, N. (1995) Using a Stochastic Context-Free Grammar As A Language Model For Speech Recognition. In *Proceedings of ICASSP '95*. Detroit, MI, 1995, 189-192.

64. Kaiser, E. C. (1999) Robust, Finite-State Parsing for Spoken Language Understanding. *In Student Session of ACL '99*, June 1999.

65. Kaiser, E.C., Johnston, M., and Heeman, P.A (1999) PROFER: Predictive, Robust Finite-State Parsing For Spoken Language. In *Proceedings of ICASSP '99*. Phoenix, Arizona, March 1999.

66. Kamm, C.A., Litman, D.J., and Walker, M.A (1998) From Novice To Expert: The Effect Of Tutorials On User Expertise With Spoken Dialogue Systems. In *Proceedings of 5th International Conf. On Spoken Language Processing*, 1211-1214.

67. Kawahara, T., Araki, M, and Doshita, S. (1994) Heuristic Search Integrating Syntactic, Semantic and Dialog-Level Constraints. In *1994 IEEE International Conference on Acoustics, Speech and Signal Processing, II-25-28.*

68. Katz, S.M (1987) Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. In *Proceedings of IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1987, 35 (3):400-401.

69. Kita, K. and Ward, W. (1991) Incorporating LR Parsing Into SPHINX. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '91)*, 269-272.

70. Kneser, R. and Ney, H (1995) Improved Backing-off for M-gram Language Modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Volume I, Detroit, Michigan, May 1995, 181-184.

71. Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R. and Lewin, I. (2001) Comparing Grammar-Based and Robust Approaches to Speech Understanding: A Case Study. In *Eurospeech 2001.*

72. Lavie, L. (1996) GLR*: A Robust Grammar-Focused Parser for Spontaneously Spoken language. *Carnegie Mellon University Ph.D.'s dissertation.* 1996.

73. Lee, K. (1989) Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System. *Proc. of ICASSP-88,* (1988-4) 123-126.

74. Lewin, I., Becket, R., Boye, J., Carter, D., Rayner, M, and Wiren, M. (1999) Language Processing For Spoken Dialogue Systems: Is Shallow Parsing Enough? In *Accessing Information in Spoken Audio: 23 Proceedings of ESCA ETRW Workshop*, Cambridge, 19&20th April 1999, 37-42.

75. Litman, D.J. and Pan, S. (2000) Predicting and Adapting to Poor Speech Recognition in a Spoken Dialogue System. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, August 2000, 722-728.

76. Litman, D.J. and Pan, S. (1999) Empirically Evaluating an Adaptable Spoken Dialogue System. In *Judy Kay, editor, User Modeling: Proceedings of the Seventh International Conference, UM99*, Springer, Vienna, New York, 1999. 55--64.

77. Loken-Kim, K. (1988) Error Detection And Correction In a Speech Recognition System: A Knowledge Based System Approach. *North Carolina State University Ph.D's Dissertation, 1988.*

78. Lucas,B., Walker, W., and Hunt, A. (1999) ECMAScript Action Tags for JSGF, *Proposal for discussion*, Sep 2, 1999.

79. Magerman, D.M. and Marcus, M.P. (1990) Parsing a Natural Language Using Mutual Information Statistics. In *Proceedings, Eight National Conference on Artificial Intelligence,* August 1990, 984--989.

80. Mahajan, M. Beeferman, D., and Huang, X.D. (1999) Improved Topic-Dependent language Modeling Using Information Retrieval Techniques. In *Acoustics, Speech, and Signal Processing, 1999 Proceedings, 1999 IEEE International Conference on*, vol. 1, 15-19 March 1999, Phoenix, AZ, USA, 541-544.

81. McCandless, M.K. and Glass, J. (1994) Empirical Acquisition of Language Models for Speech Recognition. In *Proceedings of ICSLP '94*, Yokohama, Japan.

82. McTear, M.F. (2002) Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Press New York, NY, USA,* Periodical-Issue-Article, 2002, ISSN: 0360-0300, 90-169.

83. Menzel, W. (1996) Robust Processing of Natural Language. In *Proceedings of 19the German Conference on Artificial Intelligence (KI'95)*, Bielefeld, Germany, 1995.

84. Miller, L.G. and Levinson, S.E. (1988) Syntactic Analysis for Large Vocabulary Speech Recognition Using a Context-Free Covering Grammar. In *Acoustics, Speech, and Signal processing, 1988, ICASSP-88., 1988 International Conference on*, vol. 1, 11-14 April 1988, New York, NY, USA, 271-274.

85. Milward, D., SRI International (1999) Towards a Robust Semantics for Dialogue using Flat Structures, in *Proceedings of Amsteogue '99, Workshop on the Semantics and Pragmatics of Dialoge,* Amsterdam University, 7-9 May 1999, Part II.

86. Milward, D. and Knight, S. (2001) Improving On Phrase Spotting For Spoken Dialogue Processing. *In Proceedings of the Workshop on Innovation in Speech*

Processing, WISP 2001. Institute of Acoustics, 2-3 April 2001, Stratford-upon-Avon, UK.

87. Milward, D. (2000) Distributing Representation for Robust Interpretation of Dialogue Utterances. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics, ACL-2000*, Hong Kong, 2000, 133-141.
88. Moody, T.S. (1988) The Effects of Restricted Vocabulary Size On Voice Interactive Discourse Structure. *North Carolina State University Ph.D's Dissertation, DAI-B 49/06*, Dec 1988, 2333.
89. Moore, R.C. (1999) Using Natural-Language Knowledge Sources in Speech Recognition, *Research Institute for Advanced Computer Science NASA Ames Research Center,* Moffett Field, CA 94035.
90. Moore, R.C., Dowding, J., Bratt, H., Gawron, J., Gorfu, Y. and Cheyer, A. (1997) CommandTalk: A Spoken-Language Interface for Battlefield Simulations. In *Proceedings of the Fifth Conference on Applied Natural Language processing*, 1-7.
91. Moore, R.C., Appelt,D., Dowding, J., Gawron, J.M., and Moran, D. (1995) Combining Linguistic and Statistical Knowledge Sources in Natural-Language Processing for ATIS. In *Spoken Language Systems Technology Workshop*, Austin, Texas, February 1995. Morgan Kaufmann Publishers, Inc., 261-264.
92. Moore, R.C. (1995) Integration of Speech With Natural-Language Understanding. In *Voice Communication Between Humans and Machines*, Roe, D. and Wilpon, J. Eds. Natuional Academy Press, Washington, DC, 254-271.
93. Muller, J. and Stahl, H. (1998) Speech Understanding and Speech Translation in Various Domains by Maximum A-Posteriori semantic Decoding. In *Proceedings of EIS 98*, La Laguna, Spain, 1998, 256-267.
94. Murveit, H. and Moore, R. (1990) Integrating Natural Language Constraints Into HMM-Based Speech Recognition. In *Acoustics, Speech, and Signal Processing, 1990, ICASSP-90., 1990 International Conference on.* Pages: 573-576, vol. 1, April 1990, Albuqerque, NM, USA, 3-6.
95. Ney, H. Essen, U., andKneser, R (1994) On Structuring Probabilistic dependences in Stochastic Language Modeling. In *Computer Speech and Language*, 8:1-38, 1994.

96. Noord, G.V., Bouma, G., Koeling, R., and Nederhof, M.J. (1999) Robust Grammatical Analysis for Spoken Dialogue Systems. In *Journal of Natural Language Engineering, 1999*. 5(1):45—93.

97. Noth, E., Mori, R., Fischer, J., Gebhard, A., Harbeck, S., Kompe, R. Kuhn, R., Niemann, H., and mast, M. (1996) An Integrated Model Of Acoustics And Language Using Semantic Classification Trees. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 1, Atlanta, 1996, 419-422.

98. Rabiner, L.R., Juang, B.H. and Lee, C.H. (1996) An Overview of Automatic Speech Recognition. In C.H. LEE, F.K. SOONG & K.K. PALLIWAL, Eds., *Automatic Speech and Speaker Recognition*. Kluwer.

99. Rayner, M., Hockey, B., James, F., Bratt, H., Bratt, E. O., Gawron, M., Goldwater, S., Dowding, J. and Bhagat, A. (2000a) Corpus-Based Optimization of Language Models Derived from Unification Grammars. *Technical Report TR08-00, RIACS, USRA*.

100. Rayner, M., Hockey, B.A., James, F., Bratt, E.O., Goldwater, S., and Gawron, J.M. (2000b) Compiling Language Models From A Linguistically Motivated Unification Grammar. In *COLING 2000*.

101. Rayner, M., Carter, D., Bouillon, P., Digalakis, V. and Wiren, M. (2000c) The Spoken Language Translator. *Cambridge University Press*, 2000.

102. Rayner, M. and Carter, D. (1997) Hybrid Language Processing In The Spoken Language Translator. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97)*, 107--110.

103. Rayner, M., Carter, D., Digalakis V., and Price, P. (1994) Combining Knowledge Sources To Reorder N-Best Speech Hypothesis Lists. In *Proceedings of the 1994 ARPA Workshop on Human Language Technology, Princeton*. 212-217.

104. Reynar, J.C. (1998) Topic Segmentation: Algorithms and Applications. *University of Pennsylvania Ph.D Dissertation, 1998. DAI-B 59/04*, Oct 1998, 1741.

105. Riccardi, G. and Gorin, A. (1998) Stochastic Language Models For Speech Recognition And Understanding. In *Proceedings in the Fifth International Conference on Spoken Language Proceeding (ICSLP'98)*. Sidney.

106. Riccardi, G., and Bangalore, S. (1996) Automatic Acquisition of Phrase Grammars for Stochastic Language Modeling. *Computer Speech and Language*, 10, 1996, 265-293.

107. Ringger, E.K. (2000) Correcting Speech Recognition Errors. *The University of Rochester Ph.D. Dissertation, 2000.*

108. Ringger, E.K. and Allen, J.F. (1996) Error Correction Via a Post-Processor for Continuous Speech Recognition. In *Proceedings of IEEE International Conference on Acoustics on Acoustics, Speech, and Signal Processing (ICASSP'96)*, May 7-10 1996, Atlanta, GA.

109. Ringger, E.K. and Allen, J.F. (1997) Robust Error Correction of Continuous Speech Recognition. In *proceedings of ESCA-NATO Robust Workshop '97.*

110. Rosenfeld. R. (2000a) Two Decades Of Statistical language Modeling: Where Do We Go From Here? In *Proceedings of the IEEE*, 2000, vol. 88, 1270-1278.

111. Rosenfeld, R. (2000b) Incorporating Linguistic Structure into Statistical Language Models. In *Philosophical Transactions of the Royal Society of London A, 358,* 2000, 1311-1324.

112. Savage-Carmona, J., Holden, A. and Billinghurst, M. (1995) A Hybrid System with Symbolic AI and Statistical Methods for Speech Recognition. *In Manuela Veloso and Agnar Aamodt (Eds.) Proceedings of International Conference on Case-Based Reasoning, ICCBR 95,* October 23-26, 1995, Sesimbra, Portugal, Berlin: Springer Verlag, 1-13.

113. Seide, F., Rueber, B., and Kellner, A. (1996) Improving Speech Understanding By Incorporating Database Constraints And Dialogue History. In *Proceedings of ICSLP'96*, volume 2, Philadelphia, PA, 1996, 1017-1020.

114. Seneff, S., McCandless, M., and Zue, V. (1995) Integrating Natural Language Into the Word Graph Search For Simultaneous Speech Recognition And Understanding. In *4th European Conference on Speech Communication and Technology (Eurospeech'95)*, 1781-1784.

115. Seneff, S. (1992) Robust Parsing for Spoken Language Systems. *In Proceedings of ICASSP*, 189-192.

116. Siu, M., and Ostendorf, M. (2000) Integrating A Context-Dependent Phrase Grammar In The Variable N-Gram Framework. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Istanbul, Turkey, 5-9 Jun, 3: 1643-1646.

117. Soltau, H, and Waibel, A (1998) On The Influence of Hyperarticulated Speech On Recognition Performance. In *Proceedings of ICSLP '98*, Sydney, Australia, 1998, paper 736, 225-228.

118. Souto, N., Meinedo, H., and Neto, J.P. (2002) Building Language Models for Continuous Speech Recognition Systems. I*n Proceedings PORTAL 2002*, Faro, Portugal, 2002, 101-110.

119. Stahl, H., Muller, J., and Lang, M. (1997) Controlling Limited-Domain Applications by Probabilistic Semantic Decoding of Natural Speech. In *Proceedings of ICASSP-97*. Vol. 2, München, Germany, 1163-1166.

120. Stahl, H., Muller, J. and Lang, M. (1996) An Efficient Top-Down Parsing Algorithm For Understanding Speech By Using Stochastic Syntactic And Semantic Models. In *Proc. ICASSP-96,* Atlanta, Georgia, USA, 397-400.

121. Stent, A., Dowding, J., Gawron, J., Bratt, E. and Moore, R. (1999) The CommandTalk Spoken Dialogue System. In *Proceedings of the Thirty-Seventh Annual meeting of the Association for Computational Linguistics*, 183-190.

122. Stolcke, A., Konig, Y., and Weintraub, M. (1997) Explicit Word Error Minimization In N-Best List Rescoring. In *Proc. EUROSPEECH*, vol. 1, Rhodes, Greece, 163-166.

123. Sun Microsystems, Inc. (2000) *Jspeech Grammar Format*, W3C Note 05 June 2000.

124. Takezawa, T., Kita, K., Hosaka, J., and Morimoto, T. (1991) Linguistic Constraints for Continuous Speech Recognition in Goal-Directed Dialogue. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. May 1991, 801-804.

125. Tellme Studio (2002) http://studio.tellme.com.

126. Tran, B., Seide, F. and Steinbiss, V. (1996) A Word Graph Based N-Best Search in Continuous Speech Recognition. In *Proceedings of ICSLP '96*, Volume 4, Philadelphia, PA, 1996, 2127-2130.

127. Valverde-Albacete, F.J., and Pardo, J.M. (1996) A Multi-Level Lexical-Semantics Based Language Model For Guided Integrated Continuous Speech Recognition. In *Spoken Language, 1996. ICSLP '96. Proceeding Fourth International Conference on*, vol. 1, 3-6 Oct. 1996, Philadelphia, PA, USA, 224-227.

128. VXML Forum (2000) Voice eXtensible Markup Language Specification, version 1.0, Mar 2000. http://www.w3.org/TR/2000/NOTE-voicexml-20000505/.

129. Walker, M., Wright, J. and Langkilde, I. (2000) Using Natural Language Processing And Discourse Features to Identify Understanding Errors In a Spoken Dialogue System. In *Proceedings of 17th International Conference On Machine Learning*. Proc. 1111-1118.

130. Wang, Y.Y., Mahajan, M., and Huang, X (2000) A Unified Context-Free Grammar and N-gram Model For Language Processing. In *Acoustics, Speech, and Signal Processing, 2000 ICASSP '00. Proceedings of 2000 IEEE International Conference on*. Vol. 3, 5-9 June 2000, Istanbul, Turkey, 1639-1642.

131. Ward, N. (1996) Second Thoughts on an Artificial Intelligence Approach to Speech Understanding, In *14th Spoken Language and Discourse Workshop Notes (SIGSLUD-14),* Japan Society for Artificial Intelligence, 16-23. ftp:ftp.sanpo.t.utokyo.

132. Ward, W., and Issar, S. (1996) Recent Improvements in the CMU Spoken Language Understanding System. In *Proceedings of ARPA Human Language Technology Workshop*, 1996, 213-216.

133. Ward, W., and Issar, S. (1994) Integrating Semantic Constraints into the SPHINX-II Recognition Search. In *Proceedings of ICASSP'94*. Adelaide, Australia, 1994, 2017- 2019.

134. Ward, W., and Young, S. (1993) Flexible Use of Semantic Constraints in Speech Recognition. In *Proceedings of ICASSP '93. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, USA, IEE; New York, NY, USA, 1993.

135. Weber, H. and Gőrz, G. (1999) Symbolic Parsing and Probabilistic Decision Making. The Speech and language Experience with Hybrid Information Processing. In *Proceedings of HCI (2)*, April 6, 1999, 802-806.

136. White, G.M. (1990) Natural language Understanding and Speech Recognition. *Communications of the ACM August 1990*, Volume 33 Issue 8.

137. Witten, I.H, and Bell, T.C (1991) The Zero-frequency Problem: Estimating the Probabilites of Novel Events in Adaptive Test Compression. In *Proceedings of IEEE Transactions on Information Theory*, July 1991, 37(4): 1085-1094.

138. Xu, D., Zhu, P, Huang, T., and Chen, D. (1988) Using High-Level Linguistic Knowledge for Chinese Speech Recognition. In *Pattern recognition, 1988, 9th International Conference on*. 14-17 Nov., 1988, Rome, Italy, vol.1, 197-199.

139. Yamamoto, H., Isogai, S. and Sagisaka, Y. (2001), Multi-Class Composite N-gram Language Model for Spoken Language Processing Using Multiple Word Clusters. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. In *Meeting of the Association for Computational Linguistics*. 531-538.

140. Zhu, X. and Rosenfeld, R. (2000) Improving Trigram Language Modeling with the World Wide Web. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2001*. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Appendix A: A Survey of Research on Using Natural Language Features to Improve Speech Recognition Accuracy

ABSTRACT

With the growing interest and demand for human-machine interaction, more and more work concerning speech-recognition has been carried out over the past decades. This survey investigates the techniques involved in speech-recognition, including the widely-used robust stochastic approaches, the prevalent grammar-based methods, combined N-gram and grammar-based techniques, parsing techniques used for speech recognition, the approaches of integrating syntax and semantics, and other techniques. Since language features play a significant role in speech-recognition, the techniques of using semantics in speech-recognition are emphasized. Although many research prototypes and even commercial applications have been deployed, many challenges remain in the development of speech-recognition technologies.

Table of Contents of Survey

1. THE NEED FOR SPEECH RECOGNITION

Looking back on human history, language marked the evolvement of the humankind, words recorded the civilization of the human society, and speech communication has been the most common, convenient, and preferred methods of communication of human beings. For the majority of human beings, speech communication is the easiest way to convey information from human to human, for it can make hands free, can proceed in the dark, and even can reach very far distance through radio and telephone.

The question is, can machines make use of all of the advantages of human's natural language speech? If a machine can understand natural language, one can easily interact with that machine (just like communicating with another human) in natural language to retrieve information, conduct transactions, or perform other problem-solving tasks. For example, people can direct the machine, in spoken language, to execute commands; with the assistance of external equipment (e.g., telephone), activate remote controls or fulfill remote commercial transactions; visit the speech web with natural spoken language input and voice output without text or graphic interfaces; virtual-reality technology can be strengthened with more-real natural-speech interactions; machines can dictate what one says and save it as a text document; machines can automatically translate one language into other languages and the people with vision disability will suffer less on account of the help of machines equipped with a natural-language ability.

Over decades, a number of Artificial Intelligence (AI) researchers have been striving to build models to interact between humans and machines with natural-language speech. The conversational interfaces in the 1950s marked the origin of spoken-dialogue systems (McTear, 2002), whereas, it is only in the past decade that speech technology has achieved advanced progress with the introduction of both research prototypes and commercial applications, such as SPHINX (the first accurate large-vocabulary continuous speaker-independent speech recognition system, which was developed at Carnegie Mellon University (Huang et at., 1992) (Lee, 1988) (Kita and Ward, 1991)), ATIS (an actual spoken language Air Travel Information System (Moore *et al.*, 1995)) , CommandTalk (a spoken-language interface to a battle-field simulator (Goldwater *et al.*, 2000) (Dowding *et al.*, 1999) (Stent *et al.*, 1999) (Moore *et al.*, 1997)), and the JUPITER weather information system (developed in MIT, (Glass, 1999)).

The potential of speech technology has aroused the attention of some telecommunication and software companies. Some newly-developing areas, e.g. computer-telephony integration, are demanding speech solutions. Subsequently, the corresponding products were created, such as, voice portals (McTear, 2002), which provide a speech-based interface between a telephone user and web-based services.

A complete spoken-dialogue system involves the integration of the following components (McTear, 2002) (Han, 2000) (Glass, 1999): a speech recognition component, a language understanding component, a dialogue management component, a component for communication with an external system, a response generation component, and a speech output component. These components work in a sequential stream, in which the first component receives the user's input, then the output from that component feeds into the next component as the input, and so forth, until the consequent voice output is synthesized for the user. An overview of the interaction of the components in spoken dialogue system is as follows (McTear, 2002):

The speech-recognition component receives the user's input utterance and converts the continuous-time signal into a sequence of discrete units for the use of the language-understanding component. As the language component receives the information from the previous speech-recognition component, it analyzes the discrete units and derives a meaning representation for the next dialogue control component. The dialogue-management component controls the dialogue flow by determining whether the user has provided sufficient information, also communicating with the external application and the user. Usually, it is a database that acts as the external system component for the requested information retrieval in the spoken-dialogue system. Finally, the response-generation component will construct the message retrieved from the external system component corresponding to the user's request and send it to the speech output component to synthesize the voice output for the user.

From the above architecture, it can be seen that speech recognition forms the basis, the fundamental part, and the gateway of the whole spoken-dialogue system. Recognition accuracy directly affects the performance of the subsequent processes. The main task of speech-recognition research is to build a suitable language model to determine the individual words of the input utterances and to specify the possible sentences for the system (McTear, 2002). The technology of speech recognition is concerned with various linguistic features, including syntax and semantics, and statistical or grammar-based techniques are also involved.

2. PROBLEMS IN SPEECH RECOGNITION

Since countless human conversations proceed every day without any trouble, people do not realize that they have overcome many problems in such conversations, such as, disfluencies, interruptions, confirmations, anaphora, and ellipsis. For example, Glass (1999) showed a statistic number that almost 50% acknowledgements (e.g., "okay", alright", "uh-huh") occurred in the customer dialogues. In addition, many utterances can be understood only in particular context within some domains. However, all the above challenges and others, such as noise of the background and speaker variation, are very difficult for machines to tackle. Due to the large variability and flexibility of human speech and the speciality of machines (compared to human beings), there are various problems in the speech-recognition process.

- **Recognition Accuracy.**

  A human being only makes a few mistakes in interpretation if he/she knows the words. However, it is not the same in human-machine speech interaction. There are a variety of factors that may cause recognition ambiguities or errors that degrade the performance of the whole spoken dialogue system. Improved accuracy of the speech recognizer is one of the goals that numerous AI researchers have been pursuing. High accuracy of speech recognition is very important.

- **Robustness.**

  Robustness means the extent to which a system handles errors or "unexpected" input. Robustness is crucial in language systems for the reason that the inability or low performance in processing incorrect utterances will cause unacceptable degeneration of the overall system (Ballim and Pallotta, 2000). Like human beings, the ideal spoken-language models should tolerate disfluencies, out of vocabulary words, incomplete or ungrammatical utterances to some extent in speech communication. In reality, various uncertain and flexible factors of the spontaneous dialogue add more difficulties to speech recognition. There is still a lot to be desired for the state-of-the-art language models toward the goal of robustness.

- **Large vocabulary.**

  Many spoken-language systems are supported by a large vocabulary so that they can cover as many as possible of the spontaneous utterances. On the other hand, a large vocabulary can make the language system intractable, especially, the large number of categories due to the huge unrelated entries (Rosenfeld, 2000a), is a great challenge for speech recognition. For example, in a large vocabulary, there is no closer relation between BANK and LOAN than that with COUNTRY. The relative independence in a vocabulary leads to the huge intractable parameters. Suppose that the related words can be grouped into one category, for example, BANK and LOAN belong to the same category FINANCE, the number of the categories in the vocabulary must be much fewer than the original individual words. (This idea can be found in class N-gram technique, discussed in section 3.2). Some large-vocabulary related techniques in speech recognition are discussed in section 9.2.

- **Flexibility** (Milward, 1999).

  An ideal spoken-dialogue system should be able to accept a user's flexible utterances, allow the user to supply extra information and make reasonable responses. While the fact is that the user may not realize the bounds of the domain, they may make free queries that are out of the capability of the system. For example, the JUPITER weather information system (developed in MIT) can only forecast short-term weather (Glass, 1999). So, if the user asks for "What is the weather in two months?" the JUPITER weather information system cannot give an answer. Under such circumstance, the system is expected to give the user appropriate help to direct him/her to formulate an acceptable query.

- **Brittleness across domains** (Rosenfeld, 2000a).

  The efficiency of the current language models depends much on the domains on which they are trained. For example, a language model trained on business is not appropriate to recognize utterances about sports.

- **False independence assumption** (Rosenfeld, 2000a).

  While building a tractable language model, the state-of-the-art technology assumes some independence among different portions of the same document. For example, the N-gram model determines the probability of the current word in a sentence only by the identity of the last N-1 words, which loses the long-term dependency. In particular, semantic constraints cannot be modeled with small N.

3. STOCHASTIC (STATISTICAL) TECHNIQUES IN SPEECH RECOGNITION

At present, there exist various ways to construct language models for speech recognition. Roughly, the approaches can be categorized into stochastic (statistical) models (which require a large corpus of training data) (discussed in this section), and grammar-based models (which uses grammars to specify the utterances) (discussed in section 4) (Rayner *et al.*, 2000b). A language model consists of a vocabulary (a set of words that can be recognized by the system) and grammar (a set of rules by which sentences are parsed or constructed) (Souto *et al.*, 2002). The grammar can be a set of linguistic rules or a stochastic (statistical) model. Generally, if a substantial domain corpus is available, a stochastic (statistical) language model is better as it is more robust; otherwise, a Context-Free grammar-based language model may be more appropriate.

So far, many language models have been successfully constructed for stochastic (statistical or probabilistic) techniques. Stochastic language models are designed and evaluated to optimize speech-recognition accuracy. A Statistical Language Model (SLM) is simply a probability distribution P(s) over all possible sentences s, or spoken utterances, documents, or any other linguistic units (Rosenfeld, 2000a).

The typical architecture of the speech language-understanding system that uses a stochastic model is described in (Knight *et al.*, 2001) as follows: firstly a domain corpus is collected and used to train the statistical language model; then the statistical language model is incorporated into the recognizer; after that, a robust phrase-spotting parser is built to analyze the text output of the recognizer and produce semantic representations in the form of slot/filler pairs.

3.1 N-grams

The N-gram is the most frequently-used stochastic technique in speech recognition. N-gram means, with enough amount of training data, each word can be predicted from the previous N-1 words (Souto *et al.*, 2002). Namely, the probability of a word's occurrence can be predicted by the preceding N-1 words and one or more candidate words are output in some ranked "recognition hypothesis list".

The type of training data to be collected is determined by the task of the model. For example, if it is a model for medical application, the training data should be focused on the medical reports, papers and other resource instead of that in sports or fashion. Often, a trigram (N=3) is used with large training corpora (million words), whereas a bigram (N=2) in the smaller set of training data (Rosenfeld, 2000a).

The primary advantages of the N-gram lie in its robustness.

