AN INVESTIGATION OF LINGUISTIC INFORMATION FOR SPEECH RECOGNITION ERROR DETECTION

by
Yongmei Shi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

APPROVAL SHEET

Title of Dissertation: An Investigation of Linguistic Information for Speech Recognition Error Detection

Name of Candidate: Yongmei Shi
Doctor of Philosophy, 2008

Dissertation and Abstract Approved: _____

Dr. R. Scott Cost
Adjunct Associate Professor
Department of Computer Science and Electrical Engineering
Senior Computer Scientist
Milton Eisenhower Research Center
Johns Hopkins University Applied Physics Laboratory

Dr. Lina Zhou
Associate Professor
Department of Information Systems

Date Approved: _____

ABSTRACT

Title of Dissertation: AN INVESTIGATION OF LINGUISTIC INFORMATION FOR SPEECH RECOGNITION ERROR DETECTION

Yongmei Shi, Doctor of Philosophy, 2008

Dissertation directed by: Dr. R. Scott Cost
Adjunct Associate Professor
Department of Computer Science and Electrical Engineering
Senior Computer Scientist
Milton Eisenhower Research Center
Johns Hopkins University Applied Physics Laboratory

Dr. Lina Zhou
Associate Professor
Department of Information Systems

After several decades of effort, significant progress has been made in the area of speech recognition technologies, and various speech-based applications have been developed. However, current speech recognition systems still generate erroneous output, which hinders the wide adoption of speech applications. Given that the goal of error-free output can not be realized in near future, mechanisms for automatically detecting and even correcting speech recognition errors may prove useful for amending imperfect speech recognition systems. This dissertation research focuses on the automatic detection of speech recognition errors for monologue applications, and in particular, dictation applications.

Due to computational complexity and efficiency concerns, limited linguistic information is embedded in speech recognition systems. Furthermore, when identifying speech recognition errors, humans always apply linguistic knowledge to complete the task. This dissertation therefore investigates the effect of linguistic information on automatic error detection by applying two levels of linguistic analysis, specifically syntactic analysis and semantic analysis, to the post processing of speech recognition output. Experiments are conducted on two dictation corpora which differ in both topic and style (daily office communication by students

and Wall Street Journal news by journalists).

To catch grammatical abnormalities possibly caused by speech recognition errors, two sets of syntactic features, linkage information and word associations based on syntactic dependency, are extracted for each word from the output of two lexicalized robust syntactic parsers respectively. Confidence measures, which combine features using Support Vector Machines, are used to detect speech recognition errors. A confidence measure that combines syntactic features with non-linguistic features yields consistent performance improvement in one or more aspects over those obtained by using non-linguistic features alone.

Semantic abnormalities possibly caused by speech recognition errors are caught by the analysis of semantic relatedness of a word to its context. Two different methods are used to integrate semantic analysis with syntactic analysis. One approach addresses the problem by extracting features for each word from its relations to other words. To this end, various WordNet-based measures and different context lengths are examined. The addition of semantic features in confidence measures can further yield small but consistent improvement in error detection performance. The other approach applies lexical cohesion analysis by taking both reiteration and collocation relationships into consideration and by augmenting words with probability predicted from syntactic analysis. Two WordNet-based measures and one measure based on Latent Semantic Analysis are used to instantiate lexical cohesion relationships. Additionally, various word probability thresholds and cosine similarity thresholds are examined. The incorporation of lexical cohesion analysis is superior to the use of syntactic analysis alone.

In summary, the use of linguistic information as described, including syntactic and semantic information, can provide positive impact on automatic detection of speech recognition errors.

AN INVESTIGATION OF LINGUISTIC INFORMATION FOR SPEECH RECOGNITION ERROR DETECTION

by
Yongmei Shi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

To my parents and my husband

Acknowledgements

I would like to extend my thanks to two departments: Department of Computer Science and Electrical Engineering where I spent the first half of my journey to PhD, and Information Systems Department where I finished my journey. Both departments jointly offered a wonderful environment for me to enrich my knowledge and complete my PhD study.

First of all, I wish to express my thanks to my advisors, Dr. R. Scott Cost and Dr. Lina Zhou, for their guidance and support throughout my research. Dr. Cost introduced me into the general field of human language technology and taught me how to become a researcher. Dr. Zhou provided me the opportunity to this interesting and challenging topic. She also opened the door of human computer interaction to me. I have learned a lot from her feedbacks on my research work as well as her dedication to research. My thanks also go to my other committee members: Dr. Tim Finin, Dr. Charles Nicholas, and Dr. Andrew Sears. They provided invaluable suggestions to help me to refine my research work and gave me many helpful comments on my dissertation.

I would also like to thank group members in Interactive Systems Research Center. They kindly provided the speech corpus to me and offered me the help during my processing of the corpus. They also gave me advice when I conducted human studies. Special thanks go to Dr. Jinjuan Feng at Towson University, a former member of ISRC. I had great experience working with her, and I appreciate the speech corpus newly collected by her group at Towson University. Many thanks also go to my colleagues, especially Srikanth Kallurkar, in former CADIP lab. Collaboration with them in CARROT II project gave me a chance to understanding information retrieval systems and agent technology. My friends in both CSEE department and IS department are always with me on my each step towards the finish line, and they made my experience at UMBC colorful. Thank you all.

Studying aboard was a new experience to me, and besides the excitement I also felt homesick. I'm grateful to Nancy Cutair, June Auer, and Michael Auer. They are my American family members and they let me feel the warmth of family.

I would take this chance to thank the most important ones in my life, my beloved family. The love and support from my parents and my brother always be with me during my education. I could not finish this program without my husband, whose believe in me gave me confidence and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivations

Automatic speech recognition (ASR) is "the process by which a computer maps an acoustic speech signal into text [1]" and has been an important and active research area for over 50 years [86]. The importance of ASR lies in the role that speech plays in our everyday lives. Speech is the most natural communication modality used by humans [60], however, it is not the typical input modality afforded by computers. If speech were an alternative effective input modality to the keyboard and mouse, the interaction between humans and computers would become more natural. Additionally, speech is a well-known factor in enabling mobility, which is difficult to accomplish using traditional input devices [60].

In recent years, with advancement in techniques for signal processing and model building and the empowerment of computing devices, significant progress has been made in speech recognition research, and various speech based applications have been developed. The state of speech recognition technologies has achieved satisfactory performance under restrictive conditions (i.e. limited vocabulary, reading speech, and noise-free environment). As a result, speech recognition has been widely used in applications such as command-control, call routing, and telephone directories. When it comes to open environments, especially in noisy environments and for natural free-style speech [24], however, the performance of speech recognition technologies remains unsatisfactory, which results in their limited usage.

Minimizing speech recognition errors and making speech recognition effective under all conditions is the ultimate goal of speech recognition research. After several decades' worth of effort, today's reality is still far from the desired outcome. Given that existing speech recognition technologies remain error prone, a

good mechanism for detecting and even correcting speech recognition errors would be instrumental to wide adoption of speech recognition systems.

1.2 Scope

Based on the target of speech (human or computer) and speech interaction style (monologue or dialogue), speech recognition applications can be generally classified into four categories [30]:

- human-human dialogue: speech recognition systems generate transcription of human dialogue, such as conversational telephone speech and meetings. Transcribing human dialogue is a challenging task due to the spontaneity of speech.
- human-computer dialogue: spoken dialogue systems allow humans to talk with computers and to complete certain tasks using their voices. A spoken dialogue system, which includes a speech understanding system and a speech synthesis system, is more than a speech recognition system. Restricted by the limited progress in automatic language understanding, spoken dialogue systems are confined to specific domains such as travel planning services.
- human-human monologue: in this category, human speech is used for conveying information, such as broadcast news, TV shows, and lectures. Speech recognition is the secondary task in which recorded audio data can be transcribed and indexed for future information management.
- human-computer monologue: a human speaks to a computer that transcribes speech into text. Dictation, which is generally used as a document generation tool, is the most prevalent application in this category.

This dissertation focuses on monologue applications in general and dictation in particular. Based on the degree of preparation for the dictated text, dictation can be further divided into two sub-categories: transcription and composition [51]. Transcription is the task of reproducing written documents; speakers read written documents to speech recognition systems which transcribe the speech and convert documents into a machine-readable format. Compared to transcription, composition offers a more realistic speaking scenario. People directly use dictation software to compose documents such as emails and reports by speaking to computers with little or no preparation. People spontaneously dictate what they want to write down, which makes com-

position more error prone. Since the final goal of dictation is to generate a written document, the speaking style, as reflected in dictated sentences, is close to the writing style used in written documents.

To produce an error-free speech transcript in monologue applications such as dictation, error correction is important and indispensable. Error correction involves three steps: detecting an error, navigating to the error, and correcting the error. Manual error correction is neither easy nor efficient. Sears et al. [100] found that for a hand-free composition task users spent one third of their time detecting and navigating to errors and another third of their time correcting errors. These efforts required by error correction can compromise the promise of fast and natural speech input. This dissertation focuses on error detection, the first step in and the premise of speech recognition error correction.

1.3 Problem Identification

A speech recognition system aims to find out the most likely word sequence W given the acoustic signal A. With this criteria, the posterior probability $P(W|A)$ is a theoretically accurate measure to judge the correctness of a word. However, in an actual speech recognition system, the maximum likelihood estimation is always used instead by eliminating the constant value $P(A)$, which makes the value assigned to every word a relative rather than an absolute measure. Therefore, values produced by a speech recognition system can be used for selecting among alternatives but may not be directly used for rejecting words as errors [118].

One direction of research concerning the improvement of error detection is to estimate the posterior probability. Another direction is to combine various features that are useful to judge the correctness of words. The majority of research on error detection focuses on feature combination. Various features have been proposed, and most are from the components of speech recognition systems. Features from a speech recognition system itself are not adequate because information conveyed by these features has already been used in generating output by the speech recognition system. Therefore, additional sources of complementary information should be sought.

Linguistic analysis is one such source. In current speech recognition systems, linguistic information is represented by language models. The commonly used language models are n-gram models, which can only detect short, immediate collocation patterns (i.e. n words). Therefore, there always exist errors in the recognition output that are out of context or that cause the sentence to be ungrammatical. Figure 1.1 gives several examples; sentences under *REF* are transcriptions manually generated by humans and convey what

speakers actually said, and sentences under *REC* are automatically generated by a speech recognition system. Words in bold are recognition errors. In sentence *a*, *gear* is mistakenly recognized as *your*, which causes sentence *a* to be ungrammatical. In sentence *b*, *hotel* is erroneously recognized as *propel* and *around* is segmented as *a rare and*, which cause sentence *b* to also be ungrammatical. In sentence group *c*, *folacin* in sentence *c.2* is the erroneous output of *follow suit* and is not semantically related to any other words in those three sentences.

REF:	a.	we would need a lot of gear .
REC:	a.	we would need a lot of **your** .
REF:	b.	Once I get to London I would like to stay in a five-star hotel in dining at the most eloquent restaurants around .
REC:	b.	**and on Friday** to London I would like to stay in five-star **propel** in **dying** at the most eloquent **rest and takes a rare and** .
REF:	c.1	Northwest Airlines launched a new airline fare war today by slashing some of its prices by up to forty five percent
	c.2	American Delta and United are expected to follow suit
	c.3	The ticket discounting comes as the busy summer season ends and the slow winter travel season begins
REC:	c.1	Northwest Airlines launched a new airline fare war **to drive** by slashing some of its prices by up to forty five percent
	c.2	American **doubt that** United are expected **folacin**
	c.3	the ticket discounting comes **a** busy summer season ends and the slow winter travel season begins

Figure 1.1: Examples of speech recognition output and corresponding manual transcript

There are two possible solutions that address these kinds of errors by compensating for the lack of high-level linguistic knowledge in a speech recognition system. One is to design complex language models that could incorporate syntactic or semantic information to lower the error rate; the other is to apply the linguistic analysis on speech recognition output to detect and correct errors. This dissertation applies the latter solution and focuses on detecting errors.

1.4 Our Approach

When processing time-sequential acoustic signal, speech recognition systems could only utilize limited local context and previous output. A post-processing approach, which focuses on speech recognition output, is then selected in this research. Two factors favor the post-processing approach in this research:

- During post-processing, the entire speech recognition output is available, making it possible to apply complex linguistic analysis.

- Selecting the recognition output as the target for post-processing reduces the reliance on detailed information about the speech recognition system itself. To a large extend, it allows the proposed methods to treat the speech recognition system as a "blackbox", which is desirable.