3.2 Multi-class Composite N-gram (Class N-gram)

The sparseness (the infrequency of word sequences in a corpus (Magerman and Marcus, 1990)) is a common problem in the N-gram approach, even with the large corpora. For example, in some training corpora, many triplets (in trigram) appear only once or few times, thus, the straightforward estimation of N-gram probabilities from counts is not viable. To address the problem of data sparseness, Rosenfeld (2000a) describes various techniques, such as the discounting the maximum likelihood estimation (Witten and Bell, 1991) (Good, 1953), recursively backing off to lower order N-grams (Kneser and Ney, 1995) (Ney *et al.*, 1994) (Katz, 1987), linearly interpolating N-grams of different order (Jelinek and Mercer, 1980), constituent boundary parsing method (discussed in section 6.6), and using high level semantic domains (discussed in section 8.7).

According to Rosenfeld (2000a), Yamamoto *et al.* (2001) propose an effective "class N-gram" technique by using vocabulary clustering to battle the sparseness problem. Multiple words are assigned to one word class representing either syntactic categories (for example, noun or verb) or semantic categories (for example, days of the week, names or airports) (McTear, 2002) (Baggia *et al.*, 1999), thus, the transition probabilities from word to word are approximately changed to that from class to class. Consequently, with the decreased search space (obviously, the number of classes is much smaller than that of the original words), the perplexity is reduced and the recognition accuracy increases. The key of this technique lies in the clustering, which determines the quality of the model. It works better within small domains by manual clustering semantic categories, and it is not the same case in the less constrained domains (Rosenfeld, 2000a).

3.3 Decision-Tree Models and Semantic Classification-Tree Models

Decision-tree models (Rosenfeld, 2000a), as well as semantic classification-tree models (Noth *et al.*, 1996) take the advantage of decision-tree structure. "A decision tree can arbitrarily partition the space of histories by asking arbitrary binary questions about the history at each of the internal nodes" (Rosenfeld, 2000a). The probability distribution of next word is constructed, based on the training data at each leaf. Interpolating leaf distribution with internal-node distribution found along the path can contribute to reduce the variance of the estimate (Rosenfeld, 2000a).

Seen from the experiment of (Noth *et al.*, 1996), the semantic classification-tree model, combined with different knowledge sources, improved the recognition rate. However, since the tree structure has space complexity of O(bd), where b is the branching factor and d is the depth of the tree, the space of the history is very large, and the space of possible questions is much larger (Rosenfeld, 2000a). Therefore, techniques to prune the large trees are needed. For example, the CART-style LM used a history window of 20 words and restricts questions to individual words to control the history space (Rosenfeld, 2000a).

3.4 Adaptive Models

Domain restriction remains one of the problems in speech recognition (discussed in section 2). Adaptive models in (Rosenfeld, 2000a) provide the possibility to alleviate the domain problem. The Cross-Domain Adaptation model takes advantage of a cache to transfer test data to the language model without training. In the Within-Domain Adaptation model, the test data comes from the same source, but this particular source consists of many subsets of various topics, styles or both. Then the adaptation can proceed among the subsets, and two different domains can be combined to construct a general model so that the language model can cover a wider domain.

A problem with the adaptive models is that an increase in training data does not guarantee a corresponding improvement in the accuracy of the language model. The reason is that the adaptive models may cover several domains and it is possible that the data increase occurs on some domains that have little influence on the model in other domains.

3.5 N-best Filtering or Rescoring

N-best filtering or rescoring is a very simple search technique (Moore, 1999). Just as its name implies, this technique always chooses the best one in the sorted recognition hypothesis list. According to different criteria or different language models, the order of the hypothesis list is different. Section 8.2 discusses several examples of the N-best technique for the post-processing of the speech recognition output. Some implementations require that N be known in advance (Murveit and Moore, 1990), while there are also techniques (Seide *et al.*, 1996) (discussed in section 8.2) that do not have this requirement.

The primary advantage of the N-best approach is its simplicity. The disadvantage is high computational cost for large N. Generally speaking, if N is small the computation rate is low, but the increase of the length of a sentence may cause an exponential increase of N (Murveit and Moore, 1990).

3.6 Learning Techniques

Stochastic techniques are popular for their good recognition accuracy and robustness. However, it is not a trivial task to obtain the huge corpus of training data. The following are some techniques to obtain the training data.

- **Bootstrapping** (Rayner *et al.*, 2000a) (McCandless and Glass, 1994) (Baggia *et al.*, 1999).

  Bootstrapping is the simplest and cheapest way to collect training data. The main idea is to build an initial version of the system using a hand-coded model, then put it into practice to collect more data. Recursively, the data is used to construct a new language model and that is used to collect new data. This cycle can be repeated until satisfactory accuracy is achieved.

- **Use of The World Wide Web**

  Nowadays, with the boom of the World Wide Web, the information available online has been growing at an exponential factor. Undoubtedly, the World Wide Web is destined to be the main source for collecting training data for stochastic methods.

Taking advantage of the World Wide Web access to a huge amount of information online and use of effective search engines, Zhu and Rosenfeld (2000a) proposed an efficient method to obtain the N-gram (N=3) estimates for statistical language modeling. The N-gram was submitted to the web search engine as a phrase query to retrieve the corresponding web pages containing the N-gram data. At the same time, the number of the retrieved web pages and the count of the N-gram were calculated.

3.7 Summary

Statistical Language Models (SLMs) have the advantages of simplicity, flexibility, adaptation, higher recognition accuracy and robust performance. Also a key advantage of SLMs over grammar-based models is the ability to handle the input that is not in the language defined by the grammar.

On the other hand, SLMs suffer the unavoidable disadvantage of the costly collection of huge amount of training data. In ATIS (Air Travel Information System (Moore *et al.*, 1995)), it took over a year and $1M to carefully collect the 20000 utterances (Knight *et al.*, 2001). According to (Rosenfeld, 2000a), an informal estimation by IBM states that an effective bigram models needs several hundred million words as training data; and the trigram models are probably to absorb a few billion words. The worst is that most of the training data comes from written language, which does not really reflect the spontaneous nature in spoken language.

Even though the World Wide Web provides a great opportunity for collecting large amount of training data in all kinds of domains, the quality of the statistical language models is not improved by a corresponding factor. Actually, the improvement of SLM is asymptotic (Rosenfeld, 2000a), which means that even though the online resource can increase at an exponential rate, the quality of the SLM is not likely to improve by a significant factor.

Data sparseness and limited scope dependencies are also two problems existing in the up-to-date standard N-gram-based statistical language models (Chappelier *et al.*, 1999)

(Weber and Görz, 1999). Section 3.2 describes a possible solution to the data sparseness problem, and section 8 discusses the possible solutions to the limited-scope dependencies problem.

4. GRAMMAR-BASED SPEECH RECOGNITION

As an alternative to Statistical Language Models (SLMs), which apply word probabilities (N-gram) as the only form of language knowledge (Rosenfeld, 2000a), grammar-based speech recognition describes the language features in a set of rules to generalize over a certain application domain.

According to (Knight *et al.*, 2001), the up-to-date grammar-based strategy (which is usually adopted by commercial organizations) is like this: use Nuance or Speechworks as a standard commercial platform; then hand-code a grammar in some subset of Context-Free Grammar (CFG), and extend the grammar with semantic annotations; later on, using a system-initiative dialogue strategy, code in Nuance's Speech Objects or Speechworks' Dialogue Models or VoiceXML.

4.1 Context-Free Grammars (CFGs)

A Context-Free Grammar (CFG) is a crude, yet well-understood, model of natural language. A CFG consists of a vocabulary, a set of non-terminal symbols, and a set of production or transition rules. Usually, a CFG can be defined as a set of rules that have a single atomic grammatical category on the left-hand side, and a sequence of atomic categories and words on the right-hand side (Moore, 1999), (Amaya *et al.,* 1999). Based on the fact that all context-free rules can contain only one symbol on the left hand side, and it is free to be replaced by the right side rules, comes the name "Context-Free Grammar".

Unlike the finite-state grammar, a CFG allows recursion (Moore, 1999), which makes it much more suitable for defining linguistically-based language models. However, it does not include more detail of the language constraints, which may be significant in the grammars definition (Moore, 1999). For example, to define the sentence with the structure that a noun phrase followed by a verb phrase, in CFG, the rule can be written as S → NP VP. In general, a sentence will express the person and number, the verb tense, and whether it is interrogative or declarative. The CFG can only define such detail information by adding more rules for each person, tense and so on. Obviously, this will greatly increase the number of the rules. An alternative is to annotate the CFG grammar in some ways, which are discussed in sections 8.4.1 and 8.4.2.

4.2 Statistical or Probabilistic Grammars

- **Probabilistic Context-Free Grammars (PCFG).**

The mathematics of Probabilistic Context-Free Grammars (PCFG) is the basis of most hybrid approaches in Natural Language Processing. Probabilistic Context-Free Grammars (PCFGs) are CFGs with a probability distribution defined over all productions that share their left-hand side (Rosenfeld, 2000b), (Moore, 1999), (Weber and Gőrz, 1999). For the example that the conditional probability of the rule S→ NP VP is 0.5, Moore (1999) explains that this means: if there is a phrase S, there is 0.5 chance that it consists of a phrase of NP followed by VP.

Rosenfeld (2000b) points out that the consequence of fusing CFGs and bigrams was a model with size (number of parameters) comparable to a bigram yet performance comparable to a trigram. However, it is necessary to consider about both the CFG itself and the context-free production probabilities to use PCFGs for unconstrained language. Since the-state-of-the-art CFG cannot sufficiently cover unconstrained English, and it is difficult to globally optimize context-free production probabilities, and even with the possible global optimum, the context-free production probabilities might not have sufficient expressive power to capture the true distribution of parses, PCFGs cannot compete (statically) with the conventional N-gram (Rosenfeld, 2000b).

Furthermore, if the probability is based on a Unification Grammar instead of a Context-Free Grammar, a Probabilistic Unification Grammar is obtained. However, Moore (1999) states that there appears to be no published reports of models that incorporate all the constraints of a complex unification grammar into a statistical model.

- **Probabilistic Dependency Grammars (PDG).**

Similar to regular N-gram, in Probabilistic Dependency Grammars (PDG), each word is predicted based on a number of other words. The difference from conventional N-gram is that, in the latter, each word is predicated from the N-1 words immediately before it; whereas in PDG, the words act as the predictors depend on a hidden variable, the dependency graph (Rosenfeld, 2000a). Typically, a sentence *s* is parsed to generate the most likely dependency graphs Gi (with attendant probabilities P(Gi)); then compute each generation probability P(s|Gi) (either N-gram style or an Maximum Entropy (ME) model); finally, the complete sentence probability $P(s) \approx \Sigma_i P(Gi)*P(s|Gi)$ (the reason for the approximation is that the P(Gi) themselves were derived from the sentence s). Sometimes P(s) is further approximated as P(s|G*), where G* is the single best scoring parse (Rosenfeld, 2000a). Rosenfeld (2000a) introduces an example model developed by Chelba *et al.* (1997), which uses the parser of (Collins, 1996) to generate the candidate parses, and uses maximum entropy to train the parameters.

4.3 Discourse Grammars

The notation of Discourse Grammar was proposed by Churcher *et al.* (1996) to break the large syntax into smaller syntaxes to improve the performance of the language models that have lower perplexity and ambiguity. The idea supporting this approach is that, generally, the smaller syntax contains fewer words and less complicated structure than the original one, hence is potentially less ambiguous. Furthermore, Churcher *et al.* (1996) broke the discourse into discourse segments that reflect a set of utterances with some properties in common. A discourse segment can be the utterances discussing a certain topic, or even the discourse between a set of speakers, namely, a dialogue. Using three syntaxes based on a corpus of transmissions between the ATC and pilots, Churcher *et al.*

(1996) explored experiments and achieved 8% increase compared to the original large syntax. Also, similar ideas can be found in (Frost and Chitte, 1999) (section 9.1 discusses the details).

4.4 Semantic Grammars

Stochastic, syntactic and semantic grammar methods are widely used in speech recognitions with their respective features. According to Demetriou and Atwell (1994a), semantic grammars are usually represented as transition networks, and provide stronger constraints than pure syntax by integrating semantic conditions closely with the syntactic rules of the grammar. A syntactic grammar is effective in describing the structure of phrases and sentences, whereas semantic constraints are more powerful for languages whose phrase orders are not very constrained, such as Japanese (Takezawa *et al.*, 1991).

For more details about the semantic grammars, refer to sections 8.4 and 8.5.

4.5 Summary

Stochastic (statistical) techniques and grammar-based techniques are two main streams in language-model constructions. It was reported in (Knight *et al.*, 2001) stochastic (statistical) language models were popular around 1995, but by 2001, grammar-based language models took the prevalent position in commercial products.

Compared to statistical techniques, grammar-based speech recognition is more common and easier to use and has reasonable recognition accuracy for small domains. Actually, within the domain covered by the grammar, the recognition accuracy is pretty high and the fact is that the user usually has the rough idea about the system and stays in the domain (Rayner *et al.*, 2000b). In addition, for simple applications, good grammars can be constructed quickly and efficiently (Rayner *et al.*, 2000b). In contrast to the stochastic techniques, grammar-based techniques have another compelling advantage that they do not require large amount of training data that is difficult and expensive to collect.

Just as a coin has two sides, the grammar-based technique also has its disadvantage that it needs experts to write high-quality grammars and the grammar rules are difficult to maintain and extend. Geistert (1998) developed a Grammar Interface Tool (GIT), by which the grammar and the lexicon for a specific application can be designed from some example sentences annotated with their respective semantic interpretation.

In addition, the grammar-based recognition is not as robust as statistical techniques. For example, it will make mistakes while encountering the utterances that are not covered by the grammar. Also, the lack of robustness can be a result of over-constraint (Glass, 1999). The DARPA ATIS program (Ward and Issar, 1996) (Noord *et al.*, 1998) successfully solved this problem by keyword and phrase spotting methods instead of the fully analyzing the whole utterance. Seneff (1992) proposed another approach that they firstly analyzed the complete utterance, then backed off to robust parsing if no complete parse was found.

5 COMBINED STOCHASTIC (STATISTICAL) AND GRAMMAR-BASED TECHNIQUES

So far, the success of the stochastic (statistical) language model approach has been proved by its simplicity, flexibility, better recognition accuracy and robustness. Meanwhile, it suffers from the unavoidable difficulty of collecting large and expensive training data corpus. On the other hand, the grammar-based language model adopts a set of grammatical rules instead of calculating the word occurrence possibilities from the training data in the recognition. For simple applications, the grammar is not too difficult to construct, but it is cannot handle out-of-coverage utterances. The question is, is it feasible to take their respective advantages and overcome the disadvantages by integrating the stochastic techniques and grammar-based techniques?

The ATIS, Air Travel Information System (Moore, *et al.* 1995), is one example of the successful language models, which uses a CFG in parsing and produces a sequence of grammatical fragments, then, the trigram (N=3) is applied. The results of such integration of CFG and SLM included a 15% reduction in a speech recognition error rate. Using syntax on trigram, Chelba (2000) carried out experiments on the Wall Street Journal, Switchboard and Broadcast News corpora and achieved the improvement in both perplexity and word error rate over the original trigram. Also, Rayner and Carter (1997), Geutner (1996), and Jones *et al.* (1993) achieved robust and efficient performance within a linguistically motivated framework by combining the rule-based and statistical methods.

Knight *et al.* (2001) implemented the preceding idea in the experiment of a home device control system. They firstly applied the Nuance Toolkit Grammar Specification Language (GSL) to set up a CFG grammar-based system. As a language model, this grammar-based system accepts the user's input and collects the utterances as the training corpus for a Stochastic (Statistical) Language Model (SLM.). The SLM uses a standard back-off trigram model over the training corpus obtained from the grammar-based system. The results show that the grammar-based language model performs well for in-coverage sentences, but very poorly on out-of-coverage ones. However, the SLM makes slightly more word errors for in-coverage sentences, but performs much better for out-of-coverage examples.

Benedi and Sanchez (2000) linearly combined the N-gram models and a stochastic grammatical model for language modeling. A classical N-gram model was used to capture the local relations between words, then, a stochastic grammatical model is used to represent the long-term relations between syntactical structures. A category-based SCFG and a probabilistic model of word distribution in the categories are used to define this grammatical model for large-vocabulary complex tasks. Experiments using the Penn Treebank corpus showed the improvement of 30% in perplexity with regard to the classical N-gram models.

In (McCandless and Glass, 1994), a simple Context-Free Grammar was firstly used to decode the training data and iteratively generalize and reduce the grammar. Then this grammar was combined with a phrase class N-gram formalism to assign probability to test sentences. Compared to traditional trigram, a unified model of CFG and N-gram significantly reduced the perplexity (Wang *et al.* 2000) (McCandless and Glass, 1994) and the number of parameters (McCandless and Glass, 1994).

In addition, Siu, and Ostendorf (2000) integrated a context-dependent phrase grammar in a variable N-gram framework, and the experiment result showed the improvement of recognition accuracy on the Switchboard corpus in comparison with both the baseline trigram and the variable N-gram alone.

6 PARSING TECHNIQUES FOR THE OUTPUT FROM A SPEECH RECOGNIZER

Parsing is usually involved in speech recognition to determine whether the word strings are valid or not, according to the defined grammar. A parser is responsible to produce the grammatically syntactic and semantic interpretation of a sentence. Parsing is used in the following two ways: (1) during the recognition process to guide the recognizer and (b) to post-process the output from the recognizer to pick the most likely sentence.

Since the spontaneous speech has its particular features such as containing ungrammatical utterances, words or sentences that are not covered by the system's lexicon and grammar, online verbal corrections or other extra-grammaticalities (Kaiser *et al.*, 1999), it is difficult to parse the output from the recognizer if it is not a grammar-directed recognizer. The following are some parsing techniques that were used in some speech-recognition systems.

6.1 Finite State Parser

A Finite State Parser explains why the input is accepted by processing the recognition of the input sequences one by one and returns the sequence of transitions that was made (Blackburn and Striegnitz, 2002). Namely, the output of the Finite State Parser is a sequence of nodes in the recognizing order.

PROFER is a Predictive **RO**bust **F**inite-state pars**ER** system with the ability to produce sequential prediction sets and incrementally build a *case-frame* representation of concepts extracted from the input (like PHOENIX, which is discussed in section 8.8) (Kaiser *et al.*, 1999). PROFER can be used as a stand-alone semantic parser, and as a stand-alone finite-state predictor. Compared to the chart-based or generalized left-right (GLR) parsers, PROFER's lower complexity and robustness has been showed in Kaiser (1999). PROFER has been used in various limited task domains by providing a higher-level, grammatical language model for speech recognition.

The importance of finite-state networks has been stressed in many speech recognition systems. Also, Casacuberta *et al.* (2001) presented the feasibility of the finite state transducer (a specific stochastic finite state network) in EUTRANS system (a speech-to-speech translation system).

6.2 Word Lattice Parsing

Word lattice parsing is probably the oldest approach to integrate complex language models into speech recognition (Moore, 1999). The architecture is as follows (Moore, 1999) (Atwell and Kevitt, 1994) (Hazen *et al.*, 2000): for the input segment, the recognizer produces a set of word hypotheses and assigns them acoustic scores, then uses the natural-language parser or other language model to find the path of the words with best acoustic and language model scores through the word lattice. The disadvantages include the heavy computational burden on the system (Murveit and Moore, 1990).

6.3 Left-corner Parsing

It is possible for the pure bottom-up or top-down parsing to make mistakes under some circumstances (Blackburn, and Striegnitz, 2002). While the combination of the preceding two methods, obtained left-corner parsing, can get dramatic effect (Blackburn and Striegnitz, 2002). A left-corner parser firstly uses a bottom-up parsing technique to look at the first word of the input string, and determines its category, and then looks for a rule with this category as the first symbol on its right hand side. Then the left-corner parser uses this rule as top-down information and tries to recognize the rest of the right-hand side.

6.4 GLR * Parsing

The GLR * parsing algorithm in (Lavie, 1996) was based on Tomita's Generalized LR (GLR) parsing algorithm. The GLR evolved from the LR parsing techniques. The mechanism of LR parser is bottom-up parsing, left-to-right scanning. Driven by a table of grammatical parsing actions, LR parsers are deterministic and efficient. Tomita's Generalized LR (GLR) parsing algorithm is an extension of LR for non-LR languages. If the actions in the parsing table conflict (non-determinism), the GLR will efficiently try all possible actions in a pseudo-parallel fashion. The data structures and the parsing table in GLR* are both similar to GLR. GLR* extends GLR only in the run-time parsing way. GLR* intends to detect and reject the ungrammatical input at the possible earliest stage. It solves the problems of noise input and limited grammar coverage by ignoring the unparsable words and fragments and conducting a search for the maximal subset of the original input that is covered by the grammar.

6.5 Feature Structure Parser

The FEAture Structure PARser, called FeasPar, which learns parsing spontaneous speech, was proposed by Buo and Waibel (1996). The primary elements of FeasPar are "chunks", their features and relations. They are structured into a neural network collection and a search. The neural network divides the input sentence into chunks, which are labeled with feature values and chunk relations. Then, depending on the feature structure, which acts as the constraint, the search obtains the most probable and consistent feature structure. After being trained, tested and evaluated, the FeasPar (with the Spontaneous Scheduling Task) was compared with a hand-modeled LR-parser from six aspects. Buo and Waibel (1996) concluded that FeasPar performed better than LR-parser

6.6 Constituent Boundary Parsing

Constituent boundary parsing was proposed by Magerman and Marcus (1990) as an alternative to traditional grammar-based parsing methods, though it actually included a distituent grammar. The constituent boundary parsing method treats part-of-speech sequences as stochastic events suitable for probabilistic models. The mutual information values of the part-of-speech N-grams within the sentence determine the constituent boundaries. Since it computes the tag N-grams for a set of tags (with sufficient frequency) rather than word N-grams, the sparseness is not the problem in constituent boundary parsing method (Magerman and Marcus, 1990).

6.7 Two-level LR Parsing

To integrate speech and language for an automatic interpreting telephone, Takezawa *et al.* (1991) explored a predictive two-level LR parser based on an inter-phrase grammar, which was developed according to a half-million-word-dialogue database on "an international conference secretarial service". Firstly, this inter-phrase LR parser predicts next phrasal categories (e.g. Noun Phrase (NP)) depending on the inter-phrase LR parsing table. Then, all the phones predicted by the NP initial state are picked up by the intra-phrase LR parser and the HMM phone model is invoked to verify the existence of these predicted phones. Once the NP candidates have been recognized, the next phrasal category (e.g. Verb Phrase (VP)) is predicted by the inter-phrase LR parsing table, and the above process continues until the entire speech data has been processed. The experiments in (Takezawa *et al.*, 1991) show the effectiveness of the two-level LR parsing over the phrase lattice parsing method.

6.8 History-Based Grammars (HBGs)

A History-Based Grammar (HBG) is essentially a probabilistic model, which incorporates the detailed linguistic information such as lexical, syntactic, semantic and structural information to resolve the ambiguity (Black *et al.* 1992). HBG combines a Treebank (a corpus of bracketed sentences) and a decision tree to determine the correct sentence from the parse tree, where the probability depends on the information of the partial derivation of decision tree. Black *et al.* (1994) reported an improvement from

PCFG to HBG of 15% increase of parsing accuracy rate.

7 USE OF SEMANTICS IN SPEECH-RECOGNITION

7.1 Introduction

Language features are very effective in any system for reducing the number of possible utterances and for prioritizing utterance hypotheses (Hermannsdottir, 1996). Takezawa *et al.* (1991) said that "the accuracy of speech recognition heavily depends on what kinds of linguistic knowledge are used". At the current state of the art, to achieve high accuracy in speech recognition with moderate to large vocabularies (hundreds to tens of thousands of words), language models are necessary (Moore, 1999), (Harper *et al.*, 2000), (Takezawa *et al.*, 1991) and (Seneff *et al.*, 1995) as discussed earlier. Takezawa *et al.* (1991) categorizes linguistic constraints into syntactic, semantic, pragmatic and contextual constraints. The models, including knowledge of syntax, semantics, domain, task and current dialogue state, can assist the speech recognition process effectively (Johnson, 2000), (Demetriou *et al.*, 2000), (Ward, 1996), (Hunt,1994), and (Loken-Kim, 1988).

One of the features of the spoken language system is its interaction, which requires the methods for representing and integrating knowledge from different sources (White, 1990). Various linguistic constraints can be incorporated into the speech recognition process tightly or loosely. Tight integration means the linguistic constraints are directly incorporated into the recognition algorithms (Chappelier, 1999) (Harper *et al.*, 1994).

The advantage of tight integration is the smaller size of hypotheses space and strong restrictions on the grammar. In addition, since the language information usually contributes to reduce the perplexity of the system, it is an advantage to tightly integrate; however, too tight integration usually reduces robustness. In addition, tight integration often makes the big systems intractable and difficult to train.

Preferred by Chappelier (1999) and Harper *et al.* (1994), loose integration architecture means knowledge sources are applied one by one in a sequential order. This modular architecture makes it possible to use each language-processing technique with little modification. The other advantage is that, the update of a powerful language model will not increase the computational cost or the amount of training data required (Harper *et al.*, 2000).

Syntax is the structure of expressions in a language. It defines the relationship among characters or groups of characters, independent of their meanings or the manner of their interpretation and use. Semantics defines the relationships between symbols and their meanings; characters or groups of characters to their meanings. Syntax is responsible for the sentence structure. Syntax can be used in conjunction with a statistical model to guide the recognizer. Semantics contributes more to the meanings of the words or sentences. Appropriate integration of syntax and semantics can help improve the recognition performance. However, in many cases syntactic information alone is not sufficient in restricting the search space for speech recognition (Takezawa *et al.*, 1991). And the fact is that almost all language models implicitly or explicitly embody the semantics.

Semantics can be built into language models explicitly or implicitly. Stochastic Language Model (SLM) performs its recognition by computing the possibilities of the word occurrences depending on large training data corpus (discussed in section 3). It is primarily based on the statistical analysis. However, it actually reflects the semantic constraints implicitly. For example, from an astronomic domain training data corpus, the possibility of "who discovers something" must be much higher than "which discovers something", which implies that it is much more possible for the word "discover" to occur after a person than after something. Therefore, stochastic language model also reflects the semantics indirectly.

As for grammar-based language model, usually, it defines more about the syntax than the semantics of the language features. But after the scrutiny, the clue of the semantics in grammar can be found. In the above example, the sample grammar may be more like:

q::= who discovers something

| what orbits something

than

q::= (who|what) (discovers|orbits) something

Here, the grammar-based language also induces embodies semantics (Frost, 2002).

Various techniques for use semantics in speech recognition are described in more detail in the remainder of this section.

7.2 Use of Large N, N-grams to try and capture semantic Information

In a traditional N-gram (discussed in section 3.1), the current word is predicted by the immediately previous N-1 words. This technique is based on the assumption that the relevant syntactic information lies in the immediate past. However, the fact is that some syntactic or semantic information does exist in the farther past. On the other hand, if use a larger N in an N-gram model is used, the free parameters will exponentially increased, which is too hard to control.

Huang *et al.* (1992) experimented with long-distance bigrams (the same principle can be applied to N-gram) with reduced number of free parameters. In the *distance-d bigram*, a word W_i is predicted by the word W_{i-d} (Huang *et al.*, 1992). The observation is that the recognition error has been reduced significantly, and the perplexity is low for d=1; and increases significantly for d=2,3,4 and 5; while remains at almost the same for d=6, 7, 8, 9, 10. Huang *et al.* (1992) made the conclusion that there is some relevant information, which is thinly spread across the history, in the distant past.