Several levels of linguistic analysis, including syntactic analysis and semantic analysis, are investigated in this dissertation to improve the error detection activities. Figure 1.2 shows the framework of proposed error detection methods. As shown in the framework, feature combination-based confidence measures and lexical cohesion analysis are used to integrate different types of knowledge.

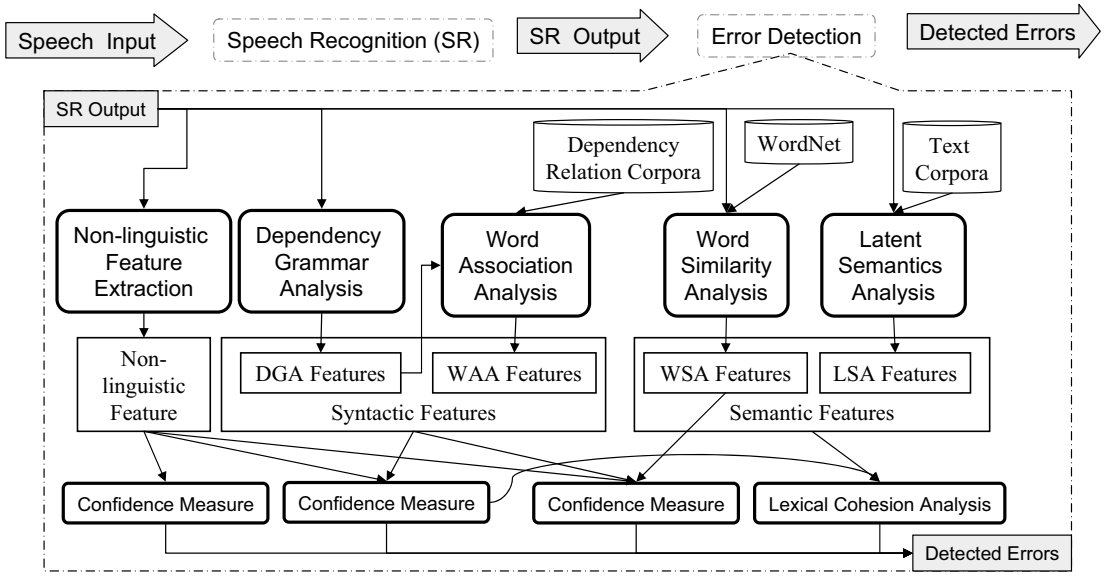


Figure 1.2: Framework of proposed error detection methods

Speech recognition errors may render the corresponding output ungrammatical. For example, in sentence *a* in Figure 1.1, the error *your* does not belong to the same grammatical category as the actual word *gear*. In this case, *your* in the output of a lexicalized parser may not be connected to any other words in the sentence. In sentence *b*, the error *rest* is a fragmentation of the word *restaurants*. Even if *rest* falls into the same grammatical category as *restaurants*, a weaker association between *eloquent* and *rest* results from the modification. Previous research [119, 96] has demonstrated the usefulness of parser based features for error detection in spoken dialogue systems. However, those features were extracted from domain-specific parsers that are difficult to apply to the general domain. This dissertation will exploit knowledge from syntactic

analysis.

Semantic discrepancy has been found to be the most used knowledge cue by participants when detecting errors [121]. Moreover, it was found that when the context of an utterance was given, participants achieved a significant improvement on their error detection performance [104]. Previous research [21, 44, 97] represented semantic knowledge as similarities among words in the context. Following this idea, this dissertation will investigate what role semantic knowledge plays in finding speech recognition errors.

One way of utilizing semantic knowledge is to analyze the lexical cohesion of the speech recognition output. Words in text are not randomly connected but stick together through certain relations. A speech recognition error may disrupt lexical cohesion. Lexical cohesion analysis has been applied to correct malapropisms intentionally created in text [40, 39]. This dissertation will examine whether lexical cohesion analysis can help find errors in natural erroneous speech recognition output.

1.5 Dissertation Outline

The organization of the dissertation will follow the framework in Figure 1.2 according to the level of linguistic analysis. Chapter 2 introduces basic concepts of speech recognition errors and reasons why errors occur. Methods used to prevent, detect, and correct speech recognition errors are also reviewed. Chapter 3 describes the data corpora that will be used in this dissertation to evaluate the proposed methods. Chapter 4 presents a baseline confidence measure that uses non-linguistic features extracted from the speech recognition output and serves as the basis for the following three chapters. Chapter 5 focuses on analyzing features extracted from the syntactic analysis of speech recognition output. Besides amending features to the baseline confidence measure, a deep analysis of error patterns is also presented. Chapter 6 focuses on incorporating features from word semantic relatedness analysis based on WordNet. Chapter 7 introduces the concept of lexical cohesion and its analysis method to find errors in speech recognition output. Chapter 8 summarizes the dissertation and outlines future directions.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

2.1 Speech Recognition Errors

Speech recognition (SR) systems usually take speech signals as input and generate text transcripts as output. Figure 2.1 shows a communication channel model of speech recognition systems, in which speakers speak out what they want to say (W) and speech signals are then processed by speech recognizer to generate the speech recognition output ($\hat{W}$). Due to imperfect models for each component of an SR system, errors, which can be conceptualized as the differences between $\hat{W}$ and W, always occur.

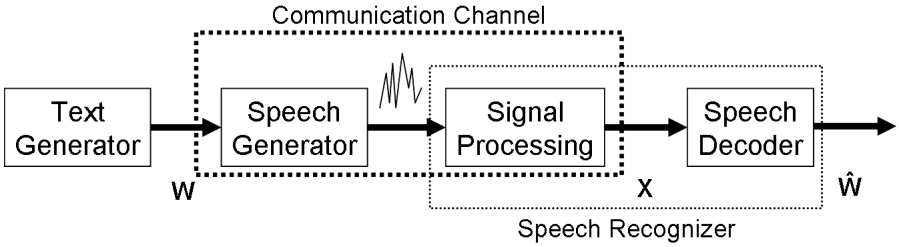


Figure 2.1: Speech recognition system (source [42] page 5)

2.1.1 Causes of Errors

Errors can be caused by one or more components of an SR system. According to Chase [14], Suhm [108] and Gibbon et al. [33], recognition errors are mainly caused by the following factors:

- Acoustic Model: The acoustic model captures the acoustic characteristics of the speech signal. The construction of an acoustic model usually involves two processes: signal processing, in which features

are extracted from speech waveform, and parameter estimation, in which model parameters are iteratively updated using the training corpus and the pronunciation dictionary. Deficiency in the training process of each component may cause the model to be inaccurate. For example, inaccuracy in the pronunciation dictionary and reference transcript and imperfect signal segmentation will make the acoustic model inaccurate.

- Language Model: Language models in SR systems are used to capture typical word usage in different applications and resolve the ambiguities in acoustic evidence. So far, there is no perfect method to build language models. The discrepancy of styles and topics between the training corpus and the test corpus will adversely affect SR performance. Sometimes, language models may overwhelm correct acoustic evidence.

- Decoder: To process a large vocabulary and to be used in real time, SR systems usually apply some heuristics to balance accuracy and efficiency. Instead of finding the global optimal output from an entire search space, a suboptimal output is found within a limited search space. Hence, during the search, beams with certain widths are set, and some impossible hypotheses are eliminated. However, correct hypotheses may also be pruned out at early stages.

- Out-Of-Vocabulary (OOV) Words: Current SR systems can only recognize words from a pre-defined vocabulary. If a new word is spoken, SR systems will try to find one or more words that best match acoustic signals as the output. Errors will occur during this selection process.

Consequently, the performance of SR systems can be improved by refining each of the above components. For example, OOV words can be avoided by expanding the vocabulary statically or dynamically, and the number of search errors can be decreased by loosing pruning thresholds to trade accuracy with time.

2.1.2 Measure of Errors

A commonly used evaluation metric for the performance of speech recognition systems is Word Error Rate (WER). It is defined based on an alignment of the recognition output with the reference transcript, as illustrated in Figure 2.2. The line starting with *REF* represents the reference transcript, and the line starting with *REC* represents the recognition output.

Based on the alignment, errors in SR output can be classified into three categories: substitution, insertion, and deletion.

REF:	with that in mind HERE'S A GLOSSARY of ******** terms used to describe *** investment risks
REC:	with that in mind FEARS ** ********* of AUSTRIA terms used to describe THE investment risks
	S D D I I

Figure 2.2: A sample speech recognition output containing three types of errors

- Substitution (S): At the same position in the alignment of an output and its reference, a word in the output is different from the word in the reference, such as "FEARS" in *REC* line in Figure 2.2.
- Insertion (I): At the same position in the alignment of an output and its reference, there is a word in the output that does not appear in the reference, such as "THE" in *REC* line in Figure 2.2.
- Deletion (D): At the same position in the alignment of an output and its reference, there is a word in the reference that does not appear in the output, such as "A" in *REF* line in Figure 2.2.

According to this classification of errors, WER can be computed with Equation 2.1. Correspondingly, substitution/deletion/insertion error rate can be computed with Equation 2.2. Word error rate is the sum of substitution error rate, deletion error rate, and insertion error rate.

$$WER = \frac{S + I + D}{N} \tag{2.1}$$

$$\begin{aligned} Substitution\ Error\ Rate &= \frac{S}{N} \\ Deletion\ Error\ Rate &= \frac{D}{N} \\ Insertion\ Error\ Rate &= \frac{I}{N} \end{aligned} \tag{2.2}$$

where N is the total number of words in the reference transcript. In the example shown in Figure 2.2, $N = 14$, and $S + I + D = 1 + 2 + 2 = 5$. Thus, the WER of the sentence is 35.71%. Because of insertion errors, WER can be larger than 100%. The minimum value of WER is 0, when there is no error.

2.2 Error Prevention by Improving Language Models

The prevention of SR errors is one major goal of speech recognition research. To improve the performance of SR systems, advanced techniques are proposed for each SR system component (front-end feature extraction,

acoustic modeling, language modeling, and decoding). For a comprehensive review of the common techniques used for these components, please refer to [42]. Given that this dissertation primarily focuses on the impact of linguistic knowledge on speech recognition error detection, in this section, we only briefly describe advanced techniques for building language models, with an emphasis on language models that incorporate linguistic knowledge.

2.2.1 Basic N-gram Language Models

A commonly used language model in SR systems is the n-gram language model. Given a word sequence S consisting of N words $(w_1, w_2, \ldots, w_N)$, the probability of S, $P(S)$, can be approximated with Equation 2.3:

$$P(S) = \prod_{i=1}^{N} P(w_i | w_{i-n+1}, \ldots, w_{i-1}) \tag{2.3}$$

From Equation 2.3, it can be seen that in n-gram language models the probability of predicting a word depends only on the previous $n-1$ words. Therefore, n-gram language models can only characterize short, immediate collocation relations. To capture longer context, knowledge from syntactic or semantic levels has been incorporated into language models.

2.2.2 Language Models Incorporating Syntactic Knowledge

Syntactic structured language models are derived from syntactic grammars for the purpose of expanding the immediate local context captured in n-gram language models to the sentence level. Various kinds of grammar have been used to extract syntactic knowledge.

Moore et al. [77] integrated statistical knowledge and linguistic knowledge extracted from a unification grammar into a multi-level n-gram language model, which was used to rescore initial speech recognition output to improve the recognition performance. Utterances were parsed into sequences of syntactic fragments. The top level of the language model was a tri-gram model of fragment types, and the lower level was a quadra-gram model of words and word classes for each fragment type. In the December 1994 ATIS benchmark test, the proposed language model led to a 15% relative improvement in WER after rescoring output generated by baseline DECIPHER which used a tri-gram word class model.

Probabilistic link grammars can include n-gram models as a subclass and incorporate long range dependence [53]. Delle Pietra et al. [84] extended the traditional trigram by conditioning word probability not only on the previous two words but also on any possible preceding pair of adjacent words. Chelba et al. [16] developed a maximum entropy language model which incorporated dependence relations similar to link relations. The constraints on histories were extracted from a finite context (only 0-2 preceding words) and a link stack (only open links). This model showed a small but significant improvement on WER over bi-gram language models on the Switchboard corpus.

Chelba et al. [17, 15] developed a structured language model based on a shift-reduce parser. Three modules were included in their model: word-predictor (probabilistically predicting a word given previous two headwords), tagger (probabilistically predicting the part-of-speech of a word given the words and parts-of-speech of headwords), and parser (performed a set of operations). Compared to the trigram language model, their model showed improvement in WER on three corpora: Wall Street Journal, Switchboard, and Broadcast News.