In (Bonafonte *et al.*,1996), the speech was decoded onto an intermediate representation in sequence, where the order of semantic units was the same as that of the words in the sentence. Also, the query was modeled as the semantic unit strings, which was suitable for N-gram to capture the semantic language.

Considering the fact that, in many languages (e.g. English), multiple words can be unified together and be treated as a single unit (phrase) in communication, Riccardi and Gorin (1998), Riccardi and Bangalore (1996) proposed "phrase-based language models" to better (over word-based language models) capture long spanning dependencies between words and without the exponential increase of parameters. They acquired the lexical features (phrases) from training data and the probability of the word sequence was computed from the process of entropy minimization over the training set and its length ranges from 1 to N. The phrase-based N-gram language model significantly outperforms a word-based language model (Riccardi and Bangalore, 1996).

7.3 Semantic Post-Processing of Output from Statistical Recognizer

At present, it is impossible to avoid errors in the earlier stage of speech recognition. Since the goal of eradicating the speech recognition errors is unpractical, many researchers are working on semantic post-processing techniques for error correction to further improve the recognition accuracy.

7.3.1 Post-processing to Choose Best Hypothesis

On account of its simplicity and efficiency, N-best search can be used in a post processing stage in the speech recognition to get better performance. Tran *et al.*, (1996) firstly constructed a recognition hypothesis word graph, and extracted N-best word sequences from the word graph. Combining with the language features, such as syntactic and/or semantic analysis, the N candidates can be re-scored with highly-reduced computational cost (Rayner *et al.*, 1994), and even many of the top N sentence hypotheses could have been eliminated before reaching the end with early syntactic and semantic analyses (Seneff *et al.*, 1995). Milward and Knight (2001) applied a class-based statistical language model to construct the word-hypothesis graph and then used the semantic knowledge which can be obtained by Spoken Language Translator (Rayner *et al.*, 2000c) to choose the hypothesis in the graph. Seneff *et al.* (1995) used an A* algorithm to search through the large hypothesis word graph, and Harper *et al.* (1992) used syntactic constraints and a Constraint Dependency Grammar (CDG) parser (discussed in section 7.6) to effectively prune the hypothesis word graph of the ungrammatical sentence hypotheses and limit the possible parses of the remaining sentences.

Stolcke *et al.* (1997) developed an algorithm to explicitly minimize the expected word errors for recognition hypotheses. The N-best lists tell the approximation of the posterior hypothesis probabilities. Then with respect to the posterior distribution, each hypothesis' expected word error is computed, and the hypothesis with the lowest error is chosen.

Ballim and Pallotta (2000) use domain knowledge to semantically constrain the hypothesis space. The architecture contains the following three modules: (1) a speech recognition system taking speech signals as input and providing N-best sequences in form of a lattice; (2) a stochastic syntactic analyzer (i.e. parser) extracting the k-best analyses; (3) a semantic module in charge of filling the frames required to query a database.

Current speech recognizers usually associate the input word with a lattice of word-hypotheses rather than a uniquely identified word. Taking into account the linguistic context, such as lexis and morphology, parts-of-speech, phrase structure, semantics and pragmatics, Atwell and Kevitt (1993) developed a language model to constrain the possible choices to the most linguistically plausible words. In (Atwell and Kevitt, 1993) (Atwell *et al.*, 1993), the linguistic knowledge sources include the Longman Dictionary of Contemporary English (LDOCE) semantic primitives, semantic tagging (semantic subject field markers), non-compositional phrase structure (syntactic phrase structure boundaries), wordtag n-grams, word-collocational preferences and the relationship between prosody and syntax. Resorting to the machine-readable dictionaries (e.g. the LDOCE) for the syntactic and semantic definition, (Atwell *et al.*, 1993) dealt with the word ambiguity by probabilistic ranking.

Stahl *et al.* (1997), Muller and Stahl (1998), Kawahara (1994) have described a speech understanding system, which has the architecture of sequential combination of a signal preprocessor, a stochastic-driven one-stage semantic decoder and a rule-based intention decoder. Goddeau (1993) proposed a probabilistic language model to integrate the local and long-distance language constraints into lexical-access search algorithms. The technique adopted the LR parser to map sentence prefixes into equivalence classes, which are further used to compute next word probabilities for speech recognition.

In (Stahl *et al.*, 1996), the semantic information was directly represented in the parse tree. This semantic tree structure consists of a finite number of semantic units (called semuns), each semun contains the semantic contribution of one significant word in the sentence. Then, an incremental technique, which integrated semantic, syntactic, acoustic-phonetic knowledge, and Viterbi-algorithm (Muller and Stahl, 1998), together with the chart-parsing technique and a top-down parsing strategy (Stahl *et al.*, 1996), was applied to achieve high efficiency and further the seamless interface between the speech recognition and understanding components.

The processing in (Seide *et al.*, 1996) can be sketched as follows: using an acoustic model and a word-unigram language model, the plausible word hypotheses are identified and scored. Then, a bigram is used to prune the word graph. Since all plausible alternative sentence hypotheses have been included in the word graph, every path through the graph represents a sentence hypothesis. Subsequently, an attributed stochastic grammar parses the word graph and assigns the language-model probability for every path (i.e., sentence hypothesis) through the information graph. Finally, taking account of the database goal and the consistency constraints, the most likely hypotheses are determined. The speciality of the technique not only lies on the consideration of the database goal and the consistency constraints, but also lies on the fact that N is not required to be known in advance. This technique computes the N best sentences one by one and discard those that are inconsistent or referring to invalid database entries.

7.3.2 Post-processing to Correct Errors

Ringger (2000), Ringer and Allen (1996), (1997) have investigated the use of statistical techniques and search algorithms for post-processing the output of a speech recognizer to correct errors. Soltau and Waibel (1998) considered the speaking style is more accentuated to disambiguate the original mistakes. Hauptmann *et al.* (1998) conducted experiments to assess the effect of words missing from the speech recognition vocabulary. Walker *et al.* (2000) developed a spoken dialogue system to allow some automatic error corrections by interacting with the user.

Loken-kim (1988) developed the Automatic Error Detection and Correction System (AutoDac), which is able to parse ill-formed sentences with a combination of left-to-right and right-to-left parsing; learn the history of recognition errors and utilize this information to subsequently recover from similar recognition errors later; and allow a user to manually correct any part of the recognized sentence. Combining automatic and manual error correction, a total of 142 out of 192 testing sentences were recovered (Loken-kim, 1988).

7.3.3 Post-processing to Modify System for Future Use

For the 10-best hypothesis lists on the 1001-unseen-utterence subset of the ATIS corpus, the best result of the experiments, which were explored by Rayner *et al.* (1994), gave a proportional reduction of 13% in the word error rate and 11% in the sentence error rate. In addition, the hypothesis reordering technique proposed by Rayner *et al.* (1994) is automatically trainable, acquiring information from both positive and negative examples.

In the voice-interactive natural language system, Fink (1984) added a special module, called an expectation system, to aid the speech-recognition process. Its basic idea is like this: the expectation system accepts the user's utterances, and studies the repetition and patterns in the dialogues to create a more general dialogue, then uses this generalized dialogue to correct errors in the future sentences by prediction. The results showed that the average sentence error rate was decreased from 53% to less than 8%. Furthermore, it can be concluded that the expectation system is capable of predicting what might happen in any situation that tends to be repeated.

As any spoken dialogue system aims to fulfill some goals in a particular domain, the user operates the system with the intentions in some specific directions. For example, the user enters into an automatic exchange board system with the intention of connecting to some person specified by name. Based on this observation, Seide *et al.* (1996) designed a system to catch the user's dialogue goals and restrict the discourse to a narrow application domain, hence further constrain the variety of possible user reactions and improve future recognition accuracy.

7.4 Grouping of Terminals/ Words/ Lexicon According to Meaning

Demetriou *et al.* (2000) developed a semantic model of language using an online dictionary, Longman Dictionary of Contemporary English (LDOCE), to acquire lexical semantic knowledge for speech-recognition modeling. The modeling of the semantic knowledge is based on the association between two words from their meanings in the dictionary, then compute how much do the meanings (sets of semantic primitives or concepts that are used to define the words in the dictionary) overlap or linkage (semantics). Furthermore, the semantic association measure for two words can be extended for computing the semantic association of longer word sequences in texts, such as phrases, sentences or paragraphs.

The experiments conducted by Demetriou *et al.* (2000) show that this model is able to capture the potential semantic dependencies between the words in texts, and reduce the language ambiguity by a considerable factor, and improve the word-recognition rates in "noisy-channel" applications. Therefore, Demetriou *et al.* (2000) stated that limited or incomplete knowledge from lexical resources such as Machine Readable Dictionaries (MRDs) can contribute to domain-independent language modeling.

7.5 Integrating Semantics into the Grammar to Better Direct the Recognizer – Unification Grammars

Belonging to the augmented or annotated Context-Free Grammars, Unification Grammar is more expressive and more concise than the traditional CFG. Unification Grammar is a higher-level formalism of Context-Free Grammar, which is obtained by applying some restriction properties to the CFG. With more constraints unified to the grammar, Unification Grammar helps reduce the system's perplexity. To better understand the Unification Grammar, refer to the following example extracted from (Moore, 1999):

S: [tensed=yes] → NP: [person=P, num= N] VP: [tensed=yes, person=P, num=N]

The distinction from traditional Context-Free Grammar (CFG) is the notion of the feature constraints (such as, person=P, num=N). The consequent unique power lies in the fact that the Unification Grammar constrains the features to a variable instead of specific values. The subsequent advantage can be seen from the above example that Unification Grammar guarantees that the person and num features of Noun Phrase (NP) and Verb Phrase (VP) must agree with each other, avoiding enumerating their respective features (person = first, num = singular, and so on).

A Unification Grammar can be compiled into a Context-Free Grammar by eliminating left recursion (detail instantiating algorithms can be found in Moore, 1999), which can be fed directly into the Nuance Toolkit's language model compiler (Rayner *et al.*, 2000a).

So far, Unification Grammars have been widely used to successfully build substantial general grammars for Natural Language Processing (NLP). Gemini, a natural language understanding system developed for spoken language applications (Dowding *et al.*, 1993), is such a successful Unification-Grammars-Based system (Moore *et al.*, 1997), where the Unification Grammars are initially specified and later compiled into standard CFG descriptions by a model compiler. In Gemini system, firstly all possible features in the grammar rules and lexicon entries are enumerated; then, each rule and entry in the original Unification Grammar are transformed into a set of rules in the derived CFG (Rayner *et al.*, 2000b).

Many significant applications, such as CommandTalk (Goldwater *et al.*, 2000) (Stent *et al.*, 1999), are built on the Gemini system. CommandTalk is a spoken-language interface to the battle-field simulator, which allows military commanders to interact with simulated forces in a manner similar to the way they would command actual forces. The unification-based grammar, based on Gemini, in CommandTalk brought twofold effects (Goldwater *et al.*, 2000): the negative is the less coverage than a statistical model; while the positive is the elimination of the usual discrepancy in coverage between the recognizer and the natural language parser.

Based on the Unification Grammars, Dowding *et al.* (1994) introduced an efficient bottom-up parser that interleaved syntactic and semantic structure building. It applied the limited left-context constraints to reduce local syntactic ambiguity, and the local semantic ambiguity was alleviated by deferred sortal-constraint application. The primary advantage of this parser lies in the dramatic reductions in both numbers of chart edges and total parsing time without sacrificing completeness.

Generally, the grammar-based language model suffers from the potential disadvantage of over-constraint, which means the grammar might exclude some reasonable utterances. To alleviate this problem, the grammar in CommandTalk was broadened to allow the word insertions and deletions (Goldwater *et al.*, 2000) if the inserted and deleted words contribute little to the meaning of the sentence.

Buo and Waibel (1996) introduced a feature structure parser, called FeasPar system (discussed in section 6.5), which is able to learn parsing spontaneous speech automatically with minor hand labeling, to challenge the unification approaches' drawback of requiring hand-designed lexicon and grammar rules, and rigidity of the grammar encountering ungrammaticality and deviations from linguistic rules.

7.6 Integrating Semantics into the Grammar to Better Direct the Recognizer – Dependency Grammars

A Dependency Grammar (DG) incorporates semantic constraints for large-vocabulary continuous-speech recognition (Takezawa *et al.*, 1991). Dependency Grammar describes sentences in terms of asymmetric pairwise relationships among words (Rosenfeld, 2000a), which means that each word in the sentence is dependent upon one other word (called its *head* or *parent*) except the *root* that serves as the head of the entire sentence.

7.6.1 **Constraint Dependency Grammar (CDG)** (Harper *et al.*, 2000), (Harper *et al.*, 1999a), (Harper *et al.*, 1995).

Harper (1999a) states that the Constraint Dependency Grammar (CDG) was first proposed by Maruyama in 1990. It is a constraint-based grammatical formalism with a weak generative capacity beyond Context-Free Grammars (CFG) and supports a very flexible parsing algorithm for working with feature grammars (Harper, 1999a). CDG uses constraints to determine the grammatical dependencies for a sentence. In CDG, the parsing rules are defined as constraints and the solutions are parses, thus, the parsing procedure has been transformed into the constraint satisfaction procedure.

A Constraint Dependency Grammar (CDG) (Harper *et al.*, 2000) (Harper *et al.*, 1999a) (Harper *et al.*, 1995) consists of four finite sets: Σ, R, L and C. Σ includes lexical categories (for example, noun, verb); R contains role types $\{r_1,\ldots, r_p\}$, L constitutes of a group of labels $\{l_1,\ldots,l_q\}$and C is a finite set of constraints, which determine the grammatical dependencies for a sentence. For example, an n-symbol sentence s = $w_1w_2\ldots w_n$ is an element of Σ^*, and each word $w_i \in \Sigma$. A *role* is a variable with the *role* values, and each label in L indicates a different syntactic function. To successfully generate a sentence, there must exist an assignment A that maps a *role* value to each of the n*p roles for *s* such that C is satisfied. If there is more than one assignment of *role* values satisfies C, ambiguity takes place. If the number of variables in a subformula of C is one or two, the subformula is called a *unary constraint* or *binary constraint* respectively. The max number of variables contained in a subformula of C is called the *arity* parameter for a CDG.

Compared with Context-Free Grammars (CFG), the Constraint Dependency Grammar (CDG) is more flexible and more tractable, but less expressive (Harper, 1999a). CDG holds the advantage of supporting a very flexible parsing algorithm for feature grammars. However, its disadvantage is the $O(n^4)$ parsing time complexity (Harper *et al.*, 1999a). Harper *et al.* (1999b) loosely integrated a CDG parser with an HMM word recognizer to reduce the parsing time.

7.6.2 Enhanced Constraint Dependency Grammar

Harper *et al.* (1999a) pointed out two difficulties existing in the original CDG parsing mentioned above: (1) the CDG is difficult to analyze the sentence where the lexical categories are multiple (for example, the word *can* belongs to noun, verb, and modal categories); (2) or the category has multiple feature values (for example, the word *the* as a determiner can modify nouns of both third person singular and third person plural). The second difficulty is its slowness (the time complexity is $O(n^4)$).

Harper *et al.* (1995), Helzerman *et al.* (1996) proposed extensions to the Constraint Dependency Grammar to address the first difficulty by allowing the simultaneous parsing of alternative sentences from lexical or feature ambiguity. The original CDG creates and applies all the possible role values for all roles at one time, which uses much computation time. Nevertheless, Harper *et al.* (1999a) adopted an Enhanced CDG to reduce the computation time by applying the feature constraints in groups and eliminating the ungrammatical role values as many as possible before preparing for another feature. The time complexity for Enhanced CDG has been improved from $O(n^4)$ to $O(n^2)$ (Harper *et al.*, 1999a).

7.6.3 Corpus-Induced Constraint Dependency Grammar (Harper *et al.*, 2000)

Corpus-Induced Constraint Dependency Grammar means extracting CDG constraints from a domain-specific corpus of sentences. Harper *et al.* (2000) conducted an experiment to test its plausibility and benefits. The result is that the Corpus-Induced Constraint Dependency Grammar significantly improved recognition accuracy over the conventional CDG.

7.6.4 The TINA Framework

TINA is a trainable natural-language model (Chung and Seneff, 1998) developed by Seneff *et al.* (1995). The base of TINA is an augmented Context-Free Grammar, which contains a set of features to enforce syntactic and semantic constraints, and a trace mechanism to handle movement phenomena. Both features and unification apply are associated with category, not on the context-free rule. Terminal words with feature values unify them with the feature pattern that is delivered to them by their parent/left sibling during the parse process (Seneff *et al.*, 1995). Constraints, such as subject-verb agreement, and semantic features, are very important syntactic features for constraining gaps.

The hand-coded grammar rules are automatically broken apart into a set of trigram sibling-sibling transition probabilities to capture both spatial (parent) and temporal (left-sibling) conditioning context. The top-level rules of the grammar are very flexible, for they permit the parser to derive a partial parse (Seneff *et al.*, 1995). A sentence can be fully parsed, also, it may be parsed by skipping one or more non-content or unknown words. The probabilities are calculated by tabulating counts in the parse trees, which are automatically built up from the training corpus. Similarly, the top-level transition probabilities are based on the tabulations on counts for the top-level transitions. In this way, full-parsed and partial-parse theories can compete side-by-side according to their probabilities (Seneff *et al.*, 1995). Seneff *et al.* (1995) showed the favorable recognition performance of TINA over a traditional word class 4-gram language model.

7.6.5 Techniques Related to Underspecified Semantic Representation

Investigating the ambiguity existing in a compact "underspecified semantic representation" (which means there are multiple meaning options for one sentence instead of a specific one) for sentences, Dorre (1997), and Milward and Knight (2001) proposed a method which constructs the compact semantic representation from input syntactic parse forests and constraint-based semantic construction rules. Milward and Knight (2001) state that this approach can improve keyword- or phrase- spotting approaches, because it can avoid many pitfalls of "over-early commitment" (e.g. to longest fragments) existing in many grammar-based systems. Dorre (1997) has fully implemented the algorithm with time complexity of $O(n^4 \log(n))$ with respect to sentence length.

The DELTA project at Tilburg University is about semantic and pragmatic interpretation of utterances in human-computer natural-language information dialogues (Bunt, 1995). It uses context-independent versus context-dependent aspects of semantic interpretation. The interpretation process calls for underspecified semantic representations, which can be further specified as contextual constraints. Bunt (1995) shows several instances of developing such representations for a variety of cases of ambiguity and vagueness.

7.7 Integrating Semantics into the Grammar to Better Direct the Recognizer- Direct Encoding of Semantics as Syntax Rules

Appropriate use of constraints can restrict the search space of input utterances, and reduce the perplexity of the speech recognition (Murveit and Moore, 1990), thereby, improving the speech recognition accuracy. Usually, recognizers return a couple of guesses of the input utterances, then, use semantic post-processing techniques to help find the most plausible guesses.

Moreover, an alternative approach is to encode the semantic rules directly in the syntax of the grammar (Frost, 2002). This technique is based on the observation that some syntactically correct utterances may be semantically wrong. Frost (2002) presented the example that the sentence "which man orbits kuiper" may be accepted by a simple grammar for its correct syntax, but in the domain used as example, people cannot orbit other people, thus it is semantically incorrect. The simple syntax that accepts the above example sentence might be as follows:

question ::= "which" nounphrase verbphrase

If we replace it with the following:

question ::= "which" animatenounphrase animateverbphrase

| "which" inanimatenounphrase inanimateverbphrase

then the semantically incorrect utterance above is not accepted as a possible utterance by the recognizer, hence the speech recognition accuracy has been improved.

The primary advantage of this technique is an improvement in speech recognition accuracy without unnaturally restricting the input utterances. However, this technique has the disadvantage that the increase of complexity and the size of the grammar by encoding semantic rules in the syntax make the system difficult to maintain. This can be overcome to some extent by combining this technique with the use of hyperlinks to create a Speech Web of speech-accessible objects, and further improve recognition accuracy by moving between domain-dependent grammars (Frost, 2002).

7.8 Integrating Semantics in Statistical Language Modeling

Coccaro and Jurafsky (1998) and Chappelier *et al.* (1999) introduced a number of techniques to help integrate semantic knowledge with N-gram language models for automatic speech recognition. The techniques in (Coccaro and Jurafsky, 1998) are able to integrate Latent Semantic Analysis (LSA), a word-similarity algorithm based on word co-occurrence information, with N-gram models. LSA can tell the presence of words in the domain of the text, but cannot tell their exact location. Since the N-gram model has the ability to work out the word location, it can complement the LSA model by filling in the missing information.

In addition, LSA performs better in predicting coherent content words than frequent words in a low dynamic range. However, the linear combination of LSA and N-gram has the poor performance. To address this problem, Coccaro and Jurafsky (1998) modified the dynamic range, applied a per-word confidence metric, and used geometric rather than linear combinations with N-grams, and the result is a more robust language model with a lower perplexity on a Wall Street Journal Test-set than a baseline N-gram model.

7.9 Semantics in Topics - High Level Semantic Domains

The frequently-used N-gram model suffers from a lack of long-term information for the reason that the next word is predicted by the preceding N-1 words (typically $2 \leqslant N \leqslant 4$).

In addition, the disorganization of the words in a large vocabulary constitutes the large number of intractable parameters (which is discussed in section 2). To capture the relationships between the words and extract the topics can not only build up the long-term context information about the topic (Mahajan *et al.*, 1999), but also dramatically reduces the dimensions (parameters), and consequently improves the performance of speech recognition.

Reynar (1998) proposed a technique to segment different topics in one document. What Rosenfeld (2000a) did was to firstly tabulated the occurrence of every word in the document; then, reduce the large matrix by Singular Value Decomposition to a lower dimension. Then, the correlations between words were captured in the smaller matrix and consequently the new document, structured by topics, was obtained. It was reported in (Rosenfeld, 2000a) that combining this adaptation with an N-gram could reduce the perplexity and obtain lower recognition errors. Using the experiments on the Wall Street Journal text corpus, Mahajan *et al.*, (1999) demonstrates the effectiveness of this technique of perplexity reduction by 37% compared to the baseline language models.

7.10 Semantic Networks

A semantic network is another powerful technique to assist in speech recognition, which is usually represented in the form of a directed graph where nodes represent word senses and links represent the types of conceptual relationships. A traversal through the network defines a sentence. Semantic networks have been used for the construction of sentence hypotheses guided by concept-relation judgements of content words (Demetriou and Atwell, 1994a).

Demetriou and Atwell (1994b) developed a large-vocabulary semantic network by systematically using semantic information on nouns and verbs from the Longman Dictionary of Contemporary English (LDOCE) using pattern-matching rules. Using semantic networks, Ahlrichs *et al.* (1999) proposed a knowledge-based approach for spoken dialsogue. Dupont (1993), Jurafsky *et al.* (1995) and Fischer *et al.*, 1999) built a semantic network as a stochastic finite-state network (called a Stochastic Context-Free

Grammar (SCFG)), where grammars, probabilities and other linguistic constraints can be added to the word connections (Savage-Carmona *et al.*, 1995) (Dupont, 1993) to minimize the perplexity (the average word branching factor).

In PHOENIX, a robust semantic parser is used in the speech recognizer of SPHINX-II, which was developed in Carnegie Mellon University, the semantic relations are represented by concept frames and the patterns for semantic fragments are represented in Recursive Transition Networks (RTNs) (Kaiser *et al.*, 1999). The patterns are used to fill the slots in semantic frames (Ward and Young, 1993). Out-of-grammar words that occur between slots can be skipped and the resulting partial parses (only some slots in the frame have been filled) are returned.

In this architecture, word strings with the same meaning are determined from the network, which is generated from the semantic grammar. Ward and Issar (1994) compiled the grammars into many small "phrase level" nets, instead of a single large network (which is common in other standard RTNs). For example, the words representing departure and arrival cities will respectively be assigned to two different networks. Thus, the utterance "I want to see flights from Boston to Denver after 5pm" would be interpreted as the concept sequence [list] [select_field] [from_location] [to_location] [depart_time_range], where the concept sequences are specified by RTN (Ward and Young, 1993).

The semantic hierarchy contributes to restrictions in the way that the inheritance of the networks can help generalize role fillers (Demetriou and Atwell, 1994a). Also taking advantage of the finite-state language constraints (Murveit and Moore, 1990), various search algorithms can be used here, such as a beam search and A* search algorithms (Kaiser *et al.*, 1999). Also, it can be combined with context-free grammars and word bigram methods (Ward and Young, 1993) (Ward and Issar, 1994). The "concept-spotting" approach in PHOENIX is considerably robust and has been widely used in spoken language information systems (Kaiser *et al.*, 1999). Dupont (1993) applied a beam-pruning technique and Savage-Carmona *et al.* (1995) used a Viterbi algorithm to further limit the search space growth, consequently, the complexity of the network expansion decreased dramatically.

In addition, Jurafsky *et al.*, (1995) mentioned another advantage of the above SCFG that it included the language model at the frame level of the acoustic decoding, hence significantly improved the recognition accuracy of decreasing the word error rate from 34.6% (bigram) to 29.6% (SCFG).

However, even though SCFGs are good at modeling long-term relations and limited-domain tasks of low perplexity, it may be intractable if the lexicon size or the language model is too large due to the difficulty of the computation of word transition probabilities for complex real tasks (Benedi and Sanchez, 2000). The worst is that if there exists self-embedded recursion in the language model, it will result in the corresponding network with infinite states and transitions.

8 OTHER APPROACHES WHICH INTEGRATE NATURAL-LANGUAGE FEATURES INTO THE RECOGNITION PROCESS

8. 1 Speech Webs

It is not easy to construct speech interfaces to large knowledge bases for the reason that large knowledge source require large and complicated grammars, which are not trivial to implement and which have high perplexity and therefore low accuracy (Frost and Chitte, 1999). Instead, Frost and Chitte (1999) proposes a new approach of dividing large knowledge sources into several smaller domain-based knowledge bases, called “sihlos”, and using relatively narrow grammars in each individual sihlo. Only when the sihlo is visited, are its grammar and other related properties downloaded to respond to the user. With the decrease of the scope of the knowledge source, the query language is shrunk, which can significantly contribute to the speech recognition accuracy.

The user can move from sihlo to sihlo by “speaking” hyperlinks. Under such a schema, the semantic constraints in syntax (the techniques are discussed in section 7.7) of each sihlo have to be considered for the fact that some semantic constraints are appropriate in one context and might be inappropriate in another one. Frost (2002) gave the example that the constraint that "people cannot orbit anything" is appropriate in the "solarman object", while not appropriate in the object about astronauts.

In addition, a spoken-dialogue system may perform differently for different users and even the same user during different dialogues. To solve this problem, Litman and Pan (2000) (1999) developed TOOT, a spoken-dialogue system for retrieving train schedule on the web which predicts a user's behaviour in a particular dialogue process. According to such predictions whether he/she is having speech-recognition problems, TOOT will automatically adapt its dialogue strategies.