Probabilistic context-free grammar (PCFG) defines a probabilistic distribution over word strings and can be directly used to build structured language models [91]. However, PCFGs are not adequate for modeling language, since the rule probabilities of PCFGs are conditioned on the left side of the rule instead of on the lexical context, which is vital to language models [93]. To address this problem, PCFGs are extended by associating constitutes with headwords. Roark [91] developed a probabilistic top-down parser based language model. The probabilities of rule expansions were conditioned on the parents and siblings of the left side constitutes of the rule. This language model produced about an 8.5% relative improvement in WER on DAPRA 93 Hub1 test data over that achieved by the tri-gram model trained on the same data.

Charniak [12] developed a language model based on an immediate-head parser. The probability of a parse was estimated by a Trihead model, that is, the probabilities of constitutes were conditional on their lexical heads, parents, and grandparents. The perplexity of the developed language model on Wall Street Journal (WSJ) Penn Treebank was significantly reduced when compared to the tri-gram model.

The SuperARV language model developed by Wang et al. [112] was a highly lexicalized language model based on the Constraint Dependency Grammar. It was a class-based language model, in which each class encoded lexical information as well as syntactic and semantic constraints enforced by the grammar. When applied into all stages of a multi-pass SR system, the SuperARV language model achieved a 6.2% relative WER reduction on DARPA RT-02 CTS evaluation data [113].

These works have effectively demonstrated that incorporating syntactic information into language models can help reduce word error rate. Dependency grammars have been used in some researches like [16, 112] and they can provide useful information.

2.2.3 Language Models Incorporating Semantic Knowledge

The length of the context considered by language models can be increased from several words to entire sentences by incorporating syntactic knowledge. Contextual information that extends beyond the sentence level can be exploited by performing semantic analysis on the text.

Topic language models are commonly used to better describe heterogeneous training data. Training data is clustered into several subsets, each of which has a different topic [45], and then a topic language model is built on each subset. The final model is the interpolation of the general language model and topic language models. Topic language models are also used as adaptation models, in which topics are dynamically identified and only the identified topic models are combined with the general model [103]. Instead of being trained beforehand, topic language models can also be dynamically constructed; words from recent history are used as queries to find relevant documents from other corpora or WWW using information retrieval techniques [101, 18]. Language models are then built on the retrieved documents and interpolated with static language models to reflect current language usage.

Another method for incorporating semantic knowledge focuses on word relations. Semantic relations among words can be caught through word co-occurrence analysis. From the extant work, it is evidenced that language models enhanced with word co-occurrence relations help reduce word error rate.

Trigger pairs [58] are long distance word pairs highly correlated in training documents. They carry a semantic relation that, if one word appears, its highly correlated word will have a higher probability of appearing. When using trigger pairs as features in a maximum entropy model, a 14% WER reduction was achieved on WSJ corpus when compared to using the trigram model alone [92].

Bellegarda [4] integrated latent semantic analysis (LSA) [23, 6] into language models to catch underlying word co-occurrence relationships across documents. After the singular value decomposition (SVD) was applied, low dimension matrixes were generated to capture the correlations between combinations of words and clusters of documents. In this paradigm, every possible combination of words in the vocabulary was a potential trigger [5]. When tested on the WSJ0 data, the LSA based language model achieved a 9% relative word error rate reduction over the baseline trigram [5].

2.3 Error Detection

Although the performance of SR systems improves as techniques for individual components of SR systems advance, the occurrence of errors is still unavoidable. Given the imperfection of SR systems, detecting and correcting SR errors becomes vital to the usage of SR systems. A reliable measure of word correctness, referred to as a confidence measure, is desirable for error identification.

SR systems try to find the most likely word sequence W given the acoustic signal A, which is $\arg\max_W P(W|A)$. According to the Bayes rule, maximizing the probability of $P(W|A)$ is equivalent to maximizing the posterior probability $P(A|W)P(W)/P(A)$, as shown in Equation 2.4

$$W = \arg\max_W P(W|A) = \arg\max_W \frac{P(A|W)P(W)}{P(A)} = \arg\max_W P(A|W)P(W) \tag{2.4}$$

The posterior probability $P(A|W)P(W)/P(A)$ is a theoretically accurate measure of the correctness of words. However, in actual implementations of SR systems, the prior probability $P(A)$ is eliminated in finding the optimal word sequence W. One reason for this elimination is that $P(A)$ is a constant for all possible sequences and its elimination does not change the order of competing sequences. Furthermore, it is not practical to accurately compute $P(A)$ by summing up $P(A|W)P(W)$ on all possible sequences of W. Therefore, the score of each possible sequence is not an absolute measure of the correctness of recognition output, but a relative measure which can only be used to compare competing sequences.

Estimating posterior probability by estimating $P(A)$ thus can be considered a venue for obtaining good measures. Various methods have been proposed to estimate $P(A)$ based on word-lattice (compact representation of alternative hypotheses) output [50, 71, 117] or n-best list (top hypotheses) output [114], both of which provide a certain number of possible sequences and enable computation. This estimated posterior probability can be used as a confidence measure.

The majority of methods for computing confidence measures rely on feature combination. After a brief description of feature combination methods, this section focuses on the selection of features.

2.3.1 Confidence Measures Based on Feature Combination

Four factors should be considered when designing a confidence measure approach: the level of the confidence measure, error definition, features used and the combination mechanism, and the evaluation metrics [14].

Level of confidence measure

Confidence measures can be assigned at different levels of abstraction, such as word level [13, 71], concept level [85], and sentence/utterance level [81]. The level of the confidence measure depends on back-end applications. In spoken dialogue systems, concept or utterance level is usually preferred, because the language understanding component needs to analyze the meaning of utterances to determine the response. In transcription applications, for which correct transcripts are needed, the word level confidence measure is preferable, because the correction of the recognition output is always word based. Word level confidence measure for transcription applications is the case for our research.

Error definition

Errors are usually defined as mismatches between the recognition output and the reference transcript, and the unit of mismatch depends on the level of the confidence measure. In the case of word level confidence measures, errors are mismatched words. In the case of utterance level confidence measures, errors are utterances that are not the same as corresponding utterances in the reference transcript.

Features and combination mechanism

Features are an important part of confidence measure design. Generally, a classifier is used to combine good heuristic features that are related to the correctness of hypotheses. Features can be extracted not only from different components of SR systems (SR dependent features), including acoustic model [13], language model [13], and decoder [70] but also from linguistic analysis of the output (SR independent features). The computed confidence measure can be interpreted either as a probability or as a binary variable [71].

Various classifiers have been used as combination tools. Some examples are listed here: decision tree [13], neural network [114], linear discriminant analysis [81], Support Vector Machine (SVM) [119], regressions [70], and fuzzy inference [37].

Evaluation metrics

Depending on its interpretation and applications, a confidence measure can be evaluated by different metrics. Classification error rate indicates the proportion of wrongly judged words (e.g. [116, 10]), and it can be used no matter when the confidence measure is a continuous or a binary variable. ROC curve measures the trade-off between benefits (true positive rate) and the cost (false positive rate) (e.g. [37, 85]), and it can only be

used when the confidence measure is a continuous variable. Normalized cross-entropy is a metric based on information theory and measures the changes of uncertainty when additional information is included, and it requires that the confidence measure represents the probability of a recognized word to be correct (e.g. [43]).

2.3.2 SR Dependent Features

As we discussed above, SR dependent features are extracted from various components of an SR system. Some commonly used features are listed in Table 2.1, including posterior word probability, path ratio, and so on.

Models	Features
acoustic model	normalized acoustic score [119, 95]
language model	language model score [95, 115]
	language model backoff mode [119, 95]
decoder	posterior word probability [50, 117, 114, 119, 115]
	path ratio: ratio between the number of paths containing the word and the total number of paths in the nbest list [13, 34, 119, 95, 122, 115]
	lattice density: the number of alternative paths to the word in the word graph [116, 50, 95]

Table 2.1: Examples of SR dependent features

These SR dependent features provide useful information. However, the information contained in these features has already been considered when SR systems generate output. A common observation is that using a combination of all SR dependent features can only marginally improve the performance achieved by only using the best single feature [119, 96]. Hence, using merely these features is inadequate, and additional information sources should be considered.

When word level confidence measures are used, selected SR dependent features are usually confined to the word itself. Researchers have also explored contextual information (nearby words) as a source for additional features.

Immediate both-side context

In SR systems, n-gram language models are used to capture the previous short context and may induce errors which render the word sequence plausible while not truly mapping the acoustic signal. N-gram language models only consider the left side context. Including both sides of the context can provide additional information about the correctness of words, and the following works provide such evidence.

Hernández-Ábrego and Marino [37] proposed to examine both the immediate preceding and the immediate following context of a word when identifying recognition errors. The confidence scores and n-gram scores of the contextual words were used to rescale the confidence score of the current word. ROC curves show that on average contextual information can improve error detection by seven points over the entire range of operation points on two Spanish corpora. Their results also show that contextual information is useful for detecting continuous errors and errors consisting of short words.

Duchateau et al. [25] integrated the backward language model score, for which the probability of a word depends on its following words, into the confidence score. Their results show that for a context of five words, higher normalized cross entropy can be achieved by combining a tri-gram forward language model with a tri-gram backward language model than by using a five-gram forward language model on WSJ.

Sarikaya et al. [96] used a maximum entropy based structured language model. Language model scores were computed by examining different context lengths. The results show that larger contexts improve performance, and the length of the context is related to the average length of utterances. The context of three words achieves the best correct acceptance rate at 5% false acceptance rate and outperforms the non-context by 16%.

2.3.3 Linguistic features used for dialogue systems

Because high-level linguistic knowledge is difficult to directly incorporate into the decoding process of SR systems, it is a candidate for additional information sources. When the output of SR systems is post-processed, the whole output is visible, and contextual information can be utilized to extract useful features. Features obtained from high-level language processing, such as syntactic and semantic analysis, can complement the low-level linguistic knowledge (usually n-gram) used in SR systems.

Most research on utilizing linguistic features focuses on utterance level confidence measures. Features are extracted from the parsing products of syntactic or semantic parsers. Examples of features include full/robust/no parse, number of words parsed, gap number, slot number, grammar rule used, etc. [87, 81, 10, 95]. When confidence measures are used in spoken dialogue systems, discourse level features, such as number of turns and dialog state, are also used [10].

Several studies applied the linguistic features to the word-level confidence measure. Zhang and Rudnicky [119] utilized two parser-based features, parsing mode and slot backoff mode, extracted from the parsing product of Phoenix, a semantic parser. They combined these two parser-based features with several SR

dependent features using SVM. Addition of parser-based features from the semantic parser resulted in a 7.6% relative word error rate reduction on data from the CMU Communicator system.

Sarikaya et al. [96] used two sets of semantic features. One set of features, including classer-tag, classer-tag-arc, parser-tag, and parser-tag-arc, was obtained from a statistical classer/parser. The other set of features was from a maximum entropy based semantic structured language model that integrated n-grams and semantic structures. When combined with the posterior probability feature using a decision tree, both sets achieved about 13-14% improvement in correct acceptance at 5% false acceptance over the baseline (posterior probability only) on IBM Communicator system data.

Skantze and Edlund [104] did not use parser-based features but instead focused on lexical features (part-of-speech, syllabus, content words, etc.) and dialogue discourse features (previous dialogue act and whether words are mentioned before). They used transformation-based learning and instance-based learning as classifiers. When combined with confidence score, these linguistic features achieved a 11.9% improvement in classification accuracy over the baseline (confidence score only) on dialogue corpus collected by themselves.

All this research was conducted in conversational dialogue environments for restricted domains, such as ATIS [87], JUPITER [81] and Communicator [10, 95, 119, 96]. The findings indicate that linguistic knowledge can be used as an additional information source to determine the correctness of recognition output in spoken dialogue systems.

2.3.4 Linguistic features used for monologue applications

In the previous section, linguistic knowledge was shown to be useful for detecting errors in spoken dialogue systems, and the majority of linguistic features were extracted from semantic parsers. However, those features cannot be used in monologue applications because semantic parsers are not available for the general domain. For monologue applications, extant research mainly focuses on semantic information.

Cox and Dasmahapatra [21] proposed to exploit semantic analysis for error detection. To investigate human performance in detecting errors based only on semantics, researchers examined the SR output of about 600 sentences from WSJCAM0. They conservatively marked a word as an error only if it was clearly wrong due to incompatible meaning within the sentence context. They achieved 15% recall and 89.6% precision. This suggests that the application of computerized semantic analysis to error detection would follow the same pattern: it can only find a small number of errors but with high precision. The words they identified were uncommon nouns and verbs.

They then applied latent semantic analysis, assuming homogeneity between the training data and the test data. Similarity between two words was computed by the cosine measure. For each word w_i, its mean semantic score $\overline{L_i}$ was the average of its similarity with all other words in a lexicon. To eliminate noise introduced by function words, they developed a stop word list by setting up a threshold L_T on $\overline{L_i}$ and eliminating words with a value higher than L_T. Three different measures were used to compute the semantic similarity of word w_i in an utterance with length N:

- $MSS_i = \frac{1}{N}\sum_{j=1}^{N} S(w_i, w_j)$: mean semantic scores
- $MR_i = \frac{1}{N}\sum_{j=1}^{N} Rank(S(w_i, w_j))$: mean rank of the semantic scores
- $PSS_i = \prod P(L_i \leq S(w_i, w_j))$: probability of the semantic scores, the distribution of L_i was modeled by 5 component Gaussian mixtures

The performance of above measures was assessed using classification error rate (CER) on remaining words (after eliminating words with L_T) and its relative improvement over the error rate of SR system. All three measures produced similar results, and PSS showed the marginally best result. In varying L_T, the highest relative improvement 7.1% was obtained when $L_T = 0.25$ under which 53.6% words were eliminated. Analysis of the effect semantic values on the determination of word correctness showed that PSS is a good indicator of word correctness but not word incorrectness. High PSS scores significantly correlate with word correctness.

Although gain from LSA is small, semantic analysis was completely independent of SR and could be combined with other SR dependent information. Combining PSS with N-Best (NB) by multiplying their values, $NB * PSS$, produced a slightly better performance than NB at high recall and maintained PSS's high precision at low recall.

Inkpen and Désilets [44] pursued the idea of Cox and Dasmahapatra's [21] and explored different semantic measures. To choose a semantic measure, they compared dictionary-based measures (Roget-based, WordNet-based) and distributed-based measures (cosine, Pearson correlation, Point-wise Mutual Information (PMI)) by evaluating them on two sets of word pairs (Miller and Charles's 30 word pairs [75] and Rubenstein and Goodenough's 65 word pairs [94]). As a result, Roget-based edge counting and PMI were selected as the candidate measures.

Similar to the work of Cox and Dasmahapatra, Inkpen and Désilets eliminated noise by setting up a stop list of 779 words. The semantic score of a word w was computed via the following method: 1) getting w's

neighborhood $N(w)$, which was confined by the window size (all content words or 10 content words); 2) computing the semantic similarities of all possible content word pairs in $N(w)$ using the selected measures (Roget-based or PMI); 3) for each word w_i in $N(w)$, computing its semantic value SC_{w_i} by aggregating (average, maximum, or average of top 3 (3Max)) the semantic similarity between w_i and all other words in $N(w)$. To find out if w was an error, the average of SC_{w_i} over all w_i in $N(w)$, SC_{avg}, was computed, and w was identified as a recognition error if $SC_w < K * SC_{avg}$, where K was a threshold.

The proposed method was tested on 100 stories from the TDT2 English audio corpus recognized by two SR systems: NIST/BBN time-adaptive speaker independent SR with a WER 27.6% and Dragon NaturallySpeaking speaker dependent SR with a WER 62.3%. The comparison of PMI with the Roget-based method in forms of Precision-Recall curves showed that PMI had better performance when the window was composed of all words and the average aggregating method was used. Using PMI, two window sizes produced almost the similar results, with the exception of the 10-word window size which achieved marginally better results on maximum and 3Max methods. In the all words window setting, both maximum and 3Max performed better in the high precision area, though 3MAX outperformed maximum. Given that precision is more important than recall to the designated audio browsing application, 3Max was chosen. Therefore, PMI combined with a 10-word window size and 3MAX was chosen as a better configuration.

With the above configuration, the proposed method was shown to significantly reduce the content word error rate with the cost of losing some correct words. For BBN transcripts, the content word error rate was reduced by 50% when losing 45% of correct words. To achieve a similar error reduction for the Dragon transcription, 50% of correct words were lost.

2.4 Error Correction

The goal of error correction is to produce error-free SR output by repairing recognition errors. Typically in manual correction, error correction is used as a single, final step in the three-step error handling process (which consists of detection, navigation, and correction) [100]. In automatic correction, error correction generally combines with error detection implicitly. In this section, we briefly describe previous work on automatic error correction.

Error correction differs from error prevention in that the former aims to correct errors from the recognition output after they have occurred. Various methods have been proposed to automatically correct errors. Some

methods utilized alternative hypotheses and their associated information, while others exploited information totally independent of SR systems.

2.4.1 Replacing output with an alternative hypothesis

Methods utilizing alternative hypotheses try to find out corrections from the hypotheses SR systems generated. The replacement could be happened at utterance level or word level. One common limitation of this kind of method is that errors induced by OOV words could never be corrected because corrections do not appear in the hypotheses.

Replaced with an utterance hypothesis

Setlur et al. [102] proposed to correct recognition errors by replacing the output utterance with its second best hypothesis if necessary. An utterance verification algorithm was used to assign each utterance a confidence score. If the confidence score of an utterance was above a threshold, no replacement would take place. Otherwise, if the confidence score of the second best hypothesis was above the threshold, the second best hypothesis would be selected as the output. Their experiment on a connected digit recognition task showed that, at the threshold corresponding to the minimum WER, the WER could be reduced from 1.15% to 1.02%. However, at a higher threshold, more errors would be introduced than corrected.

Zhou et al. [123] proposed a multi-pass error detection and correction framework, in which errors were corrected after being detected. Based on the characteristics of Mandarin, characters rather than words were the basic unit of recognition. To detect errors, various SR dependent features were used. For each detected error, a list of top 20 alternative characters was generated from the lattice. For each utterance, all possible paths by replacing detected errors with their hypotheses were treated as utterance hypotheses. Each utterance hypothesis was scored by the linear interpolation of its trigram score and mutual information (MI) score. MI score of an utterance was the average of MI scores of target words in the utterance, and MI score of a target word was the average of its MI values to all other words in the utterance. The MI value between two words was got from training corpus and computed as their co-occurrence rate in the context of an utterance. The utterances were then ranked based on their scores and the candidate character of the detected error in the top utterance was used as the correction. To minimize the negative effect caused by imperfect error detection, a threshold function was trained such that a correction is allowed only when the score difference between the original and correction was higher than the threshold. The threshold was trained using grid search to

get the optimal performance on training data. The proposed method was tested only on slightly erroneous utterances which had a maximum of one to four detected errors, because both the MI and trigram scores depend on reliable context. The results showed that the rate of correct words in the detected errors improved from 35.5% to 40.6%. The overall error detection and correction procedure can improve character error rate of the slightly erroneous group from 20.1% to 19.3%.

Replaced with a word hypothesis

Mangu and Padmanabhan [72] developed an error correction mechanism based on the observation that the second best alternative sometimes is the correct word. Features, such as word posterior probability and word duration, were extracted from the confusion network for the top two word hypotheses. Transformation-based learning (TBL) was used to learn rules that indicated when to output the second best alternative from the confusion network as the true output word. Their experiment on the Switchboard corpus showed a significant WER reduction compared to the baseline consensus decoding approach. However, this method confined word choice to only the first two hypotheses.

2.4.2 Pattern Learning

Methods belonging to this group try to find error patterns from existing data corpora that include both recognition output and reference transcripts. Generally, training data should be in the same domain as test data. Data sparsity is a common problem for this method.

Kaki et al. [49] proposed to use error patterns and similar strings to correct recognition errors. Error patterns were extracted from the string mappings between reference transcripts and recognition output, and satisfied certain conditions. Similar strings were strings with certain length and occurring several times (more than threshold) in training data. The erroneous recognition output was then compared to error patterns and similar strings to find out the corrections. The experiment results showed that the number of errors decreased by 8.5% after performing error corrections on a Japanese travel management corpus.

Ringger and Allen [89] adopted the statistical noisy channel model to discover statistical error patterns from data corpus. The fertility channel model [90] captured one-to-one, one-to-many, many-to-one and many-to-many alignments between outputs and references. They suggested that a post-processor trained on a specific domain can adapt the generalized SR system to that domain. Their experiment results showed that their methods achieved a relative word error rate reduction as high as 24.0% for the TRAINS-95 system.

When processing the output of an SR system, the language model of which was trained in the same domain, the proposed method can still achieve significant improvement. However, only the best single output was considered in their method, and the word-lattice or n-best list, which could potentially provide more useful hypotheses, was ignored.

Jung et al. [48] extended Ringger and Allen' idea and proposed to alleviate the data sparsity problem by using smaller unit syllables instead of words when constructing the channel model for Korean. They used the maximum entropy language model integrated with some high-level linguistic knowledge to rescore the hypotheses. Their results showed a 42% relative improvement in WER on a corpus of queries to an in-vehicle navigation system.

2.4.3 Exploiting contextual information

Context is an information source which can be used to correct recognition errors. Co-occurrence analysis has been utilized to catch the contextual information. As we described early, Zhou et al. [123] applied mutual information to exploit the co-occurrence between words in the context of an utterance to correct the errors.

Sarma and Palmer [97] proposed an unsupervised method based on the analysis of statistical lexical co-occurrence of speech output. Their method exploits global contextual information. The method is based on the idea that a word usually appears with some highly co-occurred words, and recognition errors of the word also appear with the same set of words. Co-occurrence relations were analyzed using a large corpus of SR output on Broadcast News. Given a topic word, by sliding a window of a certain length and counting contextual words in the window, words highly concurrent with contextual words were detected as possible errors, and the word with the most similar phone was changed to the topic word. Only three words were used in the experiment, and preliminary results show that it is possible to achieve high precision with reasonable recall. For example, when using "Iraq" as a query word, the method achieved 95% precision and 61% recall when window size was fourteen and the minimum required context words was two. In addition, the applicability of the method is limited in that it can only make corrections for errors of a specific topic word such as the query word used in the spoken document retrieval.

2.5 Summary

In this chapter, we first introduced the causes of SR errors and then reviewed advanced techniques in building language models to prevent error occurrence. Next, we discussed previous work on error detection and correction.

Linguistic knowledge has demonstrated its efficacy in improving SR output by augmenting language models to prevent error. Linguistic knowledge, which is complementary to knowledge already used by SR systems, has also been used for error detection. Linguistic knowledge is widely used in spoken dialogue systems and has shown its complementary effect in detecting incorrect words, utterances and concepts. Little work has been done on the role of linguistic knowledge in monologue applications. Nevertheless, preliminary evidence has shown the promise of linguistic knowledge for monologue applications.

Chapter 3

DICTATION CORPORA

Two dictation corpora, representing different dictation contexts, were used to evaluate the proposed techniques in this dissertation. One corpus was collected from a user study, and the other was extracted from a standard speech corpus. We expected to provide a comprehensive validation of the proposed techniques by using two different corpora, which are described in detail in this chapter.

3.1 Dictation Corpus Collected from a User Study

We refer to the corpus from a user study as the Study3 corpus, which was collected from a composition dictation task using TkTalk conducted by Feng et al. [29]. TkTalk is a customized speech recognition application built on the IBM ViaVoice speech recognition engine (Millennium Edition) and developed by the Interactive Systems Research Center at UMBC. TkTalk interacts with the speech recognition engine and presents recognition output in the form of either a single best hypothesis or an n-best list for both words and utterances.

The study3 corpus contains dictation recognition from 12 native English speakers in a quiet lab environment using high-quality microphones. All participants spoke on the same topic, daily correspondence in office environments. The participants went through an enrollment process before dictation. They were not allowed to make inline error corrections during dictation. The performance of speech recognition on the corpus is reported in Table 3.1.

In Table 3.1, reference word accuracy is the ratio of number of correctly recognized words to the total number of words in the reference, as shown in Equation 3.1. Words in the reference can only be in three

Participant	Reference Word Accuracy (%)	Substitution Error Rate (%)	Deletion Error Rate (%)	Insertion Error Rate (%)	Word Error Rate (%)
S1	59.5	34.5	6.0	1.5	42.0
S2	86.0	12.9	1.1	14.6	28.6
S3	93.9	6.1	0.0	5.8	11.8
S4	87.2	12.3	0.5	2.1	14.9
S5	88.6	8.8	2.7	1.3	12.8
S6	83.9	14.8	1.4	4.3	20.5
S7	95.7	3.5	0.8	1.1	5.3
S8	88.3	9.8	1.9	2.2	13.9
S9	94.0	5.3	0.7	1.9	8.0
S10	79.9	13.8	6.3	0.9	21.0
S11	85.9	13.1	1.0	4.8	18.9
S12	93.4	5.7	0.8	2.5	9.0
Avg.	86.4	11.7	1.9	3.6	17.2

Table 3.1: Recognition performance of the study3 corpus by participant

states: correctly recognized word, substitution error, and deletion error. Therefore, reference word accuracy, substitution error rate, and deletion error rate add up to 100%.

$$reference\ word\ accuracy = \frac{\#\ of\ correctly\ recognized\ words}{\#\ of\ words\ in\ the\ reference} \tag{3.1}$$

It can be observed from Table 3.1 that there is significant variation in performance among the participants. Although the lowest word error rate achieved is 5.3%, the average word error rate of the corpus is much higher at 17.2%. The insertion error rate is higher than the deletion error rate, and consequently the number of output words is larger than that of reference words.