8.2 Large Vocabulary Related Techniques

Large vocabularies have been one of the major challenges for speech-recognition researchers (discussed in section 2). So far, a lot of work has been conducted on this point, such as the dependency grammars (discussed in section 7.6), semantics in topics – high-level semantic domains (discussed in section 7.9), and semantic networks (discussed in section 7.10) might be possible solutions to this problem. The following are some other techniques related to this problem:

- Miller (1988) describes a CFG-based syntactic component for large vocabulary speech recognition as the language model. Benedi and Sanchez (2000) proposed an approach, which is capable of capturing both local and long-term relations between words and syntactical structures (details are discussed in section 5).
- Bellegarda (1998) proposed a new framework of integrating both local and global constraints for multi-span statistical language modeling. Local constraints are captured via language modeling, while global constraints are taken into account through latent semantic analysis. The integration of these two paradigms results in several families of multi-span language for large vocabulary speech recognition.
- Moody (1988) conducted experiments to test the effects of restricted vocabulary size

in speech-recognition and natural language understanding process, and the results show the advantages of the restricted vocabulary over unrestricted vocabulary in the ways that the shorter completing time, the fewer word usage, and better recognition accuracy is achieved, especially in goal directed utterances.

Valverde-Albacete and Pardo (1996) presented a multi-level lexical-semantics based language-model design for guided integrated continuous-speech recognition to decrease the search space when the lexicon size grows. This approach consists of two mutually-recursive functions. Firstly, an auxiliary retrieval function is used to obtain lexicalized (already built) solutions to the problem, which are merged with the ones built by the second function. This second function describes the acoustical and semantic recognition process as a search problem, which is defined in the first function, and solved with the help of the A* strategy. A hierarchy of linguistic levels is used. And each level contains a particular meaning structure, a lexicon of lexicalized forms, the lexicalization probabilities, and a local lexical grammar describing how the semantic categories of the level can be built. This speech recognition architecture is tested a DARPA RM-like application by Valverde-Albacete and Pardo (1996).

8.3 Language Models for Languages Other Than English

Xu *et al.* (1988) integrated syntactic, semantic and vocabulary knowledge constraints into a linguistic processor to improve the performance of a Chinese speech-recognition system. One feature of this processor is that both sentences and phrases can become its speech input. In addition, some unique characteristics of Chinese language are taken into account.

9. OTHER SURVEYS ON THE USE OF NATURAL-LANGUAGE FEATURES IN SPEECH RECOGNITION

A great deal of work has been carried out on the use of Natural-Language features in speech recognition. Correspondingly, a number of other surveys have been done on this topic.

- Rosenfeld (2000) primarily focused on Statistical Language Model (SLM) techniques, such as N-grams, Class N-gram, Decision Tree Models, and Adaptive Models. Also, in Rosenfeld's (2000a) opinion, the Probabilistic Dependency Grammars belong to the promising current directions. In addition, Rosenfeld (2000a) mentioned that the World Wide Web is an efficient resource for obtaining the training data.
- Demetriou and Atwell (1994a) summarized the current semantic methods in speech recognition and understanding research and classified the approaches into six main categories: (1) Semantic networks, which are discussed in section 7.10. (2) Semantic grammars, which are discussed in sections 4.4, 7.5, 7.6 and 7.7. (3) Caseframe approaches, in which, the semantic constraints are expressed in the form of caseframes. These methods can be used for the production of sentence hypotheses from a word lattice and the choice of the most likely one, or for filling gaps of missing words or for post-processing correction, as well as for making word predictions during recognition. (4) Statistical approaches, which are discussed in section 3. (5) Unification-based approaches, which are discussed in section 7.5. (6) In neural networks, processing elements or nodes are connected by links with variable weights, which are adapted from training data and are continuously modified during use.
- Based on the observation that the successful SLM techniques use very little language knowledge, Rosenfeld (2000b) reviewed the extent to which aspects of natural language are captured in current models. Rosenfeld (2000b) mentioned three approaches of integrating syntax into language modeling. (1) *Probabilistic Context-Free Grammars (PCFG)* (discussed in section 4.2); (2) *Probabilistic link grammars*, which use lexicalized grammar formalism. Specific link grammars are constructed by hand. Based on the link grammar, a word can be predicted from any pair of adjacent words that precede it in the sentence. A specialized form of the grammar, called a

Grammatical trigram, has achieved a modest yet consistent perplexity improvement over the current trigram. (3) In *structured language model*, the next word is predicted based on a set of linguistic equivalence classification of the history.

Rosenfeld (2000b) also introduced four ways to capture topic coherence. (i) *Model interpolation*. The training data were partitioned into multiple sets by topic(s). Then, a separate topic-specific language model is created on each such set, and the interpolations between the various models takes place at the word level. This method achieves moderate yet consistent reductions in perplexity and speech recognition error rates. (ii) The N-gram *cache*, which has been implemented in many systems with a modest reduction in word recognition error rate, is easy to implement and capture word auto-correlations. (iii) *Word triggers* are the outcome of the generalization of the cache idea. (iv) The *dimensionality reduction* of the topic space, which can be achieved by Singular Value Decomposition (SVD), improves the modeling individual word correlations.

According to Rosenfeld (2000b), it is almost impossible to think about linguistic aspects of sentences, such as their grammar syntax, semantics or pragmatics, and say nothing of encoding in a conditional framework. Rosenfeld (2000b) proposed the exponential model, which directly models the probability of an entire sentence or utterance. In this model, each sentence or utterance is treated as a bag of features, which are arbitrary computable properties of the sentence. Furthermore, the unified structure of the model makes it possible that any linguistic theory can be incorporated without any change to the model itself.

Rosenfeld (2000b) has discussed the reason for the difficulty of integrating linguistic features with statistical language models as the following: (1) linguistic theories and statistical models have different goals. The former deal with existence, whereas the latter deal with prevalence. (2) lack of a general framework. (3) mental straight-jacket of the conditional formulation. (4) Impoverished priors. A prior is supposed to capture everything that is known about the domain before any data are observed.

However, the language (e.g. English) has such a large parameter space that any feasible amount of training data is insufficient.

- Lavie (1996) mentioned the following techniques:

Carbonell and Hayes (1984) suggested a case-frame approach to handle the extra-grammaticality. After examining the main semantic concept of the sentence, the semantic interpretation of the input is obtained. Then, search the sentence for components that instantiate the semantic frames that are associated with the main concept. This approach is flexible to the order of the semantic frames to the input, but it is domain dependent and hard to capture syntactic and other grammatical knowledge.

McDonald (1992) described an approach based on chart parsing. Semantic grammars are used to combine the lower level phrases into phrases that represent semantic concepts, and then applied to a coherent analysis by the conceptual analyzer, which allowed gaps of unanalyzed segments of text between the combined phrases. The system unified bottom-up syntactic parsing with top-down conceptual expectation-driven parsing into a flexible multi-layer parser. Thus comes the drawback of complexity.

Menzel (1995) suggested a unified approach by using the constraint grammar formalism to express syntactic, semantic and pragmatic linguistic constraints. Thus, the violation of the constraints is regarded as penalties, and the importance of satisfying a constraint can be modeled via penalty weights. Then the minimal penalty means satisfying the constraints best. Unfortunately, this approach has not been fully implemented in a large application.

10. CONCLUSION

With the growing interest and demand for the human-machine interaction, more and more work concerning speech-recognition has been carried out over the past decades.

Chappelier *et al.* (1999) has stated that, over the past decade, speech-recognition technology has made significant progress: with twofold reduction every two years, in word-recognition errors (Rabiner *et al.*, 1996), and the emergence of high-performance language systems. A variety of approaches have been proposed to address speech-recognition issues, such as the stochastic (statistical) techniques, grammar-based techniques, combined N-gram and grammar-based techniques, techniques integrated with linguistic features, and other approaches. Furthermore, it has been widely accepted that language features are playing significant roles to achieve high accuracy in speech recognition (Harper *et al.*, 2000), (Moore, 1999), (Seneff *et al.*, 1995), (Hermannsdottir, 1996), (Takezawa *et al.*, 1991). However, there are still a lot of challenges on the way of developing high-accuracy, and user-friendly speech-recognition technologies (Glass, 1999).

This survey also indicates that Rosenfeld is the person who is making significant contribution to the integramion of grammar-based and stochastical-based techniques.

Appendix B: Computation of Language Size in Detail

Note: superscripts are used to denote the obtained size of the sub-language defined by the expressions; the following comments (starting with "//") denote the computation used to calculate the size.

Figure Appendix B (1): language-size computation of semantic grammar

```
/* semantics_gram_ext1.gram */
grammar semantics_gram_ext1;
public <s>^2706249417898 = <linkingvb>^4 <termphrase_verbphrase>^455684689185
            | is <pnoun>^121 <pnoun>^121
            | is <pnoun>^121 ( a|an )^2 <nouncla>^108
            | is <pnoun>^121 ( a|an )^2 <nouncla>^108 or ( a|an )^2 <nouncla>^108
            | <quest1>^3 <sent>^294403057132
            | ( who ) <animate_verbph>^8772934
            | ( what ) <inanimate_verbph>^3837429
            | ( which | how many ) <nouncla_verbph>^126895596
            | ( which | how many ) <nouncla_verbph_other>^156297624
            | <simple>^26;
// 4*455684689185 + 121*121 + 121*2*108 + 121*2*108*2*108 + 3*294403057132+ 8772934 +
// + 3837429 + 126895596 + 156297624 + 26 =
// = 1822738756740 + 14641 + 26136 + 5645376 +820116752331 + 295803609
// = 2706249417898 = 2.70 * 10^12
<simple>^26 = | ask them to be quite
        | please introduce yourself
        | hello there
        | goodbye
        | goodbye solar man
        | fine thanks
        | thanks
        | thanks solar man
        | yes please
        | what is your name
        | who are you
        | where do youlive
        | what do youknow
        | how old are you
        | what is your favorite band
        | who is the vice president at the university of windsor
        | who is the dean of science at the university of windsor
        | tell me a poem
        | know any poems
        | tell me a joke
        | know any jokes
        | who is judy
        | can i talk to judy
```

| can i talk to solar man
| who is monty
| can i talk to monty ;
<termphrase_verbphrase>^{455684689185} = <nonhuman_termph_planet>^{6555} <transvb_by_termph>^{3447441}
| <nonhuman_termph_moon>^{14196} <animate_transvb>^{6} by <human_termph>^{25651}
| <nonhuman_termph_other>^{1069453} <animate_transvb>^{6} by <human_termph>^{25651}
| <nonhuman_termph_other>^{1069453} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_planet>^{6555}
| <nonhuman_termph_other>^{1069453} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_moon>^{14196} ;
// 6555*3447441+ 14196*6*25651+ 1069453*6*25651 + 1069453*6*2*6555 + 1069453*6*2*14196
// = 22597975755 + 2184849576 + 164595233418 + 84123172980 + 182183457456 = 455684689185
<transvb_by_termph>^{3447441} = <animate_transvb>^{6} by <human_termph>^{25651}
| <inanimate_transvb>^{6} by <nonhuman_termph_moon>^{14196}
| <inanimate_transvb_other>^{3} by <nonhuman_termph_other>^{1069453} ;
// 6 * 25651 + 6 * 14196 + 3 * 1069453 = 153906 + 85176 + 3208359 = 3447441
<sent>^{294403057132} = <human_termph>^{25651} <animate_verbph>^{8772934}
| <nonhuman_termph_moon>^{14196} <inanimate_verbph_active>^{39337}
| <nonhuman_termph_planet>^{6555} <inanimate_verbph_passive>^{340717}
| <nonhuman_termph_moon>^{14196} <inanimate_verbph_active_other>^{3208361}
| <nonhuman_termph_planet>^{6555} <inanimate_verbph_active_other>^{3208361};
// 25651*8772934 + 14196*39337 + 6555*340717 + 14196 * 3208361 + 6555 *3208361
// = 225034530034 +558428052 + 2233399935 + 45545892756 +21030806355
// = 294403057132
<nouncla_verbph>^{126895596} = <human_nouncla>^{12} <animate_verbph>^{8772934}
| <nonhuman_nouncla_moon>^{6} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_planet>^{6} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_moon>^{6} <inanimate_verbph_active>^{39337}
| <nonhuman_nouncla_planet>^{6} <inanimate_verbph_passive>^{340717} ;
//12*8772934 + 6*1611672 + 6*1611672 +6*39337 + 6*340717 =
// = 105275208 + 9670032 + 9670032 + 236022 + 2044302 =126895596
<nouncla_verbph_other>^{156297624} = <nonhuman_nouncla_other>^{84} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_other>^{84} <inanimate_verbph_passive_other>^{249014};
// 84 * 1611672 + 84*249014 = 135380448 +20917176 = 156297624
<inanimate_verbph>^{3837429} = <inanimate_verbph_active>^{39337}
| <inanimate_verbph_passive>^{340717}
| <inanimate_verbph_active_other>^{3208361}
| <inanimate_verbph_passive_other>^{249014};
// 39337 + 340717 + 3208361 + 249014 = 3837429
<human_stermph>^{113} = <human_pnoun>^{17}
| <human_detph>^{96} ; 17+96 =113
<nonhuman_stermph_planet>^{57} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_detph_planet>^{48} ; // 9 + 48 = 57
<nonhuman_stermph_moon>^{84} = <nonhuman_pnoun_moon>^{36}
| <nonhuman_detph_moon>^{48} ; // 36+48 =84
<nonhuman_stermph_other>^{731} = <nonhuman_pnoun_other>^{59}
| <nonhuman_detph_other>^{672} ; // 59 + 672 =731
<human_termph>^{25651} = <human_stermph>^{113}
| <human_stermph>^{113} (and | or) <human_stermph>^{113} ; //113+113*2*113=25651
<nonhuman_termph_planet>^{6555} = <nonhuman_stermph_planet>^{57}
| <nonhuman_stermph_planet>^{57} (and | or) <nonhuman_stermph_planet>^{57} ;
// 57 + (57*2*57) = 6555

<nonhuman_termph_moon>^{14196} = <nonhuman_stermph_moon>^{84}
| <nonhuman_stermph_moon>^{84} (and | or) <nonhuman_stermph_moon>^{84} ;
// 84 + 84*2*84 = 14196
<nonhuman_termph_other>^{1069453} = <nonhuman_stermph_other>^{731}
| <nonhuman_stermph_other>^{731} (and | or) <nonhuman_stermph_other>^{731};
// 731 + 731*2*731 = 1069453
<animate_verbph>^{8772934} = <animate_transvbph>^{8772934} ;
<inanimate_verbph_active>^{39337} = <inanimate_transvbph_active>^{39330}
| <intransvb>^{7} ; // 39330 + 7 = 39337
<inanimate_verbph_passive>^{340717} = <inanimate_transvbph_passive>^{340704}
| <intransvb>^{7}
| <inanimate_transvb>^{6} sun ; // 340704 + 7 + 6 = 340717
<inanimate_verbph_active_other>^{3208361} = <inanimate_transvbph_active_other>^{3208359}
| <intransvb_other>^{2}; // 3208359 + 2 =3208361
<inanimate_verbph_passive_other>^{249014} = <inanimate_transvbph_passive_other>^{249012}
| <intransvb_other>^{2}; // 249012 + 2 = 249014
<animate_verbph_passive>^{1611672} = <linkingvb>^{4} <animate_transvb>^{6} by <human_termph>^{25651} |
<linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2} <nonhuman_termph_planet>^{6555} |
<linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2} <nonhuman_termph_moon>^{14196} ;
// 4*6*25651 + 4*6*2*6555 + 4*6*2*14196 = 615624 + 314640 + 681408 = 1611672
<animate_transvbph>^{8772934} = <animate_transvb>^{6} (<nonhuman_termph_planet>^{6555}
| <nonhuman_termph_moon>^{14196}
| <nonhuman_termph_other>^{1069453})
| <animate_transvb_other>^{2} (<human_termph>^{25651}
| <nonhuman_termph_planet>^{6555}
| <nonhuman_termph_moon>^{14196}
| <nonhuman_termph_other>^{1069453});
//6*(6555+14196+1069453)+2*(25651+6555+14196+1069453)=6541224+2231710=8772934
<inanimate_transvbph_active>^{39330} = <inanimate_transvb>^{6} <nonhuman_termph_planet>^{6555} ;
// 6 * 6555 = 39330
<inanimate_transvbph_passive>^{340704} = <linkingvb>^{4} <inanimate_transvb>^{6} by
<nonhuman_termph_moon>^{14196} ; // 4 * 6 * 14196 =340704

<inanimate_transvbph_active_other>^{3208359} = <inanimate_transvb_other>^{3}
<nonhuman_termph_other>^{1069453}; // 3 * 1069453 = 3208359
<inanimate_transvbph_passive_other>^{249012} = <linkingvb>^{4} <inanimate_transvb_other>^{3} by
<nonhuman_termph_planet>^{6555}
| <linkingvb>^{4} <inanimate_transvb_other>^{3} by
<nonhuman_termph_moon>^{14196} ; // 4*3*6555+4*3*14196 = 249012
<human_detph>^{96} = <det>^{8} <human_nouncla>^{12} ; // 8*12 = 96
<nonhuman_detph_planet>^{48} = <det>^{8} <nonhuman_nouncla_planet>^{6}; // 8*6 = 48
<nonhuman_detph_moon>^{48} = <det>^{8} <nonhuman_nouncla_moon>^{6} ; //8*6 = 48
<nonhuman_detph_other>^{672} = <det>^{8} <nonhuman_nouncla_other>^{84} ; //8*84=672
<preposition>^{2} = on | in ;
<nouncla>^{108} = <human_nouncla>^{12} |
<nonhuman_nouncla_planet>^{6} |
<nonhuman_nouncla_moon>^{6} |
<nonhuman_nouncla_other>^{84}; //12+6+6+84=108
<human_nouncla>^{12} = <adj>^{2} <human_cnoun>^{4}
| <human_cnoun>^{4} ; //2*4+4 =12
<nonhuman_nouncla_planet>^{6} = <adj>^{2} <nonhuman_cnoun_planet>^{2}
| <nonhuman_cnoun_planet>^{2} ; // 2*2 +2 = 6

<nonhuman_nouncla_moon>^{6} = <adj>^{2} <nonhuman_cnoun_moon>^{2}
| <nonhuman_cnoun_moon>^{2} ; // 2*2+2=6
<nonhuman_nouncla_other>^{84} = <adj>^{2} <nonhuman_cnoun_other>^{28}
| <nonhuman_cnoun_other>^{28} ; //2*28+28=84
<human_cnoun>^{4} = man | men | person | people;
<nonhuman_cnoun_planet>^{2} = planet | planets ;
<nonhuman_cnoun_moon>^{2} = moon | moons;
<nonhuman_cnoun_other>^{28} = mountain | mountains | crater | craters | sea | seas | ocean | oceans |
chemical | chemicals | gas | gases | metal | metals | nonmetal | nonmetals |
country | countries | capital | capitals | city | cities | continent |continents|
river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find | finds | found ;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{121} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon> ^{36}
| <human_pnoun> ^{17}
| <nonhuman_pnoun_other>^{59}; //9+36+17+59=121
<nonhuman_pnoun_planet>^{9} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus | europa |
ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon | phobos |
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |
kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{59} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{33}; //20+6+33=59
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ; //6+9+5=20
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{33} = <country>^{6} | <capital>^{6} | <city>^{6} | <continent>^{6} | <ocean>^{4} | <river>^{3} |
<lake>^{1} | <mountain>^{1} ; 6+6+6+6+4+3+1+1=33
<country>^{6} = canada | china | England | France | Germany | united states;
<capital>^{6} = ottawa | Beijing | london | paris | berlin | washington;
<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;

<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;

Figure Appendix B (1): language-size computation of semantic grammar (Cont'd)

Figure Appendix B (2): language-size computation of syntactic grammar

/* syntax_gram_ext1.gram */
grammar syntax_gram_ext1 ;
public <s>^{3053116505638237} = <linkingvb>^{4} <termph>^{1941435} [<transvb>^{15} by] <termph>^{1941435}
| <linkingvb>^{4} <termph>^{1941435} [<transvb>^{15} <preposition>^{2}] <termph>^{1941435}
| <quest1>^{3} <sent>^{791525684027295}
| (who |what) ^{2} <verbph>^{407701357}
| (which | how many) ^{2} <nouncla>^{108}<verbph>^{407701357}
| <simple>^{26} ;
// 4*1941435*15*1941435 + 4*1941435*15*2*1941435 + 3*791525684027295 + 2*407701357 +
// + 2*108*407701357 +26 =
//=226150191553500 + 452300383107000 + 2374577052081885 + 815402714 + 88063493112+26
//= 3053116505638237 = 3.05 * 10^{15}
<simple>^{26} = | ask them to be quite
| please introduce yourself
| hello there
| goodbye
| goodbye solar man
| fine thanks
| thanks
| thanks solar man
| yes please
| what is your name
| who are you
| where do youlive
| what do youknow
| how old are you
| what is your favorite band
| who is the vice president at the university of windsor
| who is the dean of science at the university of windsor
| tell me a poem
| know any poems
| tell me a joke
| know any jokes
| who is judy
| can i talk to judy
| can i talk to solar man
| who is monty
| can i talk to monty ;
<sent>^{791525684027295} = <termph>^{1941435} <verbph>^{407701357}; // 407701357 * 1941435= 791525684027295
<stermph>^{985} = <pnoun>^{121} | <detph>^{864}; // 121+864 = 985
<termph>^{1941435} = <stermph>^{985} | <stermph>^{985} (and | or) ^{2} <stermph>^{985}; //985+985*2*985 = 1941435
<verbph>^{407701357} = <transvbph>^{407701350} | <intransvb>^{7}; //407701350 +7 =407701357

<transvbph>^{407701350} = (<transvb>^{15} | <linkingvb>^{4} <transvb>^{15} by) <termph>^{1941435} |
(<transvb>^{15} | <linkingvb>^{4} <transvb>^{15} <preposition>^{2}) <termph>^{1941435} ;
// (15 + 4*15) * 1941435 + (15+4*15*2)*1941435 = 145607625 + 262093725 =407701350
<detph>^{864} = <det>^{8} <nouncla>^{108}; 8*108 = 864
<nouncla>^{108} = <adj>^{2} <cnoun>^{36} | <cnoun>^{36}; // 2*36+36 = 108
<cnoun>^{36} = man | men | person | people | planet | planets | moon | moons | mountain | mountains |
crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal|
metals| nonmetal | nonmetals | country | countries | capital | capitals | city |cities |continent|
continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits| orbited | exist | exists ;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{121} = <pnoun_planet_moon_human>^{62}
| <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{33}; // 62+20+6+33 = 121
<pnoun_planet_moon_human>^{62} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus | almathea | ariel | callisto | charon | deimos | dione | enceladus | europa | ganymede |
hyperion | iapetus | io | janus | jupitereighth | jupitereleventh | jupiterfourteenth | jupiterninth |
jupiterseventh | jupitersixth | jupitertenth | jupiterthirteenth | jupitertwelfth | luna | mimas |
miranda | nereid | oberon | phobos | phoebe | rhea | saturnfirst | tethys | titan | titania |
triton | umbriel | bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel |
huygens | kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ; // 6+9+5 = 20
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{33} = <country>^{6} | <capital>^{6} | <city>^{6} | <continent>^{6} | <ocean>^{4} | <river>^{3} |
<lake>^{1} | <mountain>^{1} ; // 6+6+6+6+4+3+1+1 =33
<country>^{6} = canada | china | England | France | Germany | united states;
<capital>^{6} = ottawa | Beijing | london | paris | berlin | washington;
<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;
<transvb>^{15} = orbit | orbits | discover | discovered | neighbour | neighbours | neighboured | worship |
worshiped | contain | contains | contained | find | finds | found;
<preposition>^{2} = in | on ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does ;

Figure Appendix B (2): language-size computation of syntactic grammar (Cont'd)

Figure Appendix B (3): language-size computation of word-sequence grammar

```
/* 10-word word-sequence grammar
   wordSequence_gram_ext1.gram
*/
grammar wordSequence_gram_ext1;
public <s>^230791714483103775189388 2 = <word>
      |<word> <word>
      |<word> <word> <word>
      |<word> <word> <word> <word>
      |<word> <word><word> <word><word>
      |<word> <word><word> <word><word> <word>
      |<word> <word><word> <word><word> <word><word>
      |<word> <word><word> <word><word> <word><word> <word>
      |<word> <word><word> <word><word> <word><word> <word><word>
      |<word> <word><word> <word><word> <word><word> <word><word> <word>
       |<simple>^26;
// 273 + 273^2 + 273^3 + 273^4 + 273^5 +273^6 + 273^7 + 273^8 + 273^9 +273^10 =
// 273 + 74529 + 20346417 + 5554571841 + 1516398112593 + 413976684737889 +
// + 113015634933443697 + 30853268336830129281 + 8422942255954625293713 +
// + 2299463235875612705183649 =
// = 2307917144831037751893882 = 2.31 * 10^24
<simple>^26 = | ask them to be quite
         | please introduce yourself
         | hello there
         | goodbye
         | goodbye solar man
         | fine thanks
         | thanks
         | thanks solar man
         | yes please
         | what is your name
         | who are you
         | where do youlive
         | what do youknow
         | how old are you
         | what is your favorite band
         | who is the vice president at the university of windsor
         | who is the dean of science at the university of windsor
         | tell me a poem
         | know any poems
         | tell me a joke
         | know any jokes
         | who is judy
         | can i talk to judy
         | can i talk to solar man
         | who is monty
         | can i talk to monty ;
<word>^273 = <cnoun>^36 | <adj>^2 | <verb>^30 | <quest1>^3 | <det>^8 | <preposition>^2 | <pnoun>^121 |
           <nonhuman_pnoun_chemical>^20 |<space_program>^6 | <earth_geography_domain>^33 |
           <other_word>^12 ; // 36+2+30+3+8+2+121+20+6+33+12 = 273
```

<cnoun>^{36} = man | men | person | people | planet | planets | moon | moons | mountain | mountains |
crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal|
metals| nonmetal | nonmetals | country | countries | capital | capitals | city |cities |continent|
continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<verb>^{30} = <intransvb>^{7}
| <intransvb_other>^{2}
| <animate_transvb>^{6}
| <animate_transvb_other>^{2}
| <inanimate_transvb>^{6}
| <inanimate_transvb_other>^{3}
| <linkingvb>^{4}; // 7+2+6+2+6+3+4 = 30
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find |finds |found;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<preposition>^{2} = in | on;
<pnoun>^{121} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon> ^{36}
| <human_pnoun> ^{17}
| <nonhuman_pnoun_other>^{59}; //9+36+17+59=121
<nonhuman_pnoun_planet>^{9} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus | europa |
ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon |phobos|
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |
kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{59} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{33}; //20+6+33=59
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ; //6+9+5=20
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{33} = <country>^{6} | <capital>^{6} | <city>^{6} | <continent>^{6} | <ocean>^{4} | <river>^{3} |
<lake>^{1} | <mountain>^{1} ; 6+6+6+6+4+3+1+1=33
<country>^{6} = canada | china | England | France | Germany | united states;
<capital>^{6} = ottawa | Beijing | london | paris | berlin | washington;
<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;

<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;
<other_word>^{12} = sun | or | and | by | which | who | what | how | many | monty | judy | solar ;

Figure Appendix B (3): language-size computation of word-sequence grammar (Cont'd)