Because the study3 corpus was collected in a live mode, the corpus sometimes contain commands such as “NEW-PARAGRAPH” and “NEW-LINE”. Such command words were used as delimiters in segmenting dictation documents into sentences and were deleted, which yielded a total of 4813 output words. In addition, punctuation marks such as “.”, “?”, and“!” were also used in sentencing the documents. Syntactic analysis was conducted on the segmented sentences despite their potential inaccuracy due to SR errors.

The descriptive statistics of output words and their recognition accuracies are reported in Table 3.2. The average length of dictations is around 400 words, the length instructed to the participants prior to study. Participants achieved varied output word accuracy ranging from 61.64% to 95.47%.

Participant	# of Output Words	Output Word Accuracy (%)	# of Sentences	Average # of Words per Sentence	# of Topics
S1	451	61.64	44	10.25	1
S2	416	75.48	18	23.11	1
S3	356	88.76	18	19.78	1
S4	423	85.82	21	20.14	1
S5	369	89.70	21	17.57	1
S6	452	81.19	30	15.07	1
S7	375	95.47	25	15.00	1
S8	366	88.25	26	14.08	1
S9	419	93.08	24	17.46	1
S10	414	84.30	24	17.25	1
S11	404	82.18	28	14.43	1
S12	368	91.58	17	21.65	1
Avg.	401	84.79	24.67	17.15	1

Table 3.2: Descriptive statistics of the study3 corpus by participant

3.2 Dictation Corpus by Processing Sentences from a Standard Speech Corpus

The dictation sentences were extracted from the CSR-II corpus [1]. CSR-II corpus represents the Wall Street Journal-based Continuous Speech Recognition Corpus Phrase II and is also referred to as WSJ1 corpus. It was collected for APRA benchmark tests on large vocabulary, speaker-independent, continuous speech recognition systems in 1992-1993 [59]. The WSJ1 corpus includes both reading speech and spontaneous dictation speech. Spontaneous dictation sentences which conformed to the Wall Street Journal news in both topic and style were dictated by journalists with varying degrees of experience in dictation.

Spoke 9 in APRA 1993 Continuous Speech Recognition (CSR) Hub and Spoke benchmark test [52] was designed to evaluate speech recognition systems on the spontaneous dictation style speech. The test set used in Spoke 9 consisted of 200 spontaneous dictation sentences from 10 journalists and was used in this dissertation.

To generate consistent output as that of the study3 corpus, a system was developed by customizing an offline application provided by the ViaVoice Tcl API to process the 200 sentences automatically. The corpus (referring to as the WSJ corpus in this dissertation), consisting of the single best hypothesis and n-best list for both words and utterances, was generated. The performance of Speech recognition on WSJ is summarized in Table 3.3.

[1] Available from `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC94S13A`

Participant	Reference Word Accuracy (%)	Substitution Error Rate (%)	Deletion Error Rate (%)	Insertion Error Rate (%)	Word Error Rate (%)
S1	91.8	6.9	1.2	2.0	10.2
S2	80.5	11.7	7.7	1.9	21.3
S3	89.2	9.3	1.5	6.7	17.5
S4	96.0	3.0	0.9	0.9	4.9
S5	76.0	17.9	6.1	1.8	25.8
S6	93.9	5.9	0.2	2.9	9.0
S7	88.1	6.8	5.1	1.3	13.2
S8	84.5	11.6	3.9	1.8	17.4
S9	84.9	8.8	6.3	6.1	21.3
S10	89.8	7.0	3.2	1.8	12.0
Avg.	87.7	8.8	3.5	1.8	15.1

Table 3.3: Recognition performance of the WSJ corpus by participant

As seen with the study3 corpus, significant variation in performance was also observed among the participants. The lowest word error rate achieved was 4.9%, and the highest was 25.8%. The overall word error rate of the WSJ corpus is lower than that of the study3 corpus. Unlike the results with the study3 corpus, the insertion error rate of the WSJ corpus is lower than its deletion error rate.

Since the WSJ corpus was organized by sentences, syntactic analysis can be directly performed on them. In addition, journalists were instructed to dictate on several topics of their own choice and to conclude each topic in several sentences. To enable semantic analysis beyond the scope of a single sentence, we manually grouped the sentences into topics by going through every sentence. The descriptive statistics of the corpus are shown in Table 3.4. There are 4502 output words in total, which is comparable to the study3 corpus in size. There are 4.4 topics per journalist on average. Compared to the study3 corpus, the WSJ corpus has a higher output word accuracy. The average sentence length in words in the WSJ corpus is longer than that of the study3 corpus.

Participant	# of Output Words	Output Word Accuracy (%)	# of Sentences	Average # of Words per Sentence	# of Topics
S1	406	91.13	20	20.30	4
S2	353	85.55	20	17.65	5
S3	487	84.80	20	24.35	4
S4	427	96.02	20	21.35	4
S5	267	79.40	20	13.35	6
S6	456	91.45	20	22.80	4
S7	438	91.55	20	21.90	5
S8	744	86.29	20	37.20	4
S9	488	85.04	20	24.40	4
S10	436	91.06	20	21.80	4
Avg.	450	88.23	20	22.51	4.4

Table 3.4: Descriptive statistics of the WSJ corpus by participant

Chapter 4

CONFIDENCE MEASURE INCORPORATING NON-LINGUISTIC FEATURES

In this chapter, a confidence measure based on combination of non-linguistic features is introduced. This chapter serves as a baseline for the following two chapters, in which additional features will be gradually incorporated while keeping the combination mechanism and evaluation methods the same.

Our confidence measure is developed at the word level and predicts the correctness of each output word. The correctness of an output word is judged by referring to manual transcriptions. In this chapter, only non-linguistic features directly extracted from speech recognition output are considered. Support Vector Machine is selected as the classifier to combine different features, and the error detection performance is evaluated with classification error rate and precision/recall/F measure.

4.1 Non-linguistic Features

Non-linguistic features, which can be extracted directly from speech recognition output, are related to confidence score (CS). Based on the results of a previous study [120], two groups of features were considered and are listed in Table 4.1.

- Raw Confidence Score (RCS). Features in RCS are directly derived from CS. They indicate that output words with low CS are more likely to be errors.
- Contextual Confidence Score (CCS). Features in CCS model the relationship between an output word and its surrounding words. Continuous errors are common in recognition outputs and may result from

Group	Feature	Description
RCS	CS	word confidence score
	range	normalized z-score CS
CCS	utterance	length: the length of an utterance that contains the word
		position: the position of the word in an utterance
		sum CS: sum of CSs of words in the utterance
		dev CS: the standard deviation of CSs of words in the utterance
	neighbor	PCS_i: difference in CS between the current word and the word preceding it with distance i, $i = (1...3)$
		PCS_0: difference in CS between the current word and the average of preceding 3 words
		SCS_i: difference in CS between the current word and the word following it with distance i, $i = (1...3)$
		SCS_0: difference in CS between the current word and the average of following 3 words

Table 4.1: Non-linguistic features used in confidence measure

different factors, such as an incorrect segmentation of speech. Thus, the difference in CS between the current word and other words in the context may provide a clue as to the correctness of the current word.

4.2 Machine Learning Technique - Support Vector Machines

Support Vector Machine was chosen as the classifier mainly because it has been shown to exhibit performance comparable or superior to traditional models in various applications [9]. The basic idea of SVMs is to find optimal hyper-planes in the feature space. In the case of binary classes, if data are linearly separable, SVM classifiers learn a pair of optimal hyper-planes with maximum margin in the original feature space as boundaries to separate two classes. In many cases when data are not linearly separable and linear boundaries are not appropriate, SVM classifiers use kernel functions to implicitly map the original feature space into a higher dimensional space in which optimal hyper-planes can be found. Commonly used kernel functions include Linear, Polynomial, Gaussian and Sigmoid kernels. Because Gaussian kernel had the best error detection performance in [120] which is the base of current confidence measure, the following experiment results were generated by the Gaussian kernel. LibSVM [1] [11] was used as the SVM tool in this dissertation.

[1] Available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

		True Value	
		Correct Word	Recognition Error
Predicted Value	Correct Word	TP	FP
	Recognition Error	FN	TN

Table 4.2: Grouping words based on true and predicted values

4.3 Evaluation Metrics

The word error rate (see Equation (2.1)) commonly used in speech recognition systems is not suitable for the error detection task for two main reasons: 1) error detection is based not on the reference transcript but on the output hypothesis; and 2) deletion errors are not identifiable by a classifier, which is only intended for words with an explicit output. Therefore, we adopted a revised word error rate [119], which is defined in Equation 4.1. The assumption that all output words are correct allowed this revised word error rate to serve as the baseline for the error detection performance.

$$Revised\,Word\;Error\;Rate = \frac{\#\;of\;substitute\;errors + \#\;of\;insertion\;errors}{\#\;of\;words\;in\;hypothesis} \tag{4.1}$$

The output words can be categorized into four groups based on the relationships between their true values and predicted values from the classifier, as shown in Table 4.2. TP is true positive and indicates that correct words are successfully predicted as correct words. TN is true negative and indicates that recognition errors are successfully predicted as errors. FP is false positive and indicates that recognition errors are mistakenly predicted as correct words. FN is false negative and indicates that correct words are mistakenly predicted as errors.

The overall performance of SVM was then measured by the classification error rate, which is the percentage of output words that are wrongly classified. Based on the grouping of words in Table 4.2, the classification error rate can be computed with Equation 4.2.

$$Classification\;Error\;Rate = \frac{FP + FN}{TP + FP + FN + TN} \tag{4.2}$$

In addition to the overall performance, the performance in detecting errors is also important. To this end, we used Precision (PRE), Recall (REC) and F measure (F), which are defined in Equation 4.3. PRE is the

percentage of words classified as errors that are in fact recognition errors. REC denotes the proportion of actual recognition errors that are categorized as errors by the classifier. A single value measure, F, combines precision and recall to achieve a trade-off between the two. We used F_1 measure, which treats precision and recall equally. The baselines of PRE, REC, and F are all 0 as all the output words are assumed to be correct.

$$
\begin{aligned}
Precision &= \frac{TN}{TN + FN} \\
Recall &= \frac{TN}{TN + FP} \\
F &= \frac{2 * PRE * REC}{PRE + REC}
\end{aligned}
\tag{4.3}
$$

4.4 Experiment Results

A leave-one-out scheme was used in the experiment. For the study3 corpus, the data from one participant was used as test data, and the data from all the other participants was used as training data. The process was repeated by using every participant's data as the test data. For the WSJ corpus, the data from one speaker was used as test data, and the data from the rest of the speakers was used as training data. The above process was also repeated by using the data of every speaker in the corpus as test data. For convenience, we use "CS" to stand for non-linguistic features.

Table 4.3 reports the experiment results, in which Table 4.3(a) reports descriptive statistics of CER, PRE, REC, and F measure achieved by CS, Table 4.3(b) reports relative improvement of CER achieved by CS over baseline, and Table 4.3(c) reports paired t-test results of CER between CS and baseline. The average performance of all the runs is reported.

Compared to the baseline, for the study3 corpus, non-linguistic features achieved a 17.36% relative improvement in CER, which is significant at 0.05 level; for the WSJ corpus, non-linguistic features achieved a 3.06% relative improvement in CER, which is not significant. On both corpora, using CS features can help find a small number of errors with precision higher than 50%.