Figure Appendix B (4): language-size computation of extended semantic grammar

/* semantics_gram_ext2.gram */
grammar semantics_gram_ext2 ;
public <s>^{5550333776870} = <linkingvb>^{4} <termphrase_verbphrase>^{857815517151}
| is <pnoun>^{395} <pnoun>^{395}
| is <pnoun>^{395} (a|an)^{2} <nouncla>^{108}
| is <pnoun>^{395} (a|an)^{2} <nouncla>^{108} or (a|an)^{2} <nouncla>^{108}
| <quest1>^{3} <sent>^{706042576772}
| (who) <animate_verbph>^{22511168}
| (what) <inanimate_verbph>^{6692235}
| (which | how many)^{2} <nouncla_verbph>^{291754404}
| (which | how many)^{2} <nouncla_verbph_other>^{156297624}
| <simple>^{26} ;
// 4*857815517151 + 395*395 + 395*2*108 + 395*2*108*2*108 +3*706042576772 + 22511168+
// + 6692235 + 2*291754404 + 2*156297624 + 26 =
// = 3431262068604 +156025 +85320 + 18429120 +2118127730316 +29203403 +583508808+
// +312595248 +26 = 5550333776870 = 5.55 *10^{12}
<simple>^{26} = | ask them to be quite
| please introduce yourself
| hello there
| goodbye
| goodbye solar man
| fine thanks
| thanks
| thanks solar man
| yes please
| what is your name
| who are you
| where do youlive
| what do youknow
| how old are you
| what is your favorite band
| who is the vice president at the university of windsor
| who is the dean of science at the university of windsor
| tell me a poem
| know any poems
| tell me a joke
| know any jokes
| who is judy
| can i talk to judy
| can i talk to solar man

| who is monty
| can i talk to monty ;
<termphrase_verbphrase>^{857815517151} = <nonhuman_termph_planet>^{6555} <transvb_by_termph>^{6302247}
| <nonhuman_termph_moon>^{14196} <animate_transvb>^{6} by <human_termph>^{25651}
| <nonhuman_termph_other>^{2021055} <animate_transvb>^{6} by <human_termph>^{25651}
| <nonhuman_termph_other>^{2021055} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_planet>^{6555}
| <nonhuman_termph_other>^{2021055} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_moon>^{14196} ;
// 6555*6302247 +14196*6*25651+2021055*6*25651 +2021055*6*2*6555 + 2021055*6*2*14196
//= 41311229085 + 2184849576 + 311052490830+ 158976186300 + 344290761360 =
//= 857815517151
<transvb_by_termph>^{6302247} = <animate_transvb>^{6} by <human_termph>^{25651}
| <inanimate_transvb>^{6} by <nonhuman_termph_moon>^{14196}
| <inanimate_transvb_other>^{3} by <nonhuman_termph_other>^{2021055} ;
// 6*25651 + 6*14196 + 3*2021055 = 153906 + 85176 + 6063165 = 6302247
<sent>^{706042576772} = <human_termph>^{25651} <animate_verbph>^{22511168}
| <nonhuman_termph_moon>^{14196} <inanimate_verbph_active>^{39337}
| <nonhuman_termph_planet>^{6555} <inanimate_verbph_passive>^{340717}
| <nonhuman_termph_moon>^{14196} <inanimate_verbph_active_other>^{6063167}
| <nonhuman_termph_planet>^{6555} <inanimate_verbph_active_other>^{6063167} ;
// 25651*22511168 + 14196*39337 + 6555*340717 + 14196*6063167 + 6555*6063167=
//= 577433970368 + 558428052 +2233399935 + 86072718732 +39744059685 =
//= 706042576772
<nouncla_verbph>^{291754404} = <human_nouncla>^{12} <animate_verbph>^{22511168}
| <nonhuman_nouncla_moon>^{6} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_planet>^{6} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_moon>^{6} <inanimate_verbph_active>^{39337}
| <nonhuman_nouncla_planet>^{6} <inanimate_verbph_passive>^{340717} ;
// 12*22511168 + 6*1611672 + 6*1611672 + 6*39337 + 6*340717 =
// 270134016 + 19340064+ 236022 + 2044302 = 291754404
<nouncla_verbph_other>^{156297624} = <nonhuman_nouncla_other>^{84} <animate_verbph_passive>^{1611672}
| <nonhuman_nouncla_other>^{84} <inanimate_verbph_passive_other>^{249014};
// 84*1611672 + 84*249014 = 135380448+ 20917176 = 156297624
<inanimate_verbph>^{6692235} = <inanimate_verbph_active>^{39337}
| <inanimate_verbph_passive>^{340717}
| <inanimate_verbph_active_other>^{6063167}
| <inanimate_verbph_passive_other>^{249014};
// 39337 +340717+ 6063167+249014 = 6692235
<human_stermph>^{113} = <human_pnoun>^{17}
| <human_detph>^{96} ; // 17+ 96 =113
<nonhuman_stermph_planet>^{57} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_detph_planet>^{48} ; // 9+48 =57
<nonhuman_stermph_moon>^{84} = <nonhuman_pnoun_moon>^{36}
| <nonhuman_detph_moon>^{48} ; // 36+48 =84
<nonhuman_stermph_other>^{1005} = <nonhuman_pnoun_other>^{333}
| <nonhuman_detph_other>^{672} ; //333+672 = 1005
<human_termph>^{25651} = <human_stermph>^{113}
| <human_stermph>^{113} (and | or)^{2} <human_stermph>^{113} ;
//113+113*2*113=25651
<nonhuman_termph_planet>^{6555} = <nonhuman_stermph_planet>^{57}
| <nonhuman_stermph_planet>^{57} (and | or)^{2} <nonhuman_stermph_planet>^{57} ;

// 57+57*2*57 = 6555
<nonhuman_termph_moon>^{14196} = <nonhuman_stermph_moon>^{84}
| <nonhuman_stermph_moon>^{84} (and | or) <nonhuman_stermph_moon>^{84} ;
// 84 +84*2*84 = 14196
<nonhuman_termph_other>^{2021055} = <nonhuman_stermph_other>^{1005}
| <nonhuman_stermph_other>^{1005} (and | or) ^{2} <nonhuman_stermph_other>^{1005};
// 1005 + 1005*2*1005 = 2021055
<animate_verbph>^{22511168} = <animate_transvbph>^{22511168} ;
<inanimate_verbph_active>^{39337} = <inanimate_transvbph_active>^{39330}
| <intransvb>^{7} ; // 39330+7 = 39337
<inanimate_verbph_passive>^{340717} = <inanimate_transvbph_passive>^{340704}
| <intransvb>^{7}
| <inanimate_transvb>^{6} sun ; // 340704 +7 +6 = 340717
<inanimate_verbph_active_other>^{6063167} = <inanimate_transvbph_active_other>^{6063165}
| <intransvb_other>^{2}; // 6063165 + 2 =6063167
<inanimate_verbph_passive_other>^{249014} = <inanimate_transvbph_passive_other>^{249012}
| <intransvb_other>^{2}; // 249012 +2 = 249014
<animate_verbph_passive>^{1611672} = <linkingvb>^{4} <animate_transvb>^{6} by <human_termph>^{25651}
| <linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2} <nonhuman_termph_planet>^{6555}
| <linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2} <nonhuman_termph_moon>^{14196} ;
//4*6*25651 + 4*6*2*6555 + 4*6*2*14196 = 1611672
<animate_transvbph>^{22511168} = <animate_transvb>^{6} (<nonhuman_termph_planet>^{6555}
| <nonhuman_termph_moon>^{14196}
| <nonhuman_termph_other>^{2021055})
| <animate_transvb_other>^{2} (<human_termph>^{25651}
| <nonhuman_termph_planet>^{6555}
| <nonhuman_termph_moon>^{14196}
| <nonhuman_termph_other>^{2021055});
// 9*(6555+14196+2021055) + 2*(25651+6555+14196+2021055) =
// 9*2041806 + 2*2067457 = 18376254 + 4134914 = 22511168
<inanimate_transvbph_active>^{39330} = <inanimate_transvb>^{6} <nonhuman_termph_planet>^{6555} ;
//6*6555=39330
<inanimate_transvbph_passive>^{340704} =
<linkingvb>^{4} <inanimate_transvb>^{6} by <nonhuman_termph_moon>^{14196} ;
// 4*6*14196 = 340704
<inanimate_transvbph_active_other>^{6063165} =
<inanimate_transvb_other>^{3} <nonhuman_termph_other>^{2021055} ;
// 3*2021055 = 6063165
<inanimate_transvbph_passive_other>^{249012} =
<linkingvb>^{4} <inanimate_transvb_other>^{3} by <nonhuman_termph_planet>^{6555}
| <linkingvb>^{4} <inanimate_transvb_other>^{3} by <nonhuman_termph_moon>^{14196} ;
// 4*3*6555 + 4*3+14196 = 78660 + 170352 = 249012
<human_detph>^{96} = <det>^{8} <human_nouncla>^{12} ; //8*12 = 96
<nonhuman_detph_planet>^{48} = <det>^{8} <nonhuman_nouncla_planet>^{6} ; //8*6 =48
<nonhuman_detph_moon>^{48} = <det>^{8} <nonhuman_nouncla_moon>^{6} ; //6*8 =48
<nonhuman_detph_other>^{672} = <det>^{8} <nonhuman_nouncla_other>^{84} ; //8*84 =672
<preposition>^{2} = on | in ;
<nouncla>^{108} = <human_nouncla>^{12}
| <nonhuman_nouncla_planet>^{6}
| <nonhuman_nouncla_moon>^{6}
| <nonhuman_nouncla_other>^{84}; // 12+6+6+84 = 108
<human_nouncla>^{12} = <adj>^{2} <human_cnoun>^{4}

| <human_cnoun>^{4} ; //2*4+4 =12
<nonhuman_nouncla_planet>^{6} = <adj>^{2} <nonhuman_cnoun_planet>^{2}
| <nonhuman_cnoun_planet>^{2} ; //2*2+2 =6
<nonhuman_nouncla_moon>^{6} = <adj>^{2} <nonhuman_cnoun_moon>^{2}
| <nonhuman_cnoun_moon>^{2} ; //2*2+2 =6
<nonhuman_nouncla_other>^{84} = <adj>^{2} <nonhuman_cnoun_other>^{28}
| <nonhuman_cnoun_other>^{28} ; // 2*28 + 28 = 84
<human_cnoun>^{4} = man | men | person | people;
<nonhuman_cnoun_planet>^{2} = planet | planets ;
<nonhuman_cnoun_moon>^{2} = moon | moons;
<nonhuman_cnoun_other>^{28} = mountain | mountains | crater | craters | sea | seas | ocean | oceans |
chemical | chemicals | gas | gases | metal | metals | nonmetal | nonmetals |
country | countries | capital | capitals | city | cities | continent |
continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find | finds | found ;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{395} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon>^{36}
| <human_pnoun>^{17}
| <nonhuman_pnoun_other>^{333}; // 9+36+17+333 = 395
<nonhuman_pnoun_planet>^{9} = urth | jupiter | mars | mercury | neptune | pluto | saturn | uranus | venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus |
europa | ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon | phobos |
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |
kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{333} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{307}; // 20+6+207 =333
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ; //6+9+5 =20
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{307} = <country>^{187} | <capital>^{98} | <city>^{6} | <continent>^{7} | <ocean>^{4} |
<river>^{3} | <lake>^{1} | <mountain>^{1} ; // 187+98+6+7+4+3+1+1 =307
<country>^{187} = Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguilla |
Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan |
Bahrain |Bangladesh | Barbados | Bassas da India | Belarus | Belgium | Belize | Benin |

Bermuda | Bhutan | Bolivia |Bosnia and Herzegovina | Botswana |Bouvet Island | Brazil|
Brunei |Bulgaria | Burkina Faso | Burma | Burundi | Cambodia | Caneriib | Canada |
Cape Verde | Cayman Islands | Central African Republic | Chad | Chile |
China | Clipperton Island | Colombia | Comoros | Congo Democratic Republic |
Congo Republic | Cook Islands | Coral Sea Islands | Costa Rica | Croatia | Cuba | Cyprus |
Czech Republic | Denmark | Djibouti | Dominica | Dominica Republic | Ecuador | Egypt |
El Salvador | Equatorial Guinea | Eritrea | Estonia | Ethiopia | Europe Island | Fiji | Finland |
France |French Guiana | Gabon | Gambia | Gaza Strip |Georgia | Germany | Ghana |
Gibraltar | Glorioso Island | Greece | Greenland | Grenada | Guadeloupe | Guam |
Guatemala | Guernsey | Guinea | Guyana | Haiti | Heard and Mcdonald Island |Holy See |
Honduras | Howland Island | Hungary |Iceland |India |Indonesia | Iran |Iraq | Ireland | Israel|
Italy | Jamaica | Jan Mayen |Japan | Jarvis Island | Jersey | Johnston Atoll | Jordan |
Kazakhstan | Kenya |Kingman Reef | Kiribati | North Korea | South Korea | Kuwait |
KyrgyzStan | Laos | Latvia |Lebanon | Lesotho |Liberia |Libya | Liechtenstein | Lithuania |
Luxembourg | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta |
Isle of Man | Marshall Islands | Martinique | Mauritania |Mauritius | Mayotte| Mexico|
Micronesia | Midway Island |Moldova | Monaco | Mongolia | Montserrat | Morocco |
Mozambique |Myanmar | Netherlands | Norway | New Zealand | Nigeria | Oman | Portugal |
Poland | Romania | Russia | Rwanda | Tajikistan | Tanzania | Syria | Swede | Switzerland |
Sudan | Spain | Singapore | Thailand | Togo |Tokelau | Tonga |Tunisia | Turkey |
Turkmenistan | Tuvalu | Uganda | Ukraine |united Arab Emirates | United Kingdom |
United States of Amerima |Uruguay | Uzbekistan |Vietnam | Yemen | Yugoslavia |Zambia|
Zimbabwe ;

<capital>^{98} = ottawa | Beijing | london | paris | berlin | Washington| Kabul |Tirana |Algiers | Pago Pago|
Luanda | Andorra la Vella | Buenos Aires | Yerevan | Oranjestad | Canberra | Vienna |
Baku| Dhaka |Manama | BridgeTown | Brussels | Belmopan | Portonovo | Hamilton |
Thimphu | LaPaz |Gaborone |Brasilia| Phnom Penh |Yaounde | Praia | Prague | Santiago |
Bogota | Moroni | Havana | Nicosia | Copenhagen | Roseau | Cairo | Asmara |Addis Ababa|
Suva | Helsinki | Libreville | Banjul |ueregeTown | Tbilisi |Accra | Athens |
Saint George's | Conakry | Port-au-prince | Budapest | New Delhi | Jakarta | Tehran |
Baghdad | Dublin | Jerusalem | Rome | Tokyo | Amman | PYong Yang | Seoul | Kuwait |
Beirut | Maseru | Monrovia | Tripoli | Skopje | Amsterdam | Kuala Lumpur | Bamako |
Velletta | Mexico | Ulaanbaatar | Windhoek | Abuja | Wellington | Oslo | Warsaw | Lisbon |
Moscow | Stockholm | Bucharest | Singapore | Madrid | Khartoum | Bern | Damascus |
Hanoi |Ankara | Sanaa | Harare | Belgrade | Lusaka ;

<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;

<continent>^{7} = Africa | Asia | Austrilia | Europe | North America | South America;

<ocean>^{4} = Arctic | Atlantic | India | Pacific;

<river>^{3} = Yangtse | Nile | Danube ;

<lake>^{1} = ontario lake;

<mountain>^{1} = rocky mountain;

Figure Appendix B (4): language-size computation of extended semantic grammar (Cont'd)

Figure Appendix B (5): language-size computation of extended syntactic grammar

/* syntax_gram_ext2.gram */

grammar syntax_gram_ext2 ;

public <s>^{8172962971642012} = <linkingvb>^{4} <termph>^{3176460} [<transvb>^{15} by] <termph>^{3176460}
| <linkingvb>^{4} <termph>^{3176460} [<transvb>^{15} <preposition>^{2}] <termph>^{3176460}

| <quest1>^{3} <sent>^{2118878629871220}
| (who |what) ^{2} <verbph>^{667056607}
| (which | how many) ^{2} <nouncla>^{108}<verbph>^{667056607}
| <simple>^{26} ;
// 4*3176460*15*3176460 + 4*3176460*15*2*3176460 + 3*2118878629871220 + 2*667056607+
// + 2*108*667056607 +26
//=605393887896000+1210787775792000+6356635889613660 + 1334113214+144084227112+26
//= 8172962971642012 = 8.17 *10^{15}

<simple>^{26} = | ask them to be quite
| please introduce yourself
| hello there
| goodbye
| goodbye solar man
| fine thanks
| thanks
| thanks solar man
| yes please
| what is your name
| who are you
| where do youlive
| what do youknow
| how old are you
| what is your favorite band
| who is the vice president at the university of windsor
| who is the dean of science at the university of windsor
| tell me a poem
| know any poems
| tell me a joke
| know any jokes
| who is judy
| can i talk to judy
| can i talk to solar man
| who is monty
| can i talk to monty ;

<sent>^{2118878629871220} = <termph>^{3176460} <verbph>^{667056607}; // 3176460 * 667056607 = 2118878629871220
<stermph>^{1260} = <pnoun>^{396} | <detph>^{864}; // 396 + 864 = 1260
<termph>^{3176460} = <stermph>^{1260}
| <stermph>^{1260} (and | or) ^{2} <stermph>^{1260}; //1260 +1260*2*1260= 3176460
<verbph>^{667056607} = <transvbph>^{667056600} | <intransvb>^{7}; // 667056600 + 7 = 667056607
<transvbph>^{667056600} = (<transvb>^{15} | <linkingvb>^{4} <transvb>^{15} by) <termph>^{3176460}
| (<transvb>^{15} | <linkingvb>^{4} <transvb>^{15} <preposition>^{2}) <termph>^{3176460} ;
// (15 + 4*15)*3176460 + (15+4*15*2)*3176460 = 238234500 +428822100 = 667056600
<detph>^{864} = <det>^{8} <nouncla>^{108}; // 8*108 = 864
<nouncla>^{108} = <adj>^{2} <cnoun>^{36} | <cnoun>^{36}; //2*36 +36 = 108
<cnoun>^{36} = man | men | person | people | planet | planets | moon | moons | mountain | mountains | crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal | metals | nonmetal | nonmetals | country | countries | capital | capitals | city | cities | continent | continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits| orbited | exist | exists ;
<det>^{8} = a | an | every | one | two | three | four | five;

<pnoun>^{396} = <pnoun_planet_moon_human>^{63}
| <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{307}; //63+20+6+307 = 396
<pnoun_planet_moon_human>^{63} = urth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus | almathea | ariel | callisto | charon | deimos | dione | enceladus | europa |
ganymede | hyperion | iapetus | io | janus | jupiter eighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas | miras | miranda | nereid | oberon |
phobos | phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel | bernard|
bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens | kowal |kuiper|
larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ;
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{307} = <country>^{187} | <capital>^{98} | <city>^{6} | <continent>^{7} | <ocean>^{4} |
<river>^{3} | <lake>^{1} | <mountain>^{1} ; //187+98+6+7+4+3+1+1=307
<country>^{187} = Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguilla |
Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan |
Bahrain |Bangladesh | Barbados | Bassas da India | Belarus | Belgium | Belize | Benin |
Bermuda | Bhutan | Bolivia |Bosnia and Herzegovina | Botswana |Bouvet Island | Brazil|
Brunei |Bulgaria | Burkina Faso | Burma | Burundi | Cambodia | Caneriib | Canada |
Cape Verde | Cayman Islands | Central African Republic | Chad | Chile |
China | Clipperton Island | Colombia | Comoros | Congo Democratic Republic |
Congo Republic | Cook Islands | Coral Sea Islands | Costa Rica | Croatia | Cuba | Cyprus |
Czech Republic | Denmark | Djibouti | Dominica | Dominica Republic | Ecuador | Egypt |
El Salvador | Equatorial Guinea | Eritrea | Estonia | Ethiopia | Europe Island | Fiji |Finland|
France |French Guiana | Gabon | Gambia | Gaza Strip |Georgia | Germany | Ghana |
Gibraltar | Glorioso Island | Greece | Greenland | Grenada | Guadeloupe | Guam |
Guatemala | Guernsey | Guinea | Guyana | Haiti | Heard and Mcdonald Island |Holy See |
Honduras | Howland Island | Hungary |Iceland |India |Indonesia | Iran |Iraq | Ireland | Israel|
Italy | Jamaica | Jan Mayen |Japan | Jarvis Island | Jersey | Johnston Atoll | Jordan |
Kazakhstan | Kenya |Kingman Reef | Kiribati | North Korea | South Korea | Kuwait |
KyrgyzStan | Laos | Latvia |Lebanon | Lesotho |Liberia |Libya | Liechtenstein | Lithuania |
Luxembourg | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta |
Isle of Man | Marshall Islands | Martinique | Mauritania |Mauritius | Mayotte| Mexico|
Micronesia | Midway Island |Moldova | Monaco | Mongolia | Montserrat | Morocco |
Mozambique |Myanmar | Netherlands | Norway | New Zealand | Nigeria |
Oman | Portugal | Poland | Romania | Russia | Rwanda | Tajikistan | Tanzania |
Syria | Swede | Switzerland | Sudan | Spain | Singapore |
Thailand | Togo |Tokelau | Tonga |Tunisia | Turkey | Turkmenistan |Tuvalu | Uganda |
Ukraine |united Arab Emirates | United Kingdom | United States of Amerima | Uruguay |
Uzbekistan |Vietnam | Yemen | Yugoslavia |Zambia | Zimbabwe ;
<capital>^{98} = ottawa | Beijing | london | paris | berlin | Washington |
Kabul | Tirana |Algiers | Pago Pago | Luanda | Andorra la Vella | Buenos Aires |
Yerevan | Oranjestad | Canberra | Vienna | Baku | Dhaka |Manama | BridgeTown |
Brussels | Belmopan | Portonovo | Hamilton | Thimphu | LaPaz |Gaborone |Brasilia|

```
        Phnom Penh |Yaounde | Praia |  Prague | Santiago | Bogota | Moroni | Havana |
        Nicosia | Copenhagen | Roseau | Cairo | Asmara | Addis Ababa | Suva | Helsinki |
        Libreville | Banjul | GoregeTown | Tbilisi |Accra | Athens | Saint George's | Conakry |
        Port-au-prince | Budapest | New Delhi | Jakarta | Tehran | Baghdad | Dublin |
         Jerusalem | Rome | Tokyo | Amman | PYong Yang | Seoul | Kuwait | Beirut |
         Maseru | Monrovia | Tripoli | Skopje | Amsterdam | Kuala Lumpur | Bamako |
         Velletta | Mexico | Ulaanbaatar | Windhoek | Abuja | Wellington | Oslo |
         Warsaw | Lisbon | Moscow | Stockholm | Bucharest | Singapore | Madrid |
         Khartoum | Bern | Damascus | Hanoi |Ankara | Sanaa | Harare | Belgrade | Lusaka ;
<city>^6 = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^7 = Africa | Asia | Austrilia | Europe | North America | South America | Antarctica;
<ocean>^4 = Arctic | Atlantic |  India | Pacific;
<river>^3 = Yangtse | Nile | Danube ;
<lake>^1 = ontario lake;
<mountain>^1 = rocky mountain;
<transvb>^15 = orbit | orbits | discover | discovered | neighbour | neighbours | neighboured | worship |
            worshiped | contain | contains | contained | find | finds | found;
<preposition>^2 = in | on ;
<linkingvb>^4 = is | was | are | were ;
<quest1>^3 = did | do | does ;
```

Figure Appendix B (5): language-size computation of extended syntactic grammar (Cont'd)

Figure Appendix B (6): language-size computation of extended word-sequence grammar

```
/* extended 10-word word-sequence grammar
   wordSequence_gram_ext1.gram   */
grammar wordSequence_gram_ext1;
public <s>^2402525173996203346833004636 = <word>
     |<word> <word>
     |<word> <word> <word>
     |<word> <word> <word> <word>
     |<word> <word><word> <word><word>
     |<word> <word><word> <word><word> <word>
     |<word> <word><word> <word><word> <word><word>
     |<word> <word><word> <word><word> <word><word> <word>
     |<word> <word><word> <word><word> <word><word> <word><word>
     |<word> <word><word> <word><word> <word><word> <word><word> <word>
      |<simple>^26;
// 547 + 547^2 + 547^3 + 547^4 + 547^5 +547^6 + 547^7 + 547^8 + 547^9 +547^10 =
// 547 + 299209 + 163667323 + 89526025681 + 48970736047507 +26786992617986329 +
//+ 14652484962038521963 + 8014909274235071513761 + 4384155373006584118027267 +
// + 2398132989034601512560915049 =
// = 2402525173996203346833004636 = 2.40* 10^27
<simple>^26 = | ask them to be quite
          | please introduce yourself
          | hello there
          | goodbye
          | goodbye solar man
          | fine thanks
```

| thanks
| thanks solar man
| yes please
| what is your name
| who are you
| where do youlive
| what do youknow
| how old are you
| what is your favorite band
| who is the vice president at the university of windsor
| who is the dean of science at the university of windsor
| tell me a poem
| know any poems
| tell me a joke
| know any jokes
| who is judy
| can i talk to judy
| can i talk to solar man
| who is monty
| can i talk to monty ;

<word>^{547} = <cnoun>^{36} | <adj>^{2} | <verb>^{30} | <quest1>^{3} | <det>^{8} | <preposition>^{2} | <pnoun>^{121} |
<nonhuman_pnoun_chemical>^{20} |<space_program>^{6} | <earth_geography_domain>^{307} |
<other_word>^{12} ; // 36+2+30+3+8+2+121+20+6+33+12 = 547
<cnoun>^{36} = man | men | person | people | planet | planets | moon | moons | mountain | mountains |
crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal|
metals| nonmetal | nonmetals | country | countries | capital | capitals | city |cities |continent|
continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<verb>^{30} = <intransvb>^{7}
| <intransvb_other>^{2}
| <animate_transvb>^{6}
| <animate_transvb_other>^{2}
| <inanimate_transvb>^{6}
| <inanimate_transvb_other>^{3}
| <linkingvb>^{4}; // 7+2+6+2+6+3+4 = 30
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find |finds |found;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<preposition>^{2} = in | on;
<pnoun>^{121} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon> ^{36}
| <human_pnoun> ^{17}
| <nonhuman_pnoun_other>^{59}; //9+36+17+59=121
<nonhuman_pnoun_planet>^{9} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus | europa |

ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon |phobos|
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |
kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{59} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{307}; //20+6+33=59
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ; //6+9+5=20
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{307} = <country>^{187} | <capital>^{98} | <city>^{6} | <continent>^{7} | <ocean>^{4} |
<river>^{3} | <lake>^{1} | <mountain>^{1} ; //187+98+6+7+4+3+1+1=307
<country>^{187} = Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguilla |
Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan |
Bahrain |Bangladesh | Barbados | Bassas da India | Belarus | Belgium | Belize | Benin |
Bermuda | Bhutan | Bolivia |Bosnia and Herzegovina | Botswana |Bouvet Island | Brazil|
Brunei |Bulgaria | Burkina Faso | Burma | Burundi | Cambodia | Caneriib | Canada |
Cape Verde | Cayman Islands | Central African Republic | Chad | Chile |
China | Clipperton Island | Colombia | Comoros | Congo Democratic Republic |
Congo Republic | Cook Islands | Coral Sea Islands | Costa Rica | Croatia | Cuba | Cyprus |
Czech Republic | Denmark | Djibouti | Dominica | Dominica Republic | Ecuador | Egypt |
El Salvador | Equatorial Guinea | Eritrea | Estonia | Ethiopia | Europe Island | Fiji |Finland|
France |French Guiana | Gabon | Gambia | Gaza Strip |Georgia | Germany | Ghana |
Gibraltar | Glorioso Island | Greece | Greenland | Grenada | Guadeloupe | Guam |
Guatemala | Guernsey | Guinea | Guyana | Haiti | Heard and Mcdonald Island |Holy See |
Honduras | Howland Island | Hungary |Iceland |India |Indonesia | Iran |Iraq | Ireland | Israel|
Italy | Jamaica | Jan Mayen |Japan | Jarvis Island | Jersey | Johnston Atoll | Jordan |
Kazakhstan | Kenya |Kingman Reef | Kiribati | North Korea | South Korea | Kuwait |
KyrgyzStan | Laos | Latvia |Lebanon | Lesotho |Liberia |Libya | Liechtenstein | Lithuania |
Luxembourg | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta |
Isle of Man | Marshall Islands | Martinique | Mauritania |Mauritius | Mayotte| Mexico|
Micronesia | Midway Island |Moldova | Monaco | Mongolia | Montserrat | Morocco |
Mozambique |Myanmar | Netherlands | Norway | New Zealand | Nigeria |
Oman | Portugal | Poland | Romania | Russia | Rwanda | Tajikistan | Tanzania |
Syria | Swede | Switzerland | Sudan | Spain | Singapore |
Thailand | Togo |Tokelau | Tonga |Tunisia | Turkey | Turkmenistan |Tuvalu | Uganda |
Ukraine |united Arab Emirates | United Kingdom | United States of Amerima | Uruguay |
Uzbekistan |Vietnam | Yemen | Yugoslavia |Zambia | Zimbabwe ;
<capital>^{98} = ottawa | Beijing | london | paris | berlin | Washington |
Kabul | Tirana |Algiers | Pago Pago | Luanda | Andorra la Vella | Buenos Aires |
Yerevan | Oranjestad | Canberra | Vienna | Baku | Dhaka |Manama | BridgeTown |
Brussels | Belmopan | Portonovo | Hamilton | Thimphu | LaPaz |Gaborone |Brasilia|
Phnom Penh |Yaounde | Praia | Prague | Santiago | Bogota | Moroni | Havana |
Nicosia | Copenhagen | Roseau | Cairo | Asmara | Addis Ababa | Suva | Helsinki |
Libreville | Banjul | GoregeTown | Tbilisi |Accra | Athens | Saint George's | Conakry |

Port-au-prince | Budapest | New Delhi | Jakarta | Tehran | Baghdad | Dublin |
Jerusalem | Rome | Tokyo | Amman | PYong Yang | Seoul | Kuwait | Beirut |
Maseru | Monrovia | Tripoli | Skopje | Amsterdam | Kuala Lumpur | Bamako |
Velletta | Mexico | Ulaanbaatar | Windhoek | Abuja | Wellington | Oslo |
Warsaw | Lisbon | Moscow | Stockholm | Bucharest | Singapore | Madrid |
Khartoum | Bern | Damascus | Hanoi |Ankara | Sanaa | Harare | Belgrade | Lusaka ;

<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;
<other_word>^{12} =sun|or | and | by | which | who | what | how | many | monty | judy | solar ;

Figure Appendix B (6): language-size computation of extended word-sequence grammar (Cont'd)

Appendix C: Computation of Branching Factor in Detail

Note: superscripts are used to denote the branching factors of the preceding expressions; the underlined superscripts are used for average branching-factor computation..