(a) Descriptive statistics of CERs, RECs, PREs, and Fs

	Study3		WSJ	
	baseline	CS	baseline	CS
Mean[Std.] CER (%)	15.21[9.17]	12.57[6.36]	11.77[4.82]	11.41[4.38]
Mean[Std.] PRE (%)	0	62.91[17.97]	0	58.20[22.38]
Mean[Std.] REC (%)	0	27.96[12.76]	0	9.63[4.92]
Mean[Std.] F (%)	0	38.08[15.07]	0	16.35[7.96]

(b) Relative improvement (%) of CERs by CS over baseline

	Study3	WSJ
baseline	17.36	3.06

(c) Paired t-tests (two-tailed) on CERs between CS and baseline

	Study3	WSJ
t	-2.686	-1.504
p	0.022 **	0.167

** Significant at .05 level

Table 4.3: Experiment results of confidence measure on both the Study3 corpus and the WSJ corpus when non-linguistic features were used

Chapter 5

CONFIDENCE MEASURE INCORPORATING SYNTACTIC KNOWLEDGE

As discussed in Section 2.3, a common observation for confidence measures based on feature combination is that the combination of all SR-dependent features can only marginally improve the performance achieved by using only the best single feature [119, 96]. This marginal improvement can be attributed to the overlapping information conveyed by these features. In other words, much of the SR-dependent information has already been used by an SR system to generate text output. Therefore, additional features from complementary information sources are desired.

As evidenced by a user study on error detection conducted by Skantze and Edlund [104], linguistic knowledge and related context information are frequently used by humans to find errors in speech recognition output. Let the range of error detection score (the ratio of the sum of the number of correctly detected errors and the number of untouched correctly recognized words to the total number of words in SR output) be 0 to 1, where 1 indicates that all errors are detected and 0 indicates that no error is detected. When only individual utterances extracted from SR transcripts of dialogues were given, human participants were able to achieve a mean error detection score higher than 0.6. This score is higher than the baseline score of 0.5. When the preceding and/or following utterances of each utterance were given, participants were able to improve the mean detection score to around 0.7. Therefore, linguistic knowledge and context information are very helpful for humans to find errors, and they could be promising information sources which can be exploited to improve the performance of automatic error detection.

Syntactic analysis and semantic analysis are two methods to utilize linguistic knowledge and context

information. In the current and succeeding chapters, we introduce features extracted from syntactic and semantic analysis respectively and evaluate the value of those features in improving confidence measure.

5.1 Linguistic Knowledge Used by Humans in Error Detection

In order to investigate which types of knowledge may be useful for detecting SR errors, we conducted a user study. In this section, we only briefly introduce the experiment setup and findings, and the detailed description can be found in [121]. In this study, each participant was asked to identify errors in a selection of SR transcripts and to provide justifications for those errors. Two stages of qualitative analysis were performed on the collected data: 1) A research assistant not directly involved in this project and I manually mapped the participants' justifications for error detection to a list of linguistic cues independently, and 2) a third analyst consolidated the mapping results.

The SR transcripts used in the study were extracted from two corpora, namely the study3 corpus and the study2 corpus. The study2 corpus is a dictation corpus and was collected under a setting similar to that of the study3 corpus. The study2 corpus differs from the study3 corpus in that (i) the study2 corpus used more scenarios, and (ii) when the study2 corpus was collected participants were allowed to make the inline corrections during dictation [99]. Eight paragraphs were randomly selected from the SR transcripts of two task scenarios based on two criteria: recognition accuracy and paragraph length (measured by number of words). Specifically, the overall average recognition accuracy (84%) and the length of a medium-sized paragraph (90 words) of the corpus were used as reference values. The number of sentences per paragraph ranged from three to six.

Each participant went through the experiment via the following steps. First, an error annotation schema and several sample transcripts were provided to the participants. Second, the participant was required to master the annotation schema and understand the SR output by passing a pre-test before they could proceed with the error detection task. Finally, the participant was asked to detect and annotate errors in all eight pre-selected paragraphs and declare his/her justifications.

For each paragraph, its sentences were presented to the participants all at once. Moreover, the eight paragraphs were presented to the participants in three settings determined by the type of additional information provided. Three paragraphs were provided with no additional information, three were provided with alternative hypotheses, and two were provided with both dictation scenarios and alternative hypotheses. All the text

Cues	Support		Cues	Support	
	all	correct		all	correct
making no sense	0.32	0.36	phrase structure	0.26	0.26
sentence structure	0.16	0.19	hypotheses	0.15	0.18
confidence scores	0.15	0.18	POS confusion	0.10	0.12
incompatible semantics	0.08	0.10	highest confidence score	0.08	0.09

Table 5.1: Top knowledge cues used by humans in error detection

transcripts were supplied in hardcopy workbooks, on which each participant marked errors and made notes. The sequence of paragraphs and the presentation settings were randomized for each participant. None of the ten participants who completed the study were professional editors, and all were native English speakers.

In order to measure the effectiveness of the linguistic cues used by participants, we used a quantitative measure *support*. The measure *support* was defined as the ratio of the frequency of a cue used by the participants to the total number of errors detected by the participants under a presentation setting to which the cue is applicable. The top eight cues are listed in Table 5.1. For each cue, two variations of *support* are provided, namely *all* and *correct*. The former was computed based on all the potential errors detected by the participants and the latter was computed based on only the errors correctly detected by the participants.

The cues in Table 5.1 can be categorized as the following:

- *making no sense* and *incompatible semantics* are semantic level cues, indicating that either erroneous words are irrelevant to the meaning of sentences or erroneous words cause incompatible meanings between two constituents.
- *phrase structure* and *sentence structure* are syntactic level cues, indicating that erroneous words are likely to cause ungrammatical phrases or sentences.
- *POS confusion* is a morphological level cue, suggesting that erroneous words differ from their reference words not only in lexical forms but also in their parts-of-speech.
- The remaining three features were not generated from linguistic analysis but drawn from alternative hypotheses and their confidence scores.

The above results show that participants used various levels of linguistic knowledge in error detection. To further test the effectiveness of linguistic knowledge, we computed the participants' performance under the condition with both dictation scenarios and alternative hypotheses. On average, participants identified 71.06% errors with the precision of 86.74%.

5.2 Features Based on Syntactic Analysis

Word recognition errors may result in ungrammatical sentences given the assumption that speakers follow grammatical rules when speaking. It is reasonable to assume that ungrammatical sentences may be indicative of errors in dictation because the aim of the dictation process is to generate written style documents.

Although semantic parsers have been used in spoken dialogue systems, they are usually domain-specific, and there is no general purpose semantic parser fitting all domains. Fortunately, syntactic parsers are known to be domain independent. Therefore, syntactic parsers are used in this research to model the above assumption. In order to choose a proper syntactic parser, we considered two criteria:

- Robust parsers are preferred. This is because errors in recognition output may render an output sentence ungrammatical and halt a parser.
- Lexicalized parsers are preferred. Since our task is to judge the correctness of each word, word-level confidence measures are needed, and features should be extracted at the word-level. In the output of a lexicalized parser, syntactic information is associated with each word, and lexicalized features can be easily extracted from the parsing output.

Based on these two criteria, two general-purpose lexicalized parsers were selected, namely *Link Grammar* [1] and *Minipar* [2]. Two groups of features were extracted. One group of features, extracted from the output of Link Grammar, characterizes the dependence relations between a word and other words in a sentence. The other group of features, extracted from the output of Minipar, mainly models the strength of associations between words that have syntactic dependence relationships.

5.2.1 Features Extracted from Link Grammar

Link Grammar

Link Grammar is a context-free lexicalized grammar without explicit constituents [105]. In Link Grammar, rules are expressed as link requirements associated with words. A link requirement is a set of disjuncts. Each disjunct represents a possible usage of a word, and it consists of two ordered lists: a left connector list and a right connector list. An ordered pair of words S and T can be connected by a labeled link if and only if S has a disjunct with a right connector list that matches the left connector list of a disjunct of T. A sequence of

[1] Available from `http://www.link.cs.cmu.edu/link/`
[2] Available from `http://www.cs.ualberta.ca/~lindek/minipar.htm/`

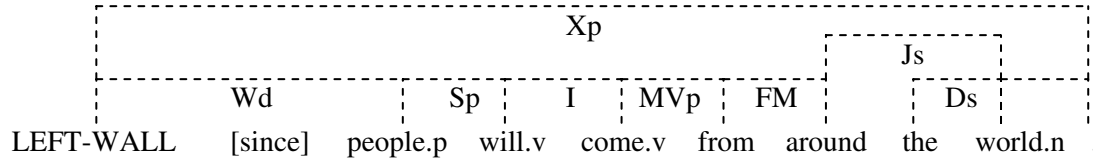


Figure 5.1: Sample parsing output of Link Grammar

words belongs to the grammar if the result linkage is a planar, connected graph in which at most one link is between each word pair and no cross-link exists. Link Grammar also supports robust parsing by incorporating null links [35].

Since speech recognition errors may result in ill-formed sentences, we hypothesize that a word without any link in a linkage of the sentence is a good indicator of the occurrence of errors. Either the word itself or other words around it are likely to be errors. It has been shown that null links can successfully ignore false starts and connect grammatical phrases in ungrammatical utterances randomly selected from the Switchboard corpus [35].

Features

In view of the lexicalization of Link Grammar, the following link-related long-range grammatical features were extracted for each word w:

- Left/right link: the number of links connecting w to its left/right word
- Left/right link of the preceding/succeeding ith word of w: suppose u is the word preceding/ succeeding w with a distance of ith, $(i = 1...3)$. This feature represents the left/right link of u.
- Left/right link of the word with the shortest left/right link to w: suppose u is connected to w by left/right link and has the shortest distance to w among all words connecting to w. This feature represents the left/right link of u.

A sample parsing output is illustrated in Figure 5.1. Links are represented as dotted lines, and annotated with link types (e.g., Wd, Xp). According to the parsing result, the word *since* has no link. The word *around* has one left link and one right link. The immediate preceding word of *around* is *from*, which also has one left link and one right link. The immediate succeeding word of *around* is *the*, which has no one left link and one right link. The word that has the shortest right link to *around* is *world*, which in turn has two left links and one right link.

5.2.2 Features Extracted from Minipar

Minipar

Minipar is a principle-based parser and a descendant of PRINCIPAR [63]. It can generate a dependency parse tree for each input sentence [68]. A dependence tree is composed of a list of dependency relations, which model the binary asymmetric relations between word pairs in the sentence. A dependency relation is represented as a triple in the form of *(head, type, modifier)*, which means that a word *head* is modified by another word *modifier* through a certain type of relation *type*. A head word can have more than one modifier word associated with it, while a modifier word can only have one head word. *type* consists of the syntactic relation and parts-of-speech of both the head and the modifier, and has the format of *headPos:syntacticRelation:modifierPos*. A sample parsing output of Minipar is shown in Figure 5.2: Figure 5.2(a) shows the constituency tree and Figure 5.2(b) lists the dependency triples.

For a grammatically correct sentence, every word in the sentence is usually connected with other words through dependency relations. However, for a sentence containing recognition errors, some words may be isolated from all other words or be connected to other words via improper types of dependency relations. The properness of dependency relations could be represented through the strength of word associations, which were measured by mutual information in this research.

Mutual Information

Mutual information, also called pointwise mutual information, is a measure based on information theory. Given two events x and y, which have the probability distribution $p(x)$ and $p(y)$ respectively, "the amount of information provided by the occurrence of the event represented by y about the occurrence of the event represented by x" [27] is defined by Equation 5.1.

$$I(x, y) = \log \frac{p(x|y)}{p(x)} \tag{5.1}$$

where $p(x|y)$ is the posterior probability. With simple manipulation, Equation 5.1 can be rewritten as Equation 5.2, where $p(x, y)$ is the joint probability of event x and event y.

(								
1	(since	~	SentAdjunct	*				)
E0	(()	fin	C	1	comp1	(gov since)		)
2	(people	~	N	4	s	(gov come from)		)
3	(will	~	Aux	4	aux	(gov come from)		)
4	(come	come from	V	E0	I	(gov fin)		)
5	(from	~	U	4	lex-mod	(gov come from)		)
E2	(()	people	N	4	subj	(gov come from)	(antecedent 2)	)
6	(around	~	A	4	guest	(gov come from)		)
7	(the	~	Det	8	det	(gov world)		)
8	(world	~	N	4	obj	(gov come from)		)
9	(.	~	U	*	punc			)
)								

(a) Parse tree

since	SentAdjunct:comp1:C	fin
fin	C:i:V	come from
come from	V:s:N	people
come from	V:aux:Aux	will
come from	V:lex-mod:U	from
come from	V:subj:N	people
come from	V:guest:A	around
come from	V:obj:N	world
world	N:det:Det	the

(b) Dependency triples

Figure 5.2: Sample parsing output of Minipar

$$I(x,y) = \log \frac{p(x|y)}{p(x)} = \log \frac{p(x|y)p(y)}{p(x)p(y)} = \log \frac{p(x,y)}{p(x)p(y)} \tag{5.2}$$

From Equation 5.2, $I(x,y) = I(y,x)$, which means "the information provided by y about x is equal to the information provided by x about y." [27] $I(x,y)$ is then referred to as the mutual information between x and y. When the joint probability $p(x,y)$ is higher than the product of $p(x)$ and $p(y)$, $I(x,y) > 0$, which indicates that x and y are more likely to occur together than to occur separately. When x and y are statistically independent, $p(x,y) = p(x)p(y)$, which results in $I(x,y) = 0$. When the joint probability $p(x,y)$ is lower than the product of $p(x)$ and $p(y)$, $I(x,y) < 0$, which indicates that x and y are less likely to occur together than to occur separately.

Mutual information has been used in many natural language applications. It has been used to measure word associations which are based on various relations ranging from lexical-syntactic relations to semantic relations [20]. Quantified word associations can be directly used to detect speech recognition errors [44], to find similar words or concepts [38, 64] that can be used to estimate unobserved co-occurrence from similar co-occurrence [22], to discover word senses from text [80], etc. Mutual information can also be used to model other types of information, such as the number of hits returned by search engines [111].

Mutual Information Based on Dependency Relations from Minipar

We used the method proposed by Lin [65] to compute the mutual information between word pairs having dependency relations. An occurrence of a dependency relation triple *(head, type, modifier)* can be treated as a conjunction of three events:

- H: *head* is a randomly selected word
- T: *type* is a randomly selected type of dependency relation
- M: *modifier* is a randomly selected word

According to the definition of mutual information, the mutual information of *(head, type, modifier)* can be derived from Equation 5.3, assuming that the three events are independent of one another.

$$I(head, type, modifier) = \log \frac{p(H, T, M)}{p(H)p(T)p(M)} \tag{5.3}$$

However, as indicated by Lin [66], this independent assumption is not suitable for the triple, because the choice of *head* and *modifier* is dependent upon the parts-of-speech embedded in *type*. Therefore, he assumed that H and M are conditionally independent of each other given T. With this new assumption, the mutual information of the triple is estimated with Equation 5.4.

$$I(head, type, modifier) = \log \frac{p(H, T, M)}{p(H|T)p(T)p(M|T)} \tag{5.4}$$

Maximum likelihood estimation (MLE) could be used to estimate the probabilities in Equation 5.4. Let $|(head, type, modifier)|$ stand for the frequency of the triple *(head, type, modifier)*. When any of the elements in the triple is replaced by a wild card *, it stands for the sum of frequency over all possible values of this element. For example,

- $|(h, t, *)|$: the total frequency of triples with *head* h and dependency relation *type* t
- $|(h, *, *)|$: the total frequency of triples with *head* h
- $|(*, *, *)|$: the total frequency of triples in the entire corpus

Using the above annotations, the MLE probabilities can be derived from Equation 5.5, and the mutual information can then be derived as shown in Equation 5.6:

$$\begin{aligned}
P_{MLE}(H, T, M) &= \frac{|(head, type, modifier)|}{|*, *, *|} \\
P_{MLE}(H|T) &= \frac{|(head, type, *)|}{|(*, type, *)|} \\
P_{MLE}(T) &= \frac{|(*, type, *)|}{|(*, *, *)|} \\
P_{MLE}(M|T) &= \frac{|(*, type, modifier)|}{|(*, type, *)|}
\end{aligned} \tag{5.5}$$

$$
\begin{aligned}
I(head, type, modifier) &= \log \frac{p(H,T,M)}{p(H|T)p(T)p(M|T)} \\
&= \log \frac{p_{MLE}(H,T,M)}{p_{MLE}(H|T)p_{MLE}(T)p_{MLE}(M|T)} \\
&= \log \frac{\frac{|(head,type,modifier)|}{|*,*,*|}}{\frac{|(head,type,*)|}{|(*,type,*)|} \times \frac{|(*,type,*)|}{|(*,*,*)|} \times \frac{|(*,type,modifier)|}{|(*,type,*)|}} \\
&= \log \frac{|(head,type,modifier)| \times |\,(*,type,*)|}{|(head,type,*)| \times |\,(*,type,modifier)|}
\end{aligned} \tag{5.6}
$$

A dependency relation database, serving as the background knowledge, was used to estimate the mutual information of a triple. The database was constructed based on the parsing results of a 125-million-word newspaper corpus using Minipar [79].

Features

Six features were extracted for each word w from the output of Minipar. Three features concern w itself, and the other three are based on the dependency relations in which w is involved:

- Word or punctuation: punctuation is always separated from surrounding words in a Minipar output. This feature is used to differentiate punctuation marks from words and is thus a binary variable. It is useful to have this feature because punctuation marks are not rare in dictation corpora.

- Phrase length: phrase structure of a sentence is available from the parsing output of Minipar. For instance, *come from* in Figure 5.2 is a phrase. This feature is measured by the total number of words in the phrase that contains w.

- Phrase head: if w belongs to a phrase, this binary variable indicates whether w is the head word of a phrase or not. For example, *come* is the phrase head of *come from*.

- Number of triples missing from the database: although the dependency database used to estimate the mutual information is not small, its coverage is still limited. Therefore, a triple may or may not be seen in the dependency relation database. This feature counts the number of unseen triples that contain w as either a head or a modifier.

- Number of triples in the database: it represents the number of seen triples that contain w as either a head or a modifier.

- Maximum mutual information of triples in the database: if w belongs to some seen triples, the value of this feature is the maximum of all mutual information values for triples to which w belongs. The value indicates the strongest relation between w and other related words. If the maximum mutual information value is a negative value, the value of this feature is set to 0 because negative mutual information is unreliable without a huge corpus [22]. If w does not belong to any seen triple, the value of this feature is set to 0.

5.3 Experiment Results

The experiment setup was the same as that used in Chapter 4. In the current experiment, the newly introduced syntactic features were used in conjunction with the original non-linguistic features introduced in Chapter 4.

Both the individual and complementary effects of every feature group were evaluated. For convenience, we used *Link* and *Dep* to represent features extracted from the output of Link Grammar and Minipar respectively. In addition, we used *SYN* to represent all syntactic features, which are the combination of *Link* and *Dep*. *CSLink*, *CSDep*, and *CSSYN* represent the combination of non-linguistic features and *Link*, *Dep*, or *SYN* respectively.

5.3.1 the Study3 corpus

Classification Error Rate

Table 5.2 reports experiment result of CERs achieved under selected settings on the study3 corpus, in which Table 5.2(a) reports descriptive statistics of CER and Table 5.2(b) shows relative improvements of CERs achieved by different feature groups over CS.

(a) Descriptive statistics of CERs achieved by different feature groups

	CS	Link	Dep	SYN	CSLink	CSDep	CSSYN
Mean (%)	12.57	15.39	15.21	15.36	11.90	12.08	11.54
Std.	6.36	9.16	9.17	9.23	5.74	6.12	5.71

(b) Relative improvement (%) of CERs achieved by different feature groups over CS

	CSLink	CSDep	CSSYN
CS	5.33	3.90	8.19

Table 5.2: Classification error rates of confidence measures on the Study3 corpus when syntactic features were used

When using features from Link Grammar alone, the average CER was 15.39%, which was higher than

(a) Descriptive statistics of Fs, PREs, and RECs achieved by different feature groups

		CS	Link	Dep	SYN	CSLink	CSDep	CSSYN
F	Mean (%)	38.08	2.24	0.00	5.81	44.42	42.87	46.52
	Std.	15.07	3.39	0.00	6.93	15.85	12.72	16.62
REC	Mean (%)	27.96	1.22	0.00	3.38	34.81	32.60	37.11
	Std.	12.76	1.87	0.00	4.19	15.28	12.51	16.53
PRE	Mean (%)	62.91	16.94	0.00	31.72	68.28	68.68	69.56
	Std.	17.97	22.49	0.00	37.00	15.16	15.32	15.18

(b) Relative improvement (%) of Fs, PREs, and RECs achieved by different feature groups over CS

		CSLink	CSDep	CSSYN
F	CS	16.65	12.58	22.16
REC	CS	24.50	16.60	32.73
PRE	CS	8.54	9.17	10.57

Table 5.3: F measure, REC, PRE of confidence measures on the Study3 corpus when syntactic features were used

the baseline (15.21%). Using features from Minipar alone did not change CER from baseline. In addition, the combination of both types of features did not improve the CER over the baseline. Therefore, syntactic features alone are not useful for the improvement of CER.

The combination of Link Grammar features with CS features and the combination of Minipar features with CS features, however, obtained 5.33% and 3.90% relative improvement of CER over that obtained using CS features alone. When both groups of syntactic features were combined with CS features, an 8.19% relative improvement on CER was achieved over that achieved using CS features alone.

F, REC, and PRE

Table 5.3 reports experiment results of F measure, PRE, and REC achieved under selected settings on the study3 corpus, in which Table 5.3(a) reports descriptive statistics and Table 5.3(b) shows the relative improvements achieved by different feature groups over CS.

It is shown from Table 5.3(a) that using features from Link Grammar alone, features from Minipar alone, and the combination of the two produced low F values. Thus, syntactic features alone are not effective for error detection. This finding is consistent with that observed for CER.

When combined with CS features, features from Link Grammar, features from Minipar, and their combination improved the F measure achieved using CS features alone by 16.65%, 12.58%, and 22.16%, respectively. Further analysis of F measure can be made by looking into REC and PRE.

When combined with CS features, features from Link Grammar, features from Minipar, and both types of features together improved the REC obtained using CS features alone by 24.50%, 16.60%, and 32.73%

	CER	F	REC	PRE
$F(2, 22)$	1.714	1.920	3.232	0.311
p	0.203 [a]	0.170 [a]	0.059 [a]	0.624 [b]

[a] Sphericity assumed
[b] Greenhouse-Geisser adjustment

Table 5.4: Repeat measure analyses results on feature combination (CSLink, CSDep, and CSSYN) for CER, F, REC, and PRE on the Study3 corpus

respectively, which are reported in Table 5.3(b).

Table 5.3(b) also reveals that the combination of CS features and features from Link Grammar or from Minipar produced 8.54% and 9.17% relative improvement over PRE achieved using CS features alone, respectively. The combination of both types of syntactic features and CS features achieved a 10.57% relative improvement over the PRE obtained using CS features alone.

Based on the findings concerning REC and PRE, it is evident that the improvement on F measure was mainly attributable to the improvement on REC and that the improvement on REC did not adversely affect PRE.

Do different syntactic features provide complementary information?

When CS features were used, combining syntactic features from both Link Grammar and Minipar led to better performance on all four performance measures (CER, F measure, REC, and PRE) than did combining either type of the syntactic features alone. To judge whether these improvements are significant or not, four separate repeated measure analyses were conducted by treating the feature combination as an independent variable. Each of the four performance measures served as the dependent variable for one repeated measure analysis. *CSLink*, *CSDep*, and *CSSYN* were the three levels of the feature combination variable. Table 5.4 shows the main effect of feature combination on the four measures. The results revealed that there were no significant differences between *CSLink*, *CSDep*, and *CSSYN* on CER, F measure, REC, and PRE ($p = n.s.$).

5.3.2 the WSJ corpus

Classification Error Rate

Table 5.5 reports experiment result of CERs achieved under selected settings on the WSJ corpus, in which Table 5.5(a) reports descriptive statistics and Table 5.5(b) shows the relative improvement of CERs achieved by different feature groups over CS.

(a) Descriptive statistics of CERs achieved by different feature groups

	CS	Link	Dep	SYN	CSLink	CSDep	CSSYN
Mean (%)	11.41	11.77	11.77	11.77	10.92	10.97	10.75
Std.	4.38	4.82	4.82	4.82	4.00	4.13	3.87

(b) Relative improvement (%) of CERs achieved by different feature groups over CS

	CSLink	CSDep	CSSYN
CS	4.29	3.86	5.78

Table 5.5: Classification error rates of confidence measures on the WSJ corpus when syntactic features were used

(a) Descriptive statistics of Fs, PREs, and RECs achieved by different feature groups

		CS	Link	Dep	SYN	CSLink	CSDep	CSSYN
F	Mean (%)	16.35	0.00	0.00	0.00	23.02	19.60	24.19
	Std.	7.96	0.00	0.00	0.00	10.27	7.87	10.70
REC	Mean (%)	9.63	0.00	0.00	0.00	14.53	11.72	15.64
	Std.	4.92	0.00	0.00	0.00	7.87	5.20	9.33
PRE	Mean (%)	58.20	0.00	0.00	0.00	67.61	67.24	69.45
	Std.	22.38	0.00	0.00	0.00	21.37	15.83	17.89

(b) Relative improvement (%) of Fs, PREs, and RECs achieved by different feature groups over CS

		CSLink	CSDep	CSSYN
F	CS	40.80	19.88	47.95
REC	CS	50.88	21.70	62.41
PRE	CS	16.17	15.53	19.33

Table 5.6: F measure, REC, PRE of confidence measures on the WSJ corpus when syntactic features were used

When using syntactic features from both Link Grammar and Minipar and from each separately, there were no changes in CER from the baseline (11.77%). Therefore, syntactic features alone had no effect on CER.