Figure Appendix C (1): branching-factor computation of semantic grammar

```
/* semantics_gram_ext1.gram */
grammar semantics_gram_ext1;
public <s>^42 = <linkingvb>^4 <termphrase_verbphrase>^524
              | is^1 <pnoun>^121 <pnoun>^121
              | is^1 <pnoun>^121 ( a|an )^2 <nouncla>^44
              | is^1 <pnoun>^121 ( a|an )^2 <nouncla>^44 or^1 ( a|an )^2 <nouncla>^44
              | <quest1>^3 <sent>^294
              | ( who )^1 <animate_verbph>^8
              | ( what )^1 <inanimate_verbph>^45
              | ( which | how many )^2 <nouncla_verbph>^22
              | ( which | how many )^2 <nouncla_verbph_other>^60
              | <simple>^26;
<simple>^26 = | ask them to be quite
        | please introduce yourself
        | hello there
        | goodbye
        | goodbye solar man
        | fine thanks
        | thanks
        | thanks solar man
        | yes please
        | what is your name
        | who are you
        | where do youlive
        | what do youknow
        | how old are you
        | what is your favorite band
        | who is the vice president at the university of windsor
        | who is the dean of science at the university of windsor
        | tell me a poem
        | know any poems
        | tell me a joke
        | know any jokes
        | who is judy
        | can i talk to judy
        | can i talk to solar man
        | who is monty
```

```
        | can i talk to monty ;
<termphrase_verbphrase>^524 = <nonhuman_termph_planet>^34 <transvb_by_termph>^15
                | <nonhuman_termph_moon>^88 <animate_transvb>^6 by^1 <human_termph>^50
                | <nonhuman_termph_other>^134 <animate_transvb>^6 by^1 <human_termph>^50
                | <nonhuman_termph_other>^134 <animate_transvb>^6 <preposition>^2
                                              <nonhuman_termph_planet>^34
                | <nonhuman_termph_other>^134 <animate_transvb>^6 <preposition>^2
                                              <nonhuman_termph_moon>^88 ;
<transvb_by_termph>^15 = <animate_transvb>^6 by^1 <human_termph>^50
                | <inanimate_transvb>^6 by^1 <nonhuman_termph_moon>^88
                | <inanimate_transvb_other>^3 by^1 <nonhuman_termph_other>^134 ;
<sent>^294 = <human_termph>^50 <animate_verbph>^8
        | <nonhuman_termph_moon>^88 <inanimate_verbph_active>^13
        | <nonhuman_termph_planet>^34 <inanimate_verbph_passive>^17
        | <nonhuman_termph_moon>^88 <inanimate_verbph_active_other>^5
        | <nonhuman_termph_planet>^34 <inanimate_verbph_active_other>^5;
<nouncla_verbph>^22 = <human_nouncla>^6 <animate_verbph>^8
                | <nonhuman_nouncla_moon>^4 <animate_verbph_passive>^12
                | <nonhuman_nouncla_planet>^4 <animate_verbph_passive>^12
                | <nonhuman_nouncla_moon>^4 <inanimate_verbph_active>^13
                | <nonhuman_nouncla_planet>^4 <inanimate_verbph_passive>^17 ;
<nouncla_verbph_other>^60 = <nonhuman_nouncla_other>^30 <animate_verbph_passive>^12
                | <nonhuman_nouncla_other>^30 <inanimate_verbph_passive_other>^10;
<inanimate_verbph>^45 = <inanimate_verbph_active>^13
                | <inanimate_verbph_passive>^17
                | <inanimate_verbph_active_other>^5
                | <inanimate_verbph_passive_other>^10;
<human_stermph>^25 = <human_pnoun>^17
                   | <human_detph>^8 ;
<nonhuman_stermph_planet>^17 = <nonhuman_pnoun_planet>^9
                             | <nonhuman_detph_planet>^8 ;
<nonhuman_stermph_moon>^44 = <nonhuman_pnoun_moon>^36
                             | <nonhuman_detph_moon>^8 ;
<nonhuman_stermph_other>^67 = <nonhuman_pnoun_other>^59
                             | <nonhuman_detph_other>^8 ;
<human_termph>^50 = <human_stermph>^25
                   | <human_stermph>^25 ( and | or )^2 <human_stermph>^25 ;
<nonhuman_termph_planet>^34 = <nonhuman_stermph_planet>^17
                   | <nonhuman_stermph_planet>^17 ( and | or )^2 <nonhuman_stermph_planet>^17 ;
<nonhuman_termph_moon>^88 = <nonhuman_stermph_moon>^44
                   | <nonhuman_stermph_moon>^44 ( and | or )^2 <nonhuman_stermph_moon>^44 ;
<nonhuman_termph_other>^134 = <nonhuman_stermph_other>^67
                   | <nonhuman_stermph_other>^67 ( and | or )^2 <nonhuman_stermph_other>^67;
<animate_verbph>^8 = <animate_transvbph>^8;
<inanimate_verbph_active>^13 = <inanimate_transvbph_active>^6
                             | <intransvb>^7 ;
<inanimate_verbph_passive>^17 = <inanimate_transvbph_passive>^4
                             | <intransvb>^7
                             | <inanimate_transvb>^6 sun^1 ;
<inanimate_verbph_active_other>^5 = <inanimate_transvbph_active_other>^3
                                  | <intransvb_other>^2;
<inanimate_verbph_passive_other>^10 = <inanimate_transvbph_passive_other>^8
```

| <intransvb_other>^{2};
<animate_verbph_passive>^{12} = <linkingvb>^{4} <animate_transvb>^{6} by^{1} <human_termph>^{50}
| <linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_planet>^{34}
| <linkingvb>^{4} <animate_transvb>^{6} <preposition>^{2}
<nonhuman_termph_moon>^{88} ;
<animate_transvbph>^{8} = <animate_transvb>^{6} (<nonhuman_termph_planet>^{34}
| <nonhuman_termph_moon>^{88}
| <nonhuman_termph_other>^{134})
| <animate_transvb_other>^{2} (<human_termph>^{50}
| <nonhuman_termph_planet>^{34}
| <nonhuman_termph_moon>^{88}
| <nonhuman_termph_other>^{134});
<inanimate_transvbph_active>^{6} = <inanimate_transvb>^{6} <nonhuman_termph_planet>^{34};
<inanimate_transvbph_passive>^{4} = <linkingvb>^{4} <inanimate_transvb>^{6} by^{1}
<nonhuman_termph_moon>^{88} ;
<inanimate_transvbph_active_other>^{3} = <inanimate_transvb_other>^{3} <nonhuman_termph_other>^{134};
<inanimate_transvbph_passive_other>^{8} = <linkingvb>^{4} <inanimate_transvb_other>^{3} by^{1}
<nonhuman_termph_planet>^{34}
| <linkingvb>^{4} <inanimate_transvb_other>^{3} by ^{1}
<nonhuman_termph_moon>^{88} ;
<human_detph>^{8} = <det>^{8} <human_nouncla>^{6} ;
<nonhuman_detph_planet>^{8} = <det>^{8} <nonhuman_nouncla_planet>^{4};
<nonhuman_detph_moon>^{8} = <det>^{8} <nonhuman_nouncla_moon>^{4} ;
<nonhuman_detph_other>^{8} = <det>^{8} <nonhuman_nouncla_other>^{30} ;
<preposition>^{2} = on | in ;
<nouncla>^{44} = <human_nouncla>^{6}
| <nonhuman_nouncla_planet>^{4}
| <nonhuman_nouncla_moon>^{4}
| <nonhuman_nouncla_other>^{30};
<human_nouncla>^{6} = <adj>^{2} <human_cnoun>^{4}
| <human_cnoun>^{4} ;
<nonhuman_nouncla_planet>^{4} = <adj>^{2} <nonhuman_cnoun_planet>^{2}
| <nonhuman_cnoun_planet>^{2} ;
<nonhuman_nouncla_moon>^{4} = <adj>^{2} <nonhuman_cnoun_moon>^{2}
| <nonhuman_cnoun_moon>^{2} ;
<nonhuman_nouncla_other>^{30} = <adj>^{2} <nonhuman_cnoun_other>^{28}
| <nonhuman_cnoun_other>^{28} ;
<human_cnoun>^{4} = man | men | person | people;
<nonhuman_cnoun_planet>^{2} = planet | planets ;
<nonhuman_cnoun_moon>^{2} = moon | moons;
<nonhuman_cnoun_other>^{28} = mountain | mountains | crater | craters | sea | seas | ocean | oceans |
chemical | chemicals | gas | gases | metal | metals | nonmetal | nonmetals |
country | countries | capital | capitals | city | cities | continent |continents|
river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find | finds | found ;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;

<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{121} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon> ^{36}
| <human_pnoun> ^{17}
| <nonhuman_pnoun_other>^{59};
<nonhuman_pnoun_planet>^{9} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus | europa |
ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon | phobos |
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |
kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{59} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{33};
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ;
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{33} = <country>^{6} | <capital>^{6} | <city>^{6} | <continent>^{6} | <ocean>^{4} | <river>^{3} |
<lake>^{1} | <mountain>^{1} ;
<country>^{6} = canada | china | England | France | Germany | united states;
<capital>^{6} = ottawa | Beijing | london | paris | berlin | washington;
<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;

The average branching factor for semantic grammar
b =((42+524)+(121+121)+(121+2+44)+(121+2+44+1+2+44)+(260+8+45+22+60)+
(15+6+1+50+6+1+50+6+2+34+6+2+88)+(1+50+1+88+1+134)+(8+13+17+5+5)+
(8+12+12+13+17)+(12+10)+(2+25)+(2+17)+(2+44)+(2+67+1)+(6+1+50+6+2+34+6+2+88)+
(34+88+134)+(50+34+88+134)+(34)+(6+1+88)+(134)+(3+1+34+3+1+88)+(6)+(4+4+30)+
(4+2+2+28)) / 92
= 3684 / 93
= 39.6

Figure Appendix C (1): branching-factor computation of semantic grammar (Cont'd)

Figure Appendix C (2): branching-factor computation of syntactic grammar

```
/* syntax_gram_ext1.gram */
grammar syntax_gram_ext1 ;
public <s>^41 = <linkingvb>^4 <termph>^258 [<transvb>^15 by^1 ] <termph>^258
              | <linkingvb>^4 <termph>^258 [<transvb>^15 <preposition>^2 ] <termph>^258
              | <quest1>^3 <sent>^258
              | ( who |what) ^2 <verbph>^45
              | ( which | how many ) ^2 <nouncla>^38<verbph>^45
              | <simple>^26 ;
<simple>^26 = | ask them to be quite
        | please introduce yourself
        | hello there
        | goodbye
        | goodbye solar man
        | fine thanks
        | thanks
        | thanks solar man
        | yes please
        | what is your name
        | who are you
        | where do youlive
        | what do youknow
        | how old are you
        | what is your favorite band
        | who is the vice president at the university of windsor
        | who is the dean of science at the university of windsor
        | tell me a poem
        | know any poems
        | tell me a joke
        | know any jokes
        | who is judy
        | can i talk to judy
        | can i talk to solar man
        | who is monty
        | can i talk to monty ;
<sent>^258 = <termph>^258 <verbph>^45;
<stermph>^129 = <pnoun>^121
              | <detph>^8;
<termph>^258 = <stermph>^129
              | <stermph>^129 (and | or)^2 <stermph>^129;
<verbph>^45 = <transvbph>^38
             | <intransvb>^7;
<transvbph>^38 = ( <transvb>^15 | <linkingvb>^4 <transvb>^15 by^1 ) <termph>^258
              |   ( <transvb>^15 | <linkingvb>^4 <transvb>^15 <preposition>^2 ) <termph>^258 ;
<detph>^8 = <det>^8 <nouncla>^38;
<nouncla>^38 = <adj>^2 <cnoun>^36
              | <cnoun>^36;
<cnoun>^36 = man | men | person | people | planet | planets |  moon | moons | mountain | mountains |
          crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal|
              metals| nonmetal | nonmetals | country | countries | capital | capitals | city |cities |continent|
```

continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits| orbited | exist | exists ;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{121} = <pnoun_planet_moon_human>^{62}
| <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{33};
<pnoun_planet_moon_human>^{62} = earth | jupiter | mars | mercury | neptune | pluto | saturn | uranus | venus | almathea | ariel | callisto | charon | deimos | dione | enceladus | europa | ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh | jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth | jupiterthirteenth | jupitertwelfth | luna | mimas | miranda | nereid | oberon | phobos | phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel | bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens | kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ;
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium | hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{33} = <country>^{6} | <capital>^{6} | <city>^{6} | <continent>^{6} | <ocean>^{4} | <river>^{3} | <lake>^{1} | <mountain>^{1} ;
<country>^{6} = canada | china | England | France | Germany | united states;
<capital>^{6} = ottawa | Beijing | london | paris | berlin | washington;
<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{6} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;
<transvb>^{15} = orbit | orbits | discover | discovered | neighbour | neighbours | neighboured | worship | worshiped | contain | contains | contained | find | finds | found;
<preposition>^{2} = in | on ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does ;

The average branching factor for syntactic grammar
b = ((41+258+15+1+258)+(258+15+2+258)+(258+45+38+45)+(45)+(2+129)+(15+1+258)+(15+2+258)+(38)+(36)) / 24
= 2291 / 24
= 95.5

Figure Appendix C (2): branching-factor computation of syntactic grammar (Cont'd)

Figure Appendix C (3): branching-factor computation of extended semantic grammar

```
/* semantics_gram_ext2.gram */
grammar semantics_gram_ext2 ;
public <s>^42 = <linkingvb>^4 <termphrase_verbphrase>^2168
                | is^1 <pnoun>^395 <pnoun>^395
                | is^1 <pnoun>^395 ( a|an )^2 <nouncla>^44
                | is^1 <pnoun>^395 ( a|an )^2 <nouncla>^44 or^1 ( a|an )^2 <nouncla>^44
                | <quest1>^3 <sent>^294
                | ( who )^1 <animate_verbph>^8
                | ( what )^1 <inanimate_verbph>^45
                | ( which | how many )^2 <nouncla_verbph>^22
                | ( which | how many )^2 <nouncla_verbph_other>^60
                | <simple>^26 ;
<simple>^26 = | ask them to be quite
         | please introduce yourself
         | hello there
         | goodbye
         | goodbye solar man
         | fine thanks
         | thanks
         | thanks solar man
         | yes please
         | what is your name
         | who are you
         | where do youlive
         | what do youknow
         | how old are you
         | what is your favorite band
         | who is the vice president at the university of windsor
         | who is the dean of science at the university of windsor
         | tell me a poem
         | know any poems
         | tell me a joke
         | know any jokes
         | who is judy
         | can i talk to judy
         | can i talk to solar man
         | who is monty
         | can i talk to monty ;
<termphrase_verbphrase>^2168 = <nonhuman_termph_planet>^34 <transvb_by_termph>^15
                | <nonhuman_termph_moon>^88 <animate_transvb>^6 by^1 <human_termph>^50
                | <nonhuman_termph_other>^682 <animate_transvb>^6 by^1 <human_termph>^50
                | <nonhuman_termph_other>^682 <animate_transvb>^6 <preposition>^2
                      <nonhuman_termph_planet>^34
                | <nonhuman_termph_other>^682 <animate_transvb>^6 <preposition>^2
                      <nonhuman_termph_moon>^88 ;
<transvb_by_termph>^15 = <animate_transvb>^6 by^1 <human_termph>^50
                | <inanimate_transvb>^6 by^1 <nonhuman_termph_moon>^88
                | <inanimate_transvb_other>^3 by^1 <nonhuman_termph_other>^682 ;
<sent>^294 = <human_termph>^50 <animate_verbph>^8
```

| <nonhuman_termph_moon>^{88} <inanimate_verbph_active>$^{\underline{13}}$
| <nonhuman_termph_planet>^{34} <inanimate_verbph_passive>$^{\underline{17}}$
| <nonhuman_termph_moon>^{88} <inanimate_verbph_active_other>$^{\underline{5}}$
| <nonhuman_termph_planet>^{34} <inanimate_verbph_active_other>$^{\underline{5}}$;
<nouncla_verbph>^{22} = <human_nouncla>^{6} <animate_verbph>$^{\underline{8}}$
| <nonhuman_nouncla_moon>^{4} <animate_verbph_passive>$^{\underline{12}}$
| <nonhuman_nouncla_planet>^{4} <animate_verbph_passive>$^{\underline{12}}$
| <nonhuman_nouncla_moon>^{4} <inanimate_verbph_active>$^{\underline{13}}$
| <nonhuman_nouncla_planet>^{4} <inanimate_verbph_passive>$^{\underline{17}}$;
<nouncla_verbph_other>^{60} = <nonhuman_nouncla_other>^{30} <animate_verbph_passive>$^{\underline{12}}$
| <nonhuman_nouncla_other>^{30} <inanimate_verbph_passive_other>$^{\underline{10}}$;
<inanimate_verbph>^{45} = <inanimate_verbph_active>^{13}
| <inanimate_verbph_passive>^{17}
| <inanimate_verbph_active_other>^{5}
| <inanimate_verbph_passive_other>^{10};
<human_stermph>^{25} = <human_pnoun>^{17}
| <human_detph>^{8} ;
<nonhuman_stermph_planet>^{17} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_detph_planet>^{8} ;
<nonhuman_stermph_moon>^{44} = <nonhuman_pnoun_moon>^{36}
| <nonhuman_detph_moon>^{8} ;
<nonhuman_stermph_other>^{341} = <nonhuman_pnoun_other>^{333}
| <nonhuman_detph_other>^{8} ;
<human_termph>^{50} = <human_stermph>^{25}
| <human_stermph>^{25} (and | or)$^{\underline{2}}$ <human_stermph>$^{\underline{25}}$;
<nonhuman_termph_planet>^{34} = <nonhuman_stermph_planet>^{17}
| <nonhuman_stermph_planet>^{17} (and | or)$^{\underline{2}}$ <nonhuman_stermph_planet>$^{\underline{17}}$;
<nonhuman_termph_moon>^{88} = <nonhuman_stermph_moon>^{44}
| <nonhuman_stermph_moon>^{44} (and | or)^{2} <nonhuman_stermph_moon>$^{\underline{44}}$;
<nonhuman_termph_other>^{682} = <nonhuman_stermph_other>^{341}
| <nonhuman_stermph_other>^{341} (and | or)^{2} <nonhuman_stermph_other>$^{\underline{341}}$;
<animate_verbph>^{8} = <animate_transvbph>^{8} ;
<inanimate_verbph_active>^{13} = <inanimate_transvbph_active>^{6}
| <intransvb>^{7} ;
<inanimate_verbph_passive>^{17} = <inanimate_transvbph_passive>^{4}
| <intransvb>^{7}
| <inanimate_transvb>^{6} sun$^{\underline{1}}$;
<inanimate_verbph_active_other>^{5} = <inanimate_transvbph_active_other>^{3}
| <intransvb_other>^{2};
<inanimate_verbph_passive_other>^{10} = <inanimate_transvbph_passive_other>^{8}
| <intransvb_other>^{2};
<animate_verbph_passive>^{12} = <linkingvb>^{4} <animate_transvb>$^{\underline{6}}$ by$^{\underline{1}}$ <human_termph>$^{\underline{50}}$
| <linkingvb>^{4} <animate_transvb>$^{\underline{6}}$ <preposition>$^{\underline{2}}$ <nonhuman_termph_planet>$^{\underline{34}}$
| <linkingvb>^{4} <animate_transvb>$^{\underline{6}}$ <preposition>$^{\underline{2}}$ <nonhuman_termph_moon>$^{\underline{88}}$;
<animate_transvbph>^{8} = <animate_transvb>^{6} (<nonhuman_termph_planet>$^{\underline{34}}$
| <nonhuman_termph_moon>$^{\underline{88}}$
| <nonhuman_termph_other>$^{\underline{682}}$)
| <animate_transvb_other>^{2} (<human_termph>$^{\underline{50}}$
| <nonhuman_termph_planet>$^{\underline{34}}$
| <nonhuman_termph_moon>$^{\underline{88}}$
| <nonhuman_termph_other>$^{\underline{682}}$);
<inanimate_transvbph_active>^{6} = <inanimate_transvb>^{6} <nonhuman_termph_planet>$^{\underline{34}}$;

<inanimate_transvbph_passive>^{4} =
<linkingvb>^{4} <inanimate_transvb>^{6} by^{1} <nonhuman_termph_moon>^{88} ;
<inanimate_transvbph_active_other>^{3} =
<inanimate_transvb_other>^{3} <nonhuman_termph_other>^{682} ;
<inanimate_transvbph_passive_other>^{8} =
<linkingvb>^{4} <inanimate_transvb_other>^{3} by^{1} <nonhuman_termph_planet>^{34}
| <linkingvb>^{4} <inanimate_transvb_other>^{3} by^{1} <nonhuman_termph_moon>^{88} ;
<human_detph>^{96} = <det>^{8} <human_nouncla>^{6} ;
<nonhuman_detph_planet>^{8} = <det>^{8} <nonhuman_nouncla_planet>^{4} ;
<nonhuman_detph_moon>^{8} = <det>^{8} <nonhuman_nouncla_moon>^{4} ;
<nonhuman_detph_other>^{8} = <det>^{8} <nonhuman_nouncla_other>^{30} ;
<preposition>^{2} = on | in ;
<nouncla>^{44} = <human_nouncla>^{6}
| <nonhuman_nouncla_planet>^{4}
| <nonhuman_nouncla_moon>^{4}
| <nonhuman_nouncla_other>^{30};
<human_nouncla>^{6} = <adj>^{2} <human_cnoun>^{4}
| <human_cnoun>^{4} ;
<nonhuman_nouncla_planet>^{4} = <adj>^{2} <nonhuman_cnoun_planet>^{2}
| <nonhuman_cnoun_planet>^{2} ;
<nonhuman_nouncla_moon>^{4} = <adj>^{2} <nonhuman_cnoun_moon>^{2}
| <nonhuman_cnoun_moon>^{2} ;
<nonhuman_nouncla_other>^{30} = <adj>^{2} <nonhuman_cnoun_other>^{28}
| <nonhuman_cnoun_other>^{28} ;
<human_cnoun>^{4} = man | men | person | people;
<nonhuman_cnoun_planet>^{2} = planet | planets ;
<nonhuman_cnoun_moon>^{2} = moon | moons;
<nonhuman_cnoun_other>^{28} = mountain | mountains | crater | craters | sea | seas | ocean | oceans |
chemical | chemicals | gas | gases | metal | metals | nonmetal | nonmetals |
country | countries | capital | capitals | city | cities | continent |
continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits | orbited | exist |exists ;
<intransvb_other>^{2} = exist | exists;
<animate_transvb>^{6} = discover | discovers | discovered | find | finds | found ;
<animate_transvb_other>^{2} = worship | worshiped;
<inanimate_transvb>^{6} = orbit | orbits | orbited | neighbour | neighbours | neighboured;
<inanimate_transvb_other>^{3} = contain | contains | contained ;
<linkingvb>^{4} = is | was | are | were ;
<quest1>^{3} = did | do | does;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{395} = <nonhuman_pnoun_planet>^{9}
| <nonhuman_pnoun_moon>^{36}
| <human_pnoun>^{17}
| <nonhuman_pnoun_other>^{333};
<nonhuman_pnoun_planet>^{9} = urth | jupiter | mars | mercury | neptune | pluto | saturn | uranus | venus ;
<nonhuman_pnoun_moon>^{36} = almathea | ariel |callisto | charon | deimos | dione | enceladus |
europa | ganymede | hyperion | iapetus | io | janus | jupitereighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas| miranda | nereid | oberon | phobos |
phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel ;
<human_pnoun>^{17} = bernard | bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens |

kowal | kuiper | larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_other>^{333} = <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{307};
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ;
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{307} = <country>^{187} | <capital>^{98} | <city>^{6} | <continent>^{7} | <ocean>^{4} |
<river>^{3} | <lake>^{1} | <mountain>^{1} ;
<country>^{187} = Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguilla |
Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan |
Bahrain |Bangladesh | Barbados | Bassas da India | Belarus | Belgium | Belize | Benin |
Bermuda | Bhutan | Bolivia |Bosnia and Herzegovina | Botswana |Bouvet Island | Brazil|
Brunei |Bulgaria | Burkina Faso | Burma | Burundi | Cambodia | Caneriib | Canada |
Cape Verde | Cayman Islands | Central African Republic | Chad | Chile |
China | Clipperton Island | Colombia | Comoros | Congo Democratic Republic |
Congo Republic | Cook Islands | Coral Sea Islands | Costa Rica | Croatia | Cuba | Cyprus |
Czech Republic | Denmark | Djibouti | Dominica | Dominica Republic | Ecuador | Egypt |
El Salvador | Equatorial Guinea | Eritrea | Estonia | Ethiopia | Europe Island | Fiji | Finland |
France |French Guiana | Gabon | Gambia | Gaza Strip |Georgia | Germany | Ghana |
Gibraltar | Glorioso Island | Greece | Greenland | Grenada | Guadeloupe | Guam |
Guatemala | Guernsey | Guinea | Guyana | Haiti | Heard and Mcdonald Island |Holy See |
Honduras | Howland Island | Hungary |Iceland |India |Indonesia | Iran |Iraq | Ireland | Israel|
Italy | Jamaica | Jan Mayen |Japan | Jarvis Island | Jersey | Johnston Atoll | Jordan |
Kazakhstan | Kenya |Kingman Reef | Kiribati | North Korea | South Korea | Kuwait |
KyrgyzStan | Laos | Latvia |Lebanon | Lesotho |Liberia |Libya | Liechtenstein | Lithuania |
Luxembourg | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta |
Isle of Man | Marshall Islands | Martinique | Mauritania |Mauritius | Mayotte| Mexico|
Micronesia | Midway Island |Moldova | Monaco | Mongolia | Montserrat | Morocco |
Mozambique |Myanmar | Netherlands | Norway | New Zealand | Nigeria | Oman | Portugal |
Poland | Romania | Russia | Rwanda | Tajikistan | Tanzania | Syria | Swede | Switzerland |
Sudan | Spain | Singapore | Thailand | Togo |Tokelau | Tonga |Tunisia | Turkey |
Turkmenistan | Tuvalu | Uganda | Ukraine |united Arab Emirates | United Kingdom |
United States of Amerima |Uruguay | Uzbekistan |Vietnam | Yemen | Yugoslavia |Zambia|
Zimbabwe ;
<capital>^{98} = ottawa | Beijing | london | paris | berlin | Washington| Kabul |Tirana |Algiers |
Pago Pago | Luanda | Andorra la Vella | Buenos Aires | Yerevan | Oranjestad | Canberra |
Vienna | Baku| Dhaka |Manama | BridgeTown | Brussels | Belmopan | Portonovo |
Hamilton | Thimphu | LaPaz |Gaborone |Brasilia| Phnom Penh |Yaounde | Praia | Prague |
Santiago | Bogota | Moroni | Havana | Nicosia | Copenhagen | Roseau | Cairo | Asmara|
Addis Ababa| Suva | Helsinki | Libreville | Banjul | GoregeTown | Tbilisi |Accra | Athens |
Saint George's | Conakry | Port-au-prince | Budapest | New Delhi | Jakarta | Tehran |
Baghdad | Dublin | Jerusalem | Rome | Tokyo | Amman | PYong Yang | Seoul | Kuwait |
Beirut | Maseru | Monrovia | Tripoli | Skopje | Amsterdam | Kuala Lumpur | Bamako |
Velletta | Mexico | Ulaanbaatar | Windhoek | Abuja | Wellington | Oslo | Warsaw | Lisbon |
Moscow | Stockholm | Bucharest | Singapore | Madrid | Khartoum | Bern | Damascus |
Hanoi |Ankara | Sanaa | Harare | Belgrade | Lusaka ;

<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;
<continent>^{7} = Africa | Asia | Austrilia | Europe | North America | South America;
<ocean>^{4} = Arctic | Atlantic | India | Pacific;
<river>^{3} = Yangtse | Nile | Danube ;
<lake>^{1} = ontario lake;
<mountain>^{1} = rocky mountain;

The average branching factor for extended semantic grammar
b = ((42+2168)+(395+395)+(395+2+44)+(395+2+44+1+2+44)+(294)+(8+45+22+60)+
(15)+(6+1+50)+(6+1+50)+(6+2+34)+(6+2+88)+(1+50)+(1+88+1+682)+(8+13+17+5+5)+
(8+12+12+13+17)+(12+10)+(2+25)+(2+17)+(2+44)+(2+341)+1+(6+1+50)+(6+2+34)+(6+2+88)+
(34+88+682)+(50+34+88+682)+34+(6+1+88)+682+(3+1+34)+(3+1+88)+(6+4+4+30)+
(4+2+2+28)) / 93
= 8890 / 93
= 95.6

Figure Appendix C (3): branching-factor computation of extended semantic grammar (Cont'd)

Figure Appendix C (4): branching-factor computation of extended syntactic grammar

/* syntax_gram_ext2.gram */
grammar syntax_gram_ext2 ;
public <s>^{41} = <linkingvb>^{4} <termph>^{808} [<transvb>^{15} by^{1}] <termph>^{808}
| <linkingvb>^{4} <termph>^{808} [<transvb>^{15} <preposition>^{2}] <termph>^{808}
| <quest1>^{3} <sent>^{808}
| (who |what) ^{2} <verbph>^{45}
| (which | how many) ^{2} <nouncla>^{38}<verbph>^{45}
| <simple>^{26} ;
<simple>^{26} = | ask them to be quite
| please introduce yourself
| hello there
| goodbye
| goodbye solar man
| fine thanks
| thanks
| thanks solar man
| yes please
| what is your name
| who are you
| where do youlive
| what do youknow
| how old are you
| what is your favorite band
| who is the vice president at the university of windsor
| who is the dean of science at the university of windsor
| tell me a poem
| know any poems
| tell me a joke
| know any jokes
| who is judy

| can i talk to judy
| can i talk to solar man
| who is monty
| can i talk to monty ;
<sent>^{808} = <termph>^{808} <verbph>$^{\underline{45}}$;
<stermph>^{404} = <pnoun>^{396}
| <detph>^{8};
<termph>^{808} = <stermph>^{404}
| <stermph>^{404} (and | or)$^{\underline{2}}$ <stermph>$^{\underline{404}}$;
<verbph>^{45} = <transvbph>^{38}
| <intransvb>^{7};
<transvbph>^{38} = (<transvb>^{15} | <linkingvb>^{4} <transvb>$^{\underline{15}}$ by$^{\underline{1}}$) <termph>$^{\underline{808}}$
| (<transvb>^{15} | <linkingvb>^{4} <transvb>$^{\underline{15}}$ <preposition>$^{\underline{2}}$) <termph>$^{\underline{808}}$;
<detph>^{8} = <det>^{8} <nouncla>$^{\underline{38}}$;
<nouncla>^{38} = <adj>^{2} <cnoun>$^{\underline{36}}$
| <cnoun>^{36};
<cnoun>^{36} = man | men | person | people | planet | planets | moon | moons | mountain | mountains |
crater | craters | sea | seas | ocean | oceans | chemical | chemicals | gas | gases | metal |
metals | nonmetal | nonmetals | country | countries | capital | capitals | city | cities |
continent | continents | river | rivers | lake | lakes ;
<adj>^{2} = red | atmospheric;
<intransvb>^{7} = spin | spins | orbit | orbits| orbited | exist | exists ;
<det>^{8} = a | an | every | one | two | three | four | five;
<pnoun>^{396} = <pnoun_planet_moon_human>^{63}
| <nonhuman_pnoun_chemical>^{20}
| <space_program>^{6}
| <earth_geography_domain>^{307};
<pnoun_planet_moon_human>^{63} = urth | jupiter | mars | mercury | neptune | pluto | saturn | uranus |
venus | almathea | ariel | callisto | charon | deimos | dione | enceladus | europa |
ganymede | hyperion | iapetus | io | janus | jupiter eighth | jupitereleventh |
jupiterfourteenth | jupiterninth | jupiterseventh | jupitersixth | jupitertenth |
jupiterthirteenth | jupitertwelfth | luna | mimas | miras | miranda | nereid | oberon |
phobos | phoebe | rhea | saturnfirst | tethys | titan | titania | triton | umbriel | bernard|
bond | cassini | dollfus | fountain | galileo | hall | herschel | huygens | kowal |kuiper|
larsen | lassell | melotte | nicholson | perrine | pickering ;
<nonhuman_pnoun_chemical>^{20} = <nonhuman_pnoun_gas>^{6}
| <nonhuman_pnoun_metal>^{9}
| <nonhuman_pnoun_nonmetal>^{5} ;
<nonhuman_pnoun_gas>^{6} = oxygen | hydrogen | nitrogen | dioxide | monoxide | helium ;
<nonhuman_pnoun_metal>^{9} = gold | silver | copper | iron | stannum | nickel | potassium | natrium |
hydrargyrum ;
<nonhuman_pnoun_nonmetal>^{5} = water | sulphur | carbon | phosphorus | calcium;
<space_program>^{6} = shuttle | rocket | launch | telescope | station | astronaut;
<earth_geography_domain>^{307} = <country>^{187} | <capital>^{98} | <city>^{6} | <continent>^{7} | <ocean>^{4} |
<river>^{3} | <lake>^{1} | <mountain>^{1} ;
<country>^{187} = Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Anguilla |
Antigua and Barbuda | Argentina | Armenia | Aruba | Australia | Austria | Azerbaijan |
Bahrain |Bangladesh | Barbados | Bassas da India | Belarus | Belgium | Belize | Benin |
Bermuda | Bhutan | Bolivia |Bosnia and Herzegovina | Botswana |Bouvet Island | Brazil|
Brunei |Bulgaria | Burkina Faso | Burma | Burundi | Cambodia | Caneriib | Canada |
Cape Verde | Cayman Islands | Central African Republic | Chad | Chile |
China | Clipperton Island | Colombia | Comoros | Congo Democratic Republic |

Congo Republic | Cook Islands | Coral Sea Islands | Costa Rica | Croatia | Cuba | Cyprus |
Czech Republic | Denmark | Djibouti | Dominica | Dominica Republic | Ecuador | Egypt |
El Salvador | Equatorial Guinea | Eritrea | Estonia | Ethiopia | Europe Island | Fiji |Finland|
France |French Guiana | Gabon | Gambia | Gaza Strip |Georgia | Germany | Ghana |
Gibraltar | Glorioso Island | Greece | Greenland | Grenada | Guadeloupe | Guam |
Guatemala | Guernsey | Guinea | Guyana | Haiti | Heard and Mcdonald Island |Holy See |
Honduras | Howland Island | Hungary |Iceland |India |Indonesia | Iran |Iraq | Ireland | Israel|
Italy | Jamaica | Jan Mayen |Japan | Jarvis Island | Jersey | Johnston Atoll | Jordan |
Kazakhstan | Kenya |Kingman Reef | Kiribati | North Korea | South Korea | Kuwait |
KyrgyzStan | Laos | Latvia |Lebanon | Lesotho |Liberia |Libya | Liechtenstein | Lithuania |
Luxembourg | Macedonia | Madagascar | Malawi | Malaysia | Maldives | Mali | Malta |
Isle of Man | Marshall Islands | Martinique | Mauritania |Mauritius | Mayotte| Mexico|
Micronesia | Midway Island |Moldova | Monaco | Mongolia | Montserrat | Morocco |
Mozambique |Myanmar | Netherlands | Norway | New Zealand | Nigeria |
Oman | Portugal | Poland | Romania | Russia | Rwanda | Tajikistan | Tanzania |
Syria | Swede | Switzerland | Sudan | Spain | Singapore |
Thailand | Togo |Tokelau | Tonga |Tunisia | Turkey | Turkmenistan |Tuvalu | Uganda |
Ukraine |united Arab Emirates | United Kingdom | United States of Amerima | Uruguay |
Uzbekistan |Vietnam | Yemen | Yugoslavia |Zambia | Zimbabwe ;

<capital>^{98} = ottawa | Beijing | london | paris | berlin | Washington |
Kabul | Tirana |Algiers | Pago Pago | Luanda | Andorra la Vella | Buenos Aires |
Yerevan | Oranjestad | Canberra | Vienna | Baku | Dhaka |Manama | BridgeTown |
Brussels | Belmopan | Portonovo | Hamilton | Thimphu | LaPaz |Gaborone |Brasilia|
Phnom Penh |Yaounde | Praia | Prague | Santiago | Bogota | Moroni | Havana |
Nicosia | Copenhagen | Roseau | Cairo | Asmara | Addis Ababa | Suva | Helsinki |
Libreville | Banjul | GoregeTown | Tbilisi |Accra | Athens | Saint George's | Conakry |
Port-au-prince | Budapest | New Delhi | Jakarta | Tehran | Baghdad | Dublin |
Jerusalem | Rome | Tokyo | Amman | PYong Yang | Seoul | Kuwait | Beirut |
Maseru | Monrovia | Tripoli | Skopje | Amsterdam | Kuala Lumpur | Bamako |
Velletta | Mexico | Ulaanbaatar | Windhoek | Abuja | Wellington | Oslo |
Warsaw | Lisbon | Moscow | Stockholm | Bucharest | Singapore | Madrid |
Khartoum | Bern | Damascus | Hanoi |Ankara | Sanaa | Harare | Belgrade | Lusaka ;

<city>^{6} = toronto | shanghai | manchester | lyon | Frankfurt | New York;

<continent>^{7} = Africa | Asia | Austrilia | Europe | North America | South America | Antarctica;

<ocean>^{4} = Arctic | Atlantic | India | Pacific;

<river>^{3} = Yangtse | Nile | Danube ;

<lake>^{1} = ontario lake;

<mountain>^{1} = rocky mountain;

<transvb>^{15} = orbit | orbits | discover | discovered | neighbour | neighbours | neighboured | worship | worshiped | contain | contains | contained | find | finds | found;

<preposition>^{2} = in | on ;

<linkingvb>^{4} = is | was | are | were ;

<quest1>^{3} = did | do | does ;

The average branching factor for extended syntactic grammar

$b = ((41+808+15+1+808)+(808+15+2+808)+(808+45+38+45)+45+(2+404)+(15+1+808+15+2+808)+38+36) / 24$

$= 6416 / 24$

$= 267.3$

Figure Appendix C (4): branching-factor computation of extended syntactic grammar (Cont'd)

Appendix D: Partial Experiment Result in Detail

(1) Notes for the experiment:

Semantics set: a set of utterances that are both semantically and syntactically correct.

Syntax set: a set of utterances that are syntactically correct, but semantically incorrect.

Word-sequence set: word sequences that are neither semantically, nor syntactically correct, but consist of words from the defined vocabulary.

(2) Notations for recording experiment-recognition results.

C: recognized Correctly;

I: recognized Incorrectly;

N: Not recognized at all.

(3) Note for the heading line in the table. The heading line indicates which grammar is u sed, the testing order, etc.

sem: semantic grammar;

syn: syntactic grammar;

wd seq: word-sequence grammar.

sem ext: extended semantic grammar;

syn ext: extended syntactic grammar;

wd seq ext: extended word-sequence grammar.

i: testing order

(n/m) : *n* utterances are correctly recognized out of *m* utterances.

(N: x, I: y): *x* utterances are not recognized at all, *y* utterances are recognized incorrectly.

e.g.: Sem #1 (60/73) (N:10, I: 3)

means the semantic grammar was the first grammar to be tested, 60 utterances were recognized correctly out of total 73 utterances, 10 utterances were not recognized at all, 3 utterances were recognized incorrectly.

Table Appendix D (1): experiment result of Person #1 on grammars before extended using semantics set

Person #1 (English Male):

No	Testing Utterances	Sem #1(60/73) (N:10, I: 3)	Sem #3(60/73) (N:9, I:4)	Syn #2(60/73) (N:2, I:11)	Syn #4(57/73) (N:5, I: 11)
1.	Was phobos discovered by a person	C	C	C	N
2.	Is titania a mountain	C	C	C	C
3.	Is cassini a moon	C	C	C	C
4.	Is pluto a mountain or a moon	C	C	C	C
5.	Is pluto an atmospheric crater	C	C	C	C
6.	Does pluto exist	C	C	C	C
7.	Does ariel neighbour pluto	C	C	C	C
8.	Does a moon neighbour a planet	N	C	C	I
9.	Does every person worship a planet	C	C	C	C
10.	Does saturn contain a crater	C	C	C	C
11.	Does phobos contain a red mountain	C	C	C	C
12.	Does janus contain nitrogen	C	C	C	C
13.	Did bernard discover a mountain	N	C	C	C
14.	Who discovered a crater	C	C	C	C
15.	Which mountain is found on uranus	C	C	C	C
16.	Which gas is found on a moon	C	C	C	C
17.	What is contained by venus	C	C	C	C
18.	What is contained by phobos	C	C	C	C
19.	Which mountain is found on janus	C	C	C	C
20.	Which sea exists	C	C	C	C
21.	Which mountains are discovered by hall	C	C	C	C
22.	Which moon orbits a planet	N	C	C	C
23.	How many moons neighbour saturn	C	C	C	C

24.	Was neptune discovered by dollfus or cassini	C	C	C	C
25.	Does triton orbit pluto or saturn	C	C	C	C
26.	Was neptune contained by hydrogen or nitrogen	C	C	N	C
27.	Does jupitereighth contain a sea or a mountain	I	N	N	N
28.	Does jupiter contain hydrogen or oxygen	C	C	C	C
29.	Does earth contain oxygen	C	C	I	I
30.	Does a moon contain hydrogen	C	C	C	C
31.	Does a moon neighbour a planet	C	C	C	I
32.	How many gases are found on mars	C	I	I	I
33.	How many craters are found on a moon	C	C	C	C
34.	How many oceans are discovered by hall	C	C	C	C
35.	How many mountains are found on earth	C	N	I	I
36.	Is gold found on earth	C	I	I	I
37.	Is silver found on janus	C	C	C	C
38.	Is a chemical found on triton	C	I	I	C
39.	Is dioxide found on phoebe	C	C	C	C
40.	Is sulphur found on luna	C	C	C	C
41.	Is oxygen found on mars	I	I	I	I
42.	Is a metal found on a planet	C	C	C	C
43.	Is a nonmetal found on pluto	C	C	C	C
44.	Is a river found on neptune	C	C	C	C
45.	Is a lake found on venus	C	C	C	C
46.	Which gas is found on titan	C	C	C	C
47.	Which chemicals are found on rhea	N	C	C	C
48.	Which nonmetals are found on jupiter	C	C	C	C
49.	Which metals are found on a moon	C	C	C	C
50.	Which river is found on hyperion	C	C	C	C

51.	Which mountains are found on rhea	C	C	C	C
52.	How many chemicals are found on pluto	N	N	C	C
53.	How many metals are found on a moon	C	C	C	C
54.	How many nonmetals are found on jupiter	N	N	C	C
55.	How many gases are found on mars	C	C	I	I
56.	How many continents are found on earth	C	N	I	I
57.	Is berlin a capital	C	C	C	C
58.	Is beijing a city	C	C	C	C
59.	Is lyon a moon	C	C	C	C
60.	Is india an ocean or a country	C	C	C	N
61.	Is canada a mountain	C	C	C	C
62.	Is england an atmospheric planet	C	C	C	C
63.	Which mountain is found on jupiter	I C	C	C	C
64.	Which rivers are found on io	C	C	C	C
65.	Which nonmetals are found on a planet	N	N	C	N
66.	Which gases are found on a moon	C	C	C	C
67.	Is an ocean found on mercury	C	C	C	C
68.	How many rivers are found on miranda	C	C	C	C
69.	How many chemicals are found on phoebe	N	N	C	C
70.	How many continents are found on earth	N	N	I	N
71.	Is an ocean found on mercury	C	C	C	C
72.	How many gases are contained by earth	N	C	I	I
73.	How many gases are found on earth	I	N	I	I

Table Appendix D (1): experiment result of Person #1 on grammars before extended using semantics set (Cont'd)

Table Appendix D (2): experiment result of Person #1 on grammars before extended using syntax set

Person #1 (English Male):

No	Testing Utterances	Sem #1 (0/25) (N:20, I:5)	Sem #3 (0/25) (N: 17, I:8)	Syn #2 (22/25) (N: 3, I:0)	Syn #4 (22/25) (N:3, I: 0)
1	Does a mountain contain a moon	N	N	C	C
2	Does a gas contain a planet	N	N	C	C
3	Does a river contain a continent	N	I	C	C
4	Was phobos discovered by a moon	I	I	C	C
5	Does water contain a river	N	N	C	C
6	Is a crater found in nitrogen	N	I	C	C
7	Does ariel neighbour hall	N	N	C	C
8	Does a moon neighbour a people	N	N	C	N
9	Does a crater contain saturn	N	N	C	C
10	Does a red mountain contain phobos	I	N	C	C
11	Does nitrogen contain janus	N	N	C	C
12	Does berlin discover a moon	N	N	C	C
13	Which mountain is found on bond	N	I	C	C
14	Which moon is found in a gas	I	I	C	C
15	Which mountains are discovered by pacific	I	I	C	C
16	Which river orbits a planet	N	N	C	C
17	How many people neighbour Saturn	N	N	C	C
18	Was neptune discovered by dollfus or lyon	N	I	C	C
19	Does triton orbit pluto or frankfurt	N	N	N	C
20	Does gold contain a sea or a mountain	N	I	N	N
21	How many moons are found in atlantic	N	N	C	C
22	How many craters are discovered by nile	I	N	C	C
23	Is gold found in cassini	N	N	C	C

24	Which chemicals are found on bond	N	N	C	C
25	How many chemicals are found on galileo	N	N	N	N

Table Appendix D (2): experiment result of Person #1 on grammars before extended using syntax set (Cont'd)

Table Appendix D (3): experiment result of Person #2 on grammars before extended using semantics set

Person #2 (non-English Female):

No	Testing Utterances	Sem #1 (48 /73) N:23, I: 2	Sem #4 (52 /73) N:19, I: 2	Syn #2 (36 /73) N:27,I: 10	Syn #5 (41/73) N:22 I:10	Wd Seq #3(9/73) N:18 I:46	Wd Seq #6(9/73) N:23 I:41
1	Was phobos discovered by a person	N	N	N	N	I	I
2	Is titania a mountain	C	C	N	N	I	N
3	Is cassini a moon	C	C	C	C	C	C
4	Is pluto a mountain or a moon	N	N	N	N	N	I
5	Is pluto an atmospheric crater	C	C	C	C	I	I
6	Does pluto exist	C	C	C	C	C	C
7	Does ariel neighbour pluto	C	C	N	N	I	N
8	Does a moon neighbour a planet	C	C	N	I	C	I
9	Does every person worship a planet	C	C	C	C	I	C
10	Does saturn contain a crater	I	C	I	I	C	I
11	Does phobos contain a red mountain	C	C	C	C	I	I
12	Does janus contain nitrogen	C	C	C	I	C	I
13	Did bernard discover a mountain	C	N	C	I	N	N
14	Who discovered a crater	C	C	C	C	I	C
15	Which mountain is found on uranus	N	C	C	C	I	N
16	Which gas is found on a moon	C	C	C	C	I	I
17	What is contained by venus	N	N	N	C	I	I
18	What is contained by phobos	C	N	N	C	C	I

19	Which mountain is found on janus	C	C	C	C	I	I
20	Which sea exists	C	C	C	C	C	C
21	Which mountains were discovered by hall	N	N	N	N	I	I
22	Which moon orbits a planet	C	C	C	C	N	C
23	How many moons neighbour saturn	C	C	I	I	I	I
24	Was neptune discovered by dollfus or cassini	N	N	N	N	N	I
25	Does triton orbit pluto or saturn	N	N	N	N	N	N
26	Does neptune contain hydrogen or nitrogen	N	N	N	C	I	I
27	Does phobos contain a sea or a mountain	N	C	C	C	I	I
28	Does phoebe contain hydrogen or oxygen	N	N	N	N	I	I
29	Does oberon contain oxygen	C	C	C	C	I	C
30	Does a moon contain hydrogen	C	C	C	C	I	I
31	Does a moon neighbour a planet	C	C	C	C	I	C
32	How many gases are found on mars	C	C	N	N	I	I
33	How many craters are found on a moon	C	C	C	C	I	I
34	How many oceans were discovered by hall	N	C	N	N	I	N
35	How many mountains are found on earth	C	C	I	I	I	N
36	Is gold found on earth	C	C	I	I	I	I
37	Is silver found on janus	C	C	C	C	I	I
38	Is a chemical found on triton	C	C	C	C	N	N
39	Is dioxide found on phoebe	C	C	C	C	N	N
40	Is sulphur found on luna	C	C	C	C	I	N
41	Is oxygen found on mars	C	C	I	C	I	I
42	Is a metal found on a planet	C	C	C	N	N	N
43	Is a nonmetal found on Pluto	C	C	I	C	I	I
44	Is a river found on Neptune	N	N	N	N	I	N
45	Is a lake found on venus	C	C	C	C	I	N

46	Which gas is found on titan	C	C	C	C	I	I
47	Which chemicals are found on rhea	C	C	N	N	N	N
48	Which nonmetals are found on jupiter	N	C	C	I	N	I
49	Which metals are found on a moon	C	C	N	C	I	N
50	Which river is found on hyperion	N	N	N	N	N	N
51	Which mountains are found on rhea	C	N	I	C	I	I
52	How many chemicals are found on pluto	C	C	C	N	N	I
53	How many metals are found on a moon	C	I	C	C	N	N
54	How many nonmetals are found on jupiter	C	N	C	C	I	N
55	How many gases are found on mars	N	C	I	N	I	I
56	How many continents are found on charon	C	C	N	I	I	I
57	Is berlin a capital	I	I	I	C	I	I
58	Is beijing a city	C	C	C	C	I	I
59	Is lyon a moon	C	C	C	C	C	C
60	Is india an ocean or a country	N	C	N	N	I	N
61	Is canada a mountain	C	C	C	C	I	I
62	Is england an atmospheric planet	N	N	N	N	C	N
63	Which mountain is found on jupiter	C	I	C	C	N	I
64	Which rivers are found on io	N	N	N	N	N	I
65	Which nonmetals are found on a planet	N	C	N	C	N	N
66	Which gases are found on a moon	N	C	N	C	I	N
67	Is an ocean found on mercury	C	C	C	C	N	N
68	How many rivers are found on miranda	N	N	N	N	I	I
69	How many chemicals are found on phoebe	C	C	C	C	I	I
70	How many continents are found on earth	C	C	I	N	I	I
71	Is an ocean found on mercury	C	C	C	C	N	N
72	How many gases are contained by earth	N	N	N	N	I	I

73	How many gases are found on earth	N	N	N	I	I	I

Table Appendix D (3): experiment result of Person #2 on grammars before extended using semantics set (Cont'd)

Table Appendix D (4): experiment result of Person #2 on grammars before extended using syntax set

Person #2 (non-English Female):

No	**Testing Utterances**	**Sem #1** (0/25) (N:21, I:4)	**Sem #4(0 /25)** (N:20, I: 5)	**Syn #2** (9/25) (N: 12 I: 4)	**Syn #5** (12/25) (N: 12, I: 1)	**Wd Seq #3** (2/25) (N:13, I:10)	**Wd Seq #6** (2/25) (N:11, I:12)
1	Does a mountain contain a moon	N	N	N	C	I	N
2	Does a gas contain a planet	N	N	C	N	N	N
3	Does a river contain a continent	N	N	N	C	N	N
4	Was phobos discovered by a moon	N	N	N	N	I	I
5	Does water contain a river	N	N	N	N	N	N
6	Is a crater found in nitrogen	I	I	C	N	N	N
7	Does ariel neighbour hall	N	N	I	C	I	C
8	Does a moon neighbour a person	N	N	C	C	I	I
9	Does a crater contain saturn	N	N	I	N	N	N
10	Does a red mountain contain phobos	N	N	C	N	I	I
11	Does nitrogen contain janus	N	I	C	C	I	I
12	Did berlin discover a moon	I	I	I	C	C	C
13	Which mountain is found on bond	I	I	C	C	I	I
14	Which moon is found in a gas	I	I	C	C	I	I
15	Which mountains were discovered by pacific	N	N	N	N	N	N
16	Which river orbits a planet	N	N	N	C	N	N
17	How many people neighbour saturn	N	N	C	I	C	I
18	Was neptune discovered by dollfus or lyon	N	N	N	N	N	I
19	Does triton orbit pluto or frankfurt	N	N	N	N	N	N

20	Does gold contain a sea or a mountain	N	N	N	N	N	I
21	How many moons are found in atlantic	N	N	N	N	N	I
22	How many craters were discovered by nile	N	N	N	C	I	I
23	Is gold found in cassini	N	N	I	C	I	N
24	Which chemicals are found on bond	N	N	C	C	N	I
25	How many chemicals are found on galileo	N	N	N	N	N	N

Table Appendix D (4): experiment result of Person #2 on grammars before extended using syntax set (Cont'd)

Table Appendix D (5): experiment result of Person #2 on grammars before extended using word-sequence set

Person #2 (non-English Female):

No	**Testing Utterances**	**Sem #1 (0/24)** (N:21, I:3)	**Sem #4 (0/24)** (N:22, I:2)	**Syn #2 (0/24** (N: 17, I:7)	**Syn #5 (0/24** (N:17 I:7)	**Word Seq #3(4/24** (N:8 I:12)	**Word Seq #6 (3 /24** (N:6 I: 15
1	Is a mountain contain a moon	N	N	N	N	I	N
2	Does a gas a planet	N	N	N	N	I	N
3	Is a river found a continent	N	N	N	N	N	I
4	Phobos discovered by a moon	N	N	N	N	I	I
5	Does water exist a river	N	N	N	N	N	I
6	Is a crater contain nitrogen	N	N	I	N	C	C
7	Is ariel neighbour a planet	N	N	I	N	N	N
8	Is a moon discover a people	N	N	I	I	C	N
9	Which crater contain on saturn	N	N	N	I	N	I
10	Is a red phobos contain a mountain	N	N	N	N	I	I
11	Is janus contain nitrogen	I	I	I	I	C	C
12	Is jupiter discovered bernard	N	N	N	N	N	N
13	Which mountain is found dione and phoebe	N	N	N	N	N	I
14	Which gas found moon	I	N	I	N	I	I

15	Which mountain discovered by metotte	N	N	N	N	I	I
16	Which moon orbits on a planet	N	N	N	N	I	I
17	How many moons neighbour on saturn	N	N	N	I	I	I
18	Was neptune discovered dollfus and kowal	N	N	N	N	C	C
19	Is triton orbit pluto or venus	N	N	N	N	N	I
20	Is gold contained a moon	N	N	I	I	I	I
21	How many mountains found on oberon	I	N	N	N	I	I
22	How many craters are found earth	N	N	N	I	I	I
23	Is gold found cassini	N	N	I	I	I	I
24	Which chemicals are found bond	N	I	N	N	N	N

Table Appendix D (5): experiment result of Person #2 on grammars before extended using word-sequence set (Cont'd)

Table Appendix D (6): experiment result of Person #1 on extended grammars using semantics set

Person #1 (English Male):

No	Testing Utterances	Sem ext #1 (63/73)(N:6 I:4)	Sem ext #3 (66 /73)(N:5, I:2)	Syn ext #2 (58 /73)(N:2I:13)	Syn ext 4 (54/73)(N:4 I:15)
1	Was phobos discovered by a person	C	C	N	C
2	Is titania a mountain	C	C	C	C
3	Is cassini a moon	C	C	C	C
4	Is pluto a mountain or a moon	N	C	C	C
5	Is pluto an atmospheric crater	C	C	C	C
6	Does pluto exist	C	C	C	C
7	Does ariel neighbour pluto	C	C	C	C
8	Does a moon neighbour a planet	C	N	C	C
9	Does every person worship a planet	I	N	C	C
10	Does saturn contain a crater	I	C	C	C
11	Does phobos contain a red mountain	C	C	C	C

12	Does janus contain nitrogen	C	C	C	C
13	Did bernard discover a mountain	C	C	I	I
14	Who discovered a crater	C	C	I	C
15	Which mountain is found on uranus	C	C	C	C
16	Which gas is found on a moon	C	C	C	C
17	What is contained by venus	C	C	C	C
18	What is contained by phobos	C	C	C	C
19	Which mountain is found on janus	C	C	C	C
20	Which sea exists	C	C	C	C
21	Which mountains were discovered by hall	C	C	C	C
22	Which moon orbits a planet	C	C	C	C
23	How many moons neighbour saturn	C	C	C	C
24	Was neptune discovered by dollfus or cassini	C	C	C	C
25	Does triton orbit pluto or saturn	C	C	C	C
26	Does neptune contain hydrogen or nitrogen	C	C	C	C
27	Does jupitereighth contain a sea or a mountain	N	C	C	I
28	Does jupiter contain hydrogen or oxygen	C	C	C	C
29	Does earth contain oxygen	C	C	C	N
30	Does a moon contain hydrogen	C	C	C	C
31	Does a moon neighbour a planet	C	C	C	I
32	How many gases are found on mars	I	C	I	N
33	How many craters are found on a moon	C	C	C	C
34	How many oceans were discovered by hall	C	C	C	C
35	How many mountains are found on earth	C	I	I	I
36	Is gold found on earth	C	C	I	I
37	Is silver found on janus	C	C	C	I
38	Is a chemical found on triton	I	I	I	I

39	Is dioxide found on phoebe	C	C	C	C
40	Is sulphur found on luna	C	C	C	C
41	Is oxygen found on mars	C	C	C	I
42	Is a metal found on a planet	C	C	C	C
43	Is a nonmetal found on pluto	C	C	C	C
44	Is a river found on neptune	C	C	C	C
45	Is a lake found on venus	C	C	C	C
46	Which gas is found on titan	C	C	C	C
47	Which chemicals are found on rhea	C	C	C	C
48	Which nonmetals are found on jupiter	C	C	C	I
49	Which metals are found on a moon	C	C	C	C
50	Which river is found on hyperion	C	C	C	C
51	Which mountains are found on rhea	C	N	C	C
52	How many chemicals are found on pluto	N	N	N	C
53	How many metals are found on a moon	C	C	C	C
54	How many nonmetals are found on jupiter	C	C	I	C
55	How many gases are found on mars	N	N	I	I
56	How many continents are found on earth	N	C	I	N
57	Is berlin a capital	C	C	C	C
58	Is beijing a city	C	C	C	C
59	Is lyon a moon	C	C	C	C
60	Is india an ocean or a country	C	C	C	N
61	Is canada a mountain	C	C	C	C
62	Is england an atmospheric planet	C	C	I	I
63	Which mountain is found on jupiter	C	C	C	I
64	Which rivers are found on io	C	C	C	C
65	Which nonmetals are found on a planet	C	C	C	C

66	Which gases are found on a moon	C	C	C	C
67	Is an ocean found on mercury	C	C	C	C
68	How many rivers are found on miranda	C	C	C	C
69	How many chemicals are found on phoebe	N	C	C	C
70	How many continents are found on earth	C	C	I	I
71	Is an ocean found on mercury	C	C	C	C
72	How many gases are contained by earth	C	C	I	I
73	How many gases are found on earth	C	C	I	I

Table Appendix D (6): experiment result of Person #1 on extended grammars using semantics set (Cont'd)

Table Appendix D (7): experiment result of Person #1 on extended grammars using syntax set

Person #1 (English Male):

No	Testing Utterances	Sem ext #1 (0/25) (N:21 I: 4)	Sem ext #3 (0/25) (N:21 I:4)	Syn ext #2 (20/25) (N:4 I:1)	Syn ext #4 (21/25) (N:4 I:0)
1	Does a mountain contain a moon	N	N	C	C
2	Does a gas contain a planet	N	N	C	C
3	Does a river contain a continent	N	N	C	C
4	Was phobos discovered by a moon	I	I	C	C
5	Does water contain a river	N	N	I	C
6	Is a crater found in nitrogen	N	N	C	C
7	Does ariel neighbour hall	N	N	C	C
8	Does a moon neighbour a people	N	N	C	C
9	Does a crater contain saturn	N	N	C	C
10	Does a red mountain contain phobos	N	N	C	C
11	Does nitrogen contain janus	N	N	C	C
12	Did berlin discover a moon	N	N	C	C
13	Which mountain is found on bond	N	N	C	C

14	Which moon is found in a gas	I	I	C	C
15	Which mountains are discovered by pacific	I	I	N	C
16	Which river orbits a planet	N	N	N	C
17	How many people neighbour saturn	N	N	C	C
18	Was neptune discovered by dollfus or lyon	N	I	C	N
19	Does triton orbit pluto or frankfurt	N	N	C	C
20	Does gold contain a sea or a mountain	I	N	N	N
21	How many moons are found in atlantic	N	N	C	N
22	How many craters were discovered by nile	N	N	C	N
23	Is gold found in cassini	N	N	C	C
24	Which chemicals are found on bond	N	N	C	C
25	How many chemicals are found on galileo	N	N	N	C

Table Appendix D (7): experiment result of Person #1 on extended grammars using syntax set (Cont'd)

Table Appendix D (8): experiment result of Person #1 on extended word-sequence grammar using semantics set

Person #1 (English Male):

Note: out of 73 testing utterances, there are 14 recognized correctly, 46 recognized Incorrectly, 13 Not recognized.

No	**Testing Utterances**	**Recognized As (*or* "Not recognized")**	**Correctness (C / total words)**
1	Was phobos discovered by a person	Was phobos discovered by person	5 / 6
2	Is titania a mountain	Is titania a mountain	4 / 4
3	Is cassini a moon	Is cassini a moon	4 / 4
4	Is pluto a mountain or a moon	Is pluto a mountain or moon	6 / 7
5	Is pluto an atmospheric crater	Is pluto *and* atmospheric crater	4 / 5
6	Does pluto exist	Does pluto exist	3 / 3
7	Does ariel neighbour pluto	Does ariel neighbour pluto	4 / 4
8	Does a moon neighbour a planet	Does a moon neighbour *atlantic*	4 / 5
9	Does every person worship a planet	Does *africa* worship planet	3 / 6

10	Does saturn contain a crater	Does saturn contain *africa*	3 / 5
11	Does phobos contain a red mountain	Does phobos contain a red mountain	6 / 6
12	Does janus contain nitrogen	Does janus contain nitrogen	4 / 4
13	Did bernard discover a mountain	Did *banjul* discover a mountain	4 / 5
14	Who discovered a crater	Who discovered a crater	4 / 4
15	Which mountain is found on uranus	Which mountain is *yaounde are* uranus	4 / 6
16	Which gas is found on a moon	Which gas is *yaounde ghana* moon	4 / 7
17	What is contained by venus	What is contained *five* venus	4 / 5
18	What is contained by phobos	What is contained *five* phobos	4 / 5
19	Which mountain is found on janus	Which mountain is *yaounde* janus	4 / 6
20	Which sea exists	Which sea exists	3 / 3
21	Which mountains were discovered by hall	Which mountains were discovered by hall	6 / 6
22	Which moon orbits a planet	Which moon *dollfus atlantic*	2 / 5
23	How many moons neighbour saturn	How many moons neighbour saturn	5 / 5
24	Was neptune discovered by dollfus or cassini	Was neptune discovered *five* dollfus or cassini	6 / 7
25	Does triton orbit pluto or saturn	Does triton orbit pluto *oxygen*	4 / 6
26	Does neptune contain hydrogen or nitrogen	Does neptune contain hydrogen *hall* nitrogen	5 / 6
27	Does phobos contain a sea or a mountain	Does phobos contain a sea or mountain	6 / 7
28	Does phoebe contain hydrogen or oxygen	Does phoebe contain hydrogen *four* oxygen	5 / 6
29	Does oberon contain oxygen	Does oberon contain oxygen	4 / 4
30	Does a moon contain hydrogen	Does a moon contain hydrogen	5 / 5
31	Does a moon neighbour a planet	Does a moon neighbour *atlantic*	4 / 6
32	How many gases are found on mars	How many gases *iceland amman*	3 / 7
33	How many craters are found on a moon	Not recognized	
34	How many oceans were discovered by hall	How *nicholson swede* discovered by hall	4 / 7
35	How many mountains are found on earth	How many mountains are *saint georges*	4 / 7
36	Is gold found on earth	Not recognized	
37	Is silver found on janus	Is suva yaounde bond janus	2 / 5
38	Is a chemical found on triton	Not recognized	
39	Is dioxide found on phoebe	Is n*ile kazakhstan dione* phoebe	2 / 6

40	Is sulphur found on luna	Is *dollfus yaounde or* luna	2 / 5
41	Is oxygen found on mars	Is oxygen *sun amman*	2 / 5
42	Is a metal found on a planet	Not recognized	
43	Is a nonmetal found on pluto	Not recognized	
44	Is a river found on neptune	Is *aruba yaounde bond* neptune	2 / 6
45	Is a lake found on venus	Is *atlantic yaounde bond* venus	2 / 6
46	Which gas is found on titan	Which gas is *yaounde bond* titan	3 / 6
47	Which chemicals are found on rhea	Which *nicholson Iceland oman* rhea	2 / 6
48	Which nonmetals are found on jupiter	Which *monaco sanaa finland bond* jupiter	2 / 6
49	Which metals are found on a moon	Not recognized	
50	Which river is found on hyperion	Which *aruba* is *mountain* hyperion	3 / 6
51	Which mountains are found on rhea	Which mountains *yaounde austria*	2 / 6
52	How many chemicals are found on pluto	Not recognized	
53	How many metals are found on a moon	How many *brussels iceland oman* a moon	4 / 8
54	How many nonmetals are found on jupiter	How *manama togo shanghai bond* jupiter	2 / 7
55	How many gases are found on mars	How many *iceland* are mars	4 / 8
56	How many continents are found on charon	How many *contains how yaounde russia*	2 / 7
57	Is berlin a capital	Is berlin *atlantic*	2 / 4
58	Is beijing a city	Is beijing a *sea*	3 / 4
59	Is lyon a moon	Is lyon a moon	4 / 4
60	Is india an ocean or a country	Is india *are nicholson harare conakry*	2 / 7
61	Is canada a mountain	Is canada a mountain	4 / 4
62	Is england an atmospheric planet	Is *finland and* atmospheric planets	3 / 5
63	Which mountain is found on jupiter	Which mountain is *yaounde* jupiter	4 / 6
64	Which rivers are found on io	Not recognized	
65	Which nonmetals are found on a planet	Not recognized	
66	Which gases are found on a moon	Not recognized	
67	Is an ocean found on mercury	Not recognized	
68	How many rivers are found on miranda	How *maseru is yaounde are* miranda	2 / 7
69	How many chemicals are found on phoebe	How many *guinea phobos mountains* phoebe	3 / 7

70	How many continents are found on earth	Not recognized	
71	Is an ocean found on mercury	Not recognized	
72	How many gases are contained by earth	How many gases are contain by *pairs*	6 / 7
73	How many gases are found on earth	How many gases *yaounde honduras*	3 / 7

Table Appendix D (8): experiment result of Person #1 on extended word-sequence grammar using semantics set (Cont'd)

Table Appendix D (9): experiment result of Person #2 on extended grammars using semantics set

Person #2 (non-English Female):

No	Testing Utterances	Sem ext #1 43 /73 N:25I:5	Sem ext #3 46/73 N:23I:4	Sem ext #5 45/73 N:22I:6	Sem ext #7 44 /73 N:26I:3	Syn ext #2 29 /73 N:35I:9	Syn ext #4 32/73 N27I:14	Syn ext #6 38/73 N25I:10	Syn ext #8 37/73 N21I:15	Wd Seq ext #9(4/73 N20I:49	Wd Seq ext #10(3/73 N:17I:53
1	Was phobos discovered by a person	N	N	N	N	N	N	I	N	I	N
2	Is titania a mountain	N	C	N	C	N	C	C	C	I	I
3	Is cassini a moon	C	C	C	C	N	C	C	C	I	I
4	Is pluto a mountain or a moon	N	I	N	N	N	N	N	N	I	I
5	Is pluto an atmospheric crater	C	C	C	C	C	C	C	I	I	C
6	Does pluto exist	C	C	C	C	C	C	C	C	C	I
7	Does ariel neighbour pluto	N	C	I	C	C	I	C	N	I	I
8	Does a moon neighbour a planet	C	C	C	C	C	N	N	C	N	N
9	Does every person worship a planet	C	C	N	C	N	N	C	C	N	I
10	Does saturn contain a crater	I	I	I	C	I	I	I	I	I	I
11	Does phobos contain a red mountain	C	C	C	C	C	I	C	C	I	I
12	Does janus contain nitrogen	C	C	C	C	C	c	c	c	I	I
13	Did bernard discover a mountain	C	N	C	N	N	N	N	C	N	N
14	Who discovered a crater	C	C	C	C	C	C	C	C	C	C
15	Which mountain is found on uranus	C	C	N	C	C	N	C	N	I	I
16	Which gas is found on a moon	C	C	C	C	I	C	C	C	I	N

17	What is contained by venus	N	N	C	N	N	N	N	C	I	I
18	What is contained by phobos	C	N	C	N	N	C	N	N	I	I
19	Which mountain is found on janus	N	C	C	C	C	C	C	C	I	I
20	Which sea exists	C	C	C	C	C	C	C	C	C	C
21	Which mountains were discovered by hall	N	C	N	C	N	N	N	N	I	I
22	Which moon orbits a planet	C	C	C	N	C	C	C	N	I	I
23	How many moons neighbour saturn	C	C	C	C	N	I	I	N	N	I
24	Was neptune discovered by dollfus or cassini	N	N	N	N	N	N	N	N	I	I
25	Does triton orbit pluto or saturn	N	N	N	N	N	N	N	N	N	N
26	Does neptune contain hydrogen or nitrogen	N	N	C	N	N	N	N	N	I	I
27	Does phobos contain a sea or a mountain	N	N	N	N	N	N	N	C	I	I
28	Does phoebe contain hydrogen or oxygen	N	N	N	N	N	C	N	N	I	I
29	Does oberon contain oxygen	C	C	C	C	C	C	C	C	I	I
30	Does a moon contain hydrogen	C	C	C	C	C	c	c	c	I	I
31	Does a moon neighbour a planet	C	C	C	N	C	I	C	C	I	N
32	How many gases are found on mars	C	C	C	C	C	N	C	I	N	I
33	How many craters are found on a moon	C	C	C	C	C	C	C	C	I	I
34	How many oceans were discovered by hall	N	C	N	N	N	N	N	C	N	N
35	How many mountains are found on earth	N	C	C	N	N	C	I	I	I	I
36	Is gold found on earth	I	N	C	I	I	I	I	I	I	I
37	Is silver found on janus	C	C	C	C	I	N	C	C	I	I
38	Is a chemical found on triton	N	C	C	C	C	C	C	C	I	I

39	Is dioxide found on phoebe	C	C	C	C	N	C	N	C	I	I
40	Is sulphur found on luna	N	C	C	C	C	I	C	C	N	N
41	Is oxygen found on mars	C	C	C	I	C	I	C	C	I	I
42	Is a metal found on a planet	C	C	C	C	C	N	N	N	N	I
43	Is a nonmetal found on pluto	C	C	C	C	C	N	C	C	N	N
44	Is a river found on neptune	N	N	N	N	N	N	N	I	N	N
45	Is a lake found on venus	C	C	C	C	C	N	N	I	N	I
46	Which gas is found on titan	C	C	C	C	C	C	C	C	I	I
47	Which chemicals are found on rhea	C	C	N	C	N	C	I	I	I	I
48	Which nonmetals are found on jupiter	N	N	N	C	N	c	c	i	N	I
49	Which metals are found on a moon	C	C	N	C	N	C	C	C	I	I
50	Which river is found on hyperion	N	N	N	N	N	N	N	N	N	I
51	Which mountains are found on rhea	C	C	C	C	N	I	I	I	I	I
52	How many chemicals are found on Pluto	C	N	C	C	C	C	C	C	I	I
53	How many metals are found on a moon	C	C	N	C	C	C	C	C	I	I
54	How many nonmetals are found on jupiter	C	N	N	C	N	C	C	N	I	N
55	How many gases are found on mars	I	I	I	N	C	N	N	I	I	I
56	How many continents are found on charon	C	C	C	C	I	C	C	C	i	I
57	Is berlin a capital	I	C	I	C	I	I	I	I	I	I
58	Is beijing a city	C	C	C	C	N	I	C	C	I	I
59	Is lyon a moon	C	C	I	C	I	C	C	C	C	N
60	Is india an ocean or a country	N	N	N	N	N	N	N	N	N	N
61	Is canada a mountain	C	C	C	N	C	C	C	C	I	I
62	Is england an atmospheric planet	N	C	N	N	N	N	N	N	I	N
63	Which mountain is found on jupiter	C	C	C	N	N	C	N	N	N	I

64	Which rivers are found on io	N	N	N	N	N	N	N	N	N	N
65	Which nonmetals are found on a planet	N	N	C	N	N	N	N	N	N	I
66	Which gases are found on a moon	N	N	C	N	N	C	C	C	I	I
67	Is an ocean found on mercury	C	C	C	C	C	I	C	C	N	N
68	How many rivers are found on miranda	C	N	C	N	N	N	N	N	I	I
69	How many chemicals are found on phoebe	C	C	C	C	C	C	C	C	I	I
70	How many continents are found on earth	C	I	C	C	I	I	I	I	I	I
71	Is an ocean found on mercury	C	N	C	C	I	C	C	I	N	N
72	How many gases are contained by earth	N	N	N	N	N	N	N	N	I	I
73	How many gases are found on earth	I	N	I	I	N	I	I	i	i	I

Table Appendix D (9): experiment result of Person #2 on extended grammars using semantics set (Cont'd)

Table Appendix D (10): experiment result of Person #2 on extended grammars using syntax set

Person #2 (non-English Female):

No	Testing Utterances	**Sem ext #1 (0/25)** N:20, I: 5	**Sem ext #3 (0 /25)** N:22, I: 3	**Syn ext #2 (12/25)** N: 10 I: 3	**Syn ext #4 (12/25)** N: 12, I: 1	**Wd Seq ext #5 (1/25)** N: 9, I: 15	**Wd Seq ext #6 (0/25)** N:11, I: 14
1	Does a mountain contain a moon	N	N	N	C	I	N
2	Does a gas contain a planet	N	N	C	C	I	I
3	Does a river contain a continent	N	N	N	N	N	N
4	Was phobos discovered by a moon	N	N	N	N	I	N
5	Does water contain a river	N	N	C	N	I	I
6	Is a crater found in nitrogen	I	N	C	C	N	N
7	Does ariel neighbour hall	N	N	I	C	I	N
8	Does a moon neighbour a person	N	N	C	C	C	I
9	Does a crater contain saturn	N	N	I	I	N	N

10	Does a red mountain contain phobos	N	N	C	C	I	I
11	Does nitrogen contain janus	N	I	C	C	I	I
12	Did berlin discover a moon	I	I	C	C	N	N
13	Which mountain is found on bond	I	I	C	N	I	I
14	Which moon is found in a gas	I	N	N	N	I	I
15	Which mountains were discovered by pacific	N	N	N	N	I	N
16	Which river orbits a planet	N	N	N	N	I	N
17	How many people neighbour saturn	N	N	I	C	N	I
18	Was neptune discovered by dollfus or lyon	N	N	N	N	N	I
19	Does triton orbit pluto or frankfurt	N	N	N	N	N	I
20	Does gold contain a sea or a mountain	N	N	C	N	I	N
21	How many moons are found in atlantic	N	N	C	C	I	I
22	How many craters were discovered by nile	N	N	N	N	I	I
23	Is gold found in cassini	N	N	C	C	I	I
24	Which chemicals are found on bond	I	N	C	C	N	I
25	How many chemicals are found on galileo	N	N	N	N	n	N

Table Appendix D (10): experiment result of Person #2 on extended grammars using syntax set (Cont'd)

Table Appendix D (11): experiment result of Person #2 on extended grammars using word-sequence set

Person #2 (non-English Female):

No	**Testing Utterances**	**Sem ext #1 (0/24)** (N:23 I: 1)	**Sem ext #4 (0/24)** N:23, I:1	**Syn ext #2 (0/24)** (N: 18, I:6)	**Syn ext #5 (0/24)** N:17, I:7	**Wd Seq ext #3 (1/24)** N:5, I: 18	**Wd Seq ext #6 (2/24)** N:8 I: 14
1	Is a mountain contain a moon	N	N	N	N	I	N
2	Does a gas a planet	N	N	N	N	I	N
3	Is a river found a continent	N	N	N	I	N	N
4	Phobos discovered by a moon	N	N	N	N	I	I

5	Does water exist a river	N	N	N	N	N	I
6	Is a crater contain nitrogen	N	N	I	I	I	I
7	Is ariel neighbour a planet	N	N	N	N	N	I
8	Is a moon discover a people	N	N	I	I	I	N
9	Which crater contain on saturn	N	N	N	I	N	N
10	Is a red phobos contain a mountain	N	N	N	N	I	I
11	Is janus contain nitrogen	I	N	I	N	I	C
12	Is Jupiter discovered bernard	N	N	I	N	I	I
13	Which mountain is found dione and phoebe	N	N	N	N	I	I
14	Which gas found moon	N	N	N	N	I	I
15	Which mountains discovered by melotte	N	N	N	N	I	N
16	Which moon orbits on a planet	N	N	N	N	I	N
17	How many moons neighbour on saturn	N	N	I	I	I	N
18	Was neptune discovered dollfus and kowal	N	N	N	N	C	C
19	Is triton orbit Pluto or venus	N	N	N	N	N	N
20	Is gold contained a moon	N	N	I	I	I	I
21	How many mountains found on oberon	N	N	N	N	I	I
22	How many craters are found earth	N	N	N	N	I	I
23	Is gold found cassini	N	N	N	I	I	I
24	Which chemicals are found bond	N	I	N	N	I	I

Table Appendix D (11): experiment result of Person #2 on extended grammars using word-sequence set (Cont'd)

Vita Auctoris

The author was born in China in 1972. She completed her B.Sc. Degree in Computer Science at Southeast University, China, in June 1993. She had been working as a software developer, system administrator, and technical support in Northeast Air Traffic Administration, Shenyang, China, before she immigrated to Canada in 2001.

She is currently a candidate for the M.Sc. degree in computer science, supervised by Dr. Richard A. Frost, at the University of Windsor, Ontario, Canada. Her primary research interest is speech recognition in natural-language interfaces.