Combining features from Link Grammar with CS features and combining features from Minipar with CS features improved the CER obtained using CS features alone by 4.29% and 3.86%, respectively. The combination of both types of syntactic features with CS features improved the CER obtained using CS features alone by 5.78%.

F, REC, and PRE

Table 5.6 reports experiment results of F measure, PRE, and REC achieved under selected settings on the WSJ corpus, in which Table 5.6(a) reports descriptive statistics and Table 5.6(b) shows relative improvement achieved by different feature groups over CS.

As shown in Table 5.6(a), features from Link Grammar, features from Minipar, and the combination of

	CER	F	REC	PRE
$F(2, 18)$	0.458	2.284	2.511	0.132
p	0.540 [b]	0.131 [a]	0.109 [a]	0.739 [b]

[a] Sphericity assumed
[b] Greenhouse-Geisser adjustment

Table 5.7: Repeat measure analyses results on feature combination (CSLink, CSDep, and CSSYN) for CER, F, REC, and PRE on the WSJ corpus

the two achieved F values of zero, which is consistent with the lack of change in CER observed from Table 5.5(a). Thus syntactic features alone are not helpful for error detection.

The F measure achieved through the use of CS features alone was improved by 40.80% with the combination of features from Link Grammar, by 19.88% with the combination of features from Minipar, and by 47.95% with the combination of both types of syntactic features.

The analysis of REC revealed similar patterns to those observed for F measure. As shown in Table 5.6(b), incorporation of features from Link Grammar along with CS features improved the REC obtained using CS features alone by 50.88%. Incorporation of features from Minipar along with CS features improved the REC obtained using CS features alone by 21.70%. Incorporation of features from both Minipar and Link Grammar along with CS features improved the REC obtained using CS features alone by 62.41%.

Incorporation of CS features along with features from Link Grammar improved the PRE obtained using CS features alone by 16.17%. Incorporation of CS features along with features from Minipar improved the PRE obtained using CS features alone by 15.53%. Incorporation of CS features along with features from both Minipar and Link Grammar improved the PRE obtained using CS features alone by 19.33%. Both REC and PRE improved with the introduction of syntactic features, and the improvement on REC was bigger than that on PRE.

Do different syntactic features provide complementary information?

Similar to the finding on the Study3 corpus, when CS features were used, combining syntactic features from both Link Grammar and Minipar led to better performance on all four performance measures (CER, F, REC, and PRE) than did combining either type of the syntactic features alone. Same set of repeated measure analyses were conducted to judge if performance changes caused by feature combination were significant or not. Table 5.7 shows the analysis results, which evidenced that the improvements on the four measures were not significant.

5.3.3 Summary

Based on the experiments using two types of syntactic features on two dictation corpora, we found that using syntactic features alone were not able to improve error detection performance over the baseline. When combined with CS features, however, syntactic features improved the error detection performance on different aspects over those by CS features alone. The combination of both types of syntactic features did not significantly outperform the individual type in error detection, but there were encouraging trends. This suggests that the two types of syntactic features provide a certain degree of redundant information for the task of error detection. The combination of both types of syntactic features and CS features can reduce CER from 12.56%, which is the CER yielded by CS features, to 11.54% with an 8.19% relative improvement on the study3 corpus; the combination of both types of syntactic features and CS features can reduce CER from 11.41%, which is the CER by CS features, to 10.75% with a 5.78% relative improvement on the WSJ corpus. With increasing recall and non-decreasing precision, the syntactic features can improve the system's ability to accurately identify errors.

All the error detection results in this section were generated using SVM models. SVM is known for its stable performance across a variety of applications. However, results of SVM are difficult to interpret, which is one of the weaknesses of the statistical classifier. To gain additional human comprehensible insights into the efficacy of different types of syntactic features, a rule-based learning method was developed and is discussed in the next section.

5.4 An In-Depth Analysis of Syntactic Features

Transformation-based learning is a rule-based learning method, in which rules are automatically learned from training data. TBL has been successfully applied to many natural language applications such as part-of-speech tagging [7], error correction [72] and error detection [104]. The rules learned by TBL show good interpretability as well as good performance. Therefore, we selected TBL to derive error patterns.

5.4.1 Transformation-Based Learning

Generally speaking, three prerequisites are required for use of TBL: an initial state annotator, a set of possible transformations, and an objective function for choosing the best transformations.

Prior to learning, the initial state annotator adds annotations to the training data. Then the following steps

are executed iteratively until no improvement can be achieved: (i) try each possible transformation on the training data, (ii) score each transformation with the objective function and choose the one with the highest score, and (iii) apply the selected transformation to update the training data and append it to the learned transformation list.

In this research, the initial state annotator simply initialized all the words as correct words. The objective function is the CER. Transformations are keys to TBL and are represented as different combinations of features.

5.4.2 Features

Due to the different natures of a statistical classifier and a rule-based classifier, different methods were used by them in feature selection and representation. For each output word, we selected two kinds of linguistic features for TBL – lexical features that do not rely on any syntactic analysis and syntactic features that are extracted from syntactic analysis output. Link Grammar is used to generate syntactic features because features from Link Grammar performed better than Minipar in the experiment described in Section 5.3 in which SVM was used as the classifier.

Lexical Features

Lexical features are features that can be extracted directly from words and sentences without any syntactic analysis. For each word w, the following features are extracted:

- word: w itself
- pos: part-of-speech tag from Brill's tagger [7]
- syllables: number of syllables in w
- position: position of w in the sentence: beginning, middle, or end

Syntactic Features

Syntactic features were selected based on the following hypotheses: 1) a word without any link is a good indicator of the occurrence of an error, and 2) a word with links may still be an error, and its correctness may affect the correctness of words linked to it, especially those words connected via the shortest links or

the closest connections. For each word w, therefore, the following features were extracted from the output of Link Grammar:

- haslink: indication if w has left links, right links, or no link
- llinkto: the word to which w links via the shortest left link
- rlinkto: the word to which w links via the shortest right link

Other Features

In addition to the linguistic features introduced above, two other features were selected:

- word confidence score (CS): Unlike in the previous study which used several confidence score-related features in addition to the CS, the only confidence score-related feature selected in this study was the confidence score (CS) itself. There are several reasons for this choice. First, transformation-based learning usually takes categorical variables as input and is not good at dealing with too many numerical variables. Second, compared with other CS-based features such as DCS and CCS, RCS underwent the second best performance in [120], and CS is the primary factor in RCS because range is derived from CS. Though CCS performed the best of all the CS-based features, its 12 features may cause unnecessary computation complexity.
- prediction label (label): The label is the indicator of the correctness of the word. Initially, the label for every word is marked as *correct* but can be updated to be marked as either *SR error* or *correct* according to the result produced in each learning iteration. This feature enables the propagation of the effect of preceding rules.

5.4.3 Transformation Templates

Pre-defined transformation templates, which specify rules acceptable for use, play a vital role in TBL. A transformation template is defined in the following format:

change the word label of a word w from X to Y, if condition C is satisfied

where, X and Y take binary values: 1 (correct recognition) or -1 (error). Each condition C is the conjunction of sub-conditions in form of $f\ op\ v$, where f represents a feature, v denotes a possible value of f, and op represents a possible operation, such as $<$, $>$ and $=$.

Category	Group	Example
Word Alone	CS	$cs(w_i) < c_i$
	L	$position(w_i) = t_i$ & $syllables(w_i) = s_i$
	LCS	$cs(w_i) < c_i$ & $pos(w_i) = p_i$
Local Context	Local	$position(w_i) = t_i$ & $label(w_{i-1}) = l_{i-1}$ & $word(w_i) = d_i$
	CSLocal	$cs(w_i) < c_i$ & $position(w_i) = t_i$ & $label(w_{i-1}) = l_{i-1}$ & $label(w_{i+1}) = l_{i+1}$
Sentence Context	Long	$position(w_i) = t_i$ & $lrHaslink(w_i) = h_i$ & $haslink(w_i) = hl_i$
	CSLong	$cs(w_i) < c_i$ & $position(w_i) = t_i$ & $llinkLabel(w_i) = ll_i$ & $pos(w_i) = p_i$

Table 5.8: Condition categories and examples

As shown in Table 5.8, conditions were classified into three categories based on the context from which features were extracted: word alone, local context, and sentence context. The scope of the context increased incrementally. The three categories were further split into seven groups according to the types of features being used.

- L: the correctness of w depends solely on itself. Conditions only involve lexical features of w.
- Local: the correctness of w depends not only on itself but also on its surrounding words. This type of conditions involves lexical features of surrounding words as well as those of w. Furthermore, prediction labels of surrounding words are also employed as a feature to capture the possible effect of their correctness.
- Long: the scope of conditions for predicting the correctness of w is expanded to include syntactic features. This type of condition consists of syntactic features of w and its surrounding words, the features in Local, and lexical features and prediction labels of words that have the shortest links to w.
- CS: this group of conditions only involves confidence scores of w.
- LCS, CSLocal, CSLong: these three groups of conditions are generated by combining the features from L, Local, and Long with the confidence scores of w, respectively.

In Table 5.8, an example is provided for each group of conditions. The features included in the examples are either a single basic feature such as CS and word or a combination of basic features such as $lrHashlink$ and $llinkLabel$. $lrHashlink$ represents whether the preceding word and the following word have links, and $llinkLabel$ represents the predicted label of the word to which w has the shortest left link. c_i, t_i, s_i, p_i, l_i, d_i, h_i, hl_i, and ll_i represent possible values of the corresponding features.

5.4.4 Experiment Results

Experiments were conducted with increasing usage of context information. Initially transformation rules under word alone conditions were used. The scope of the context was then enlarged by incrementally including transformation rules under local context and sentence context conditions. These experiment settings have two benefits: (i) they enable us to identify the impact of individual length of context on error detection, and (ii) they reveal the importance of enriching contextual information to error detection.

A prolog-based TBL tool, μ-TBL [54] [3] was used in this study. Two cut-off thresholds, namely *net positive effect* and *ratio of the positive effect*, were used to select transformations with strong positive effect. The *positive effect* (PE) of a transformation is defined as the number of words whose labels are correctly updated by applying the transformation rule. The *negative effect* (NE) of a transformation is defined as the number of words that are mistakenly updated. The *net positive effect* was defined as ($PE - NE$), and the *ratio of positive effect* was defined as ($PE/(PE + NE)$). In this experiment, the threshold for *net positive effect* was set to five to ensure that sufficient evidence was observed, and the threshold for the *ratio of the positive effect* was set to 0.5 to guarantee that a selected transformation had a positive effect.

The study3 corpus was used in this experiment. We used three-fold cross-validation in evaluation. The corpus was divided in units of sentences. The cross-validation was conducted nine times, and average performance is reported in Table 5.9. Given that a different cross-validation was used on the study3 corpus in this section compared to that used in section 4.4, the baseline CER reported here is different from that reported in section 4.4. The labels for rule combinations in the first column of Table 5.9 were created by concatenating two or more identifiers for groups of conditions. For each rule combination, legitimate rules are those conformed to any combinations of related group identifiers from Table 5.8. For example, L-CS-Local-Long accepts rules conforming to any of following groups: L, CS, Local, Long, LCS, CSLocal and CSLong.

As shown in Table 5.9, among the rule combinations which used linguistic features only, L-Local-Long achieved the best performance in terms of both CER and F measure, with a CER improvement of 4.85% over the baseline. However, the performance of L-Local-Long was inferior to that of CS in both CER and F measure. Therefore, linguistic features alone were not as effective as confidence score features in error detection.

When transformation rules combining both linguistic features and CS were applied, better performance was achieved than that achieved by using transformation rules including CS alone. The CER achieved by

[3] Available from `http://www.ling.gu.se/~lager/mutbl.html`

Combination	Average Performance				
	CER (%)	PRE (%)	REC (%)	F (%)	# of rules
Baseline	15.66	-	-	-	-
L	15.55	61.84	2.04	3.88	3
L-Local	15.58	60.88	2.19	4.17	4
L-Local-Long	14.90	61.67	13.83	22.37	8
CS	14.64	61.03	21.98	31.50	1
L-CS	13.97	61.48	31.60	41.38	8
L-CS-Local	13.81	61.28	35.52	44.50	14
L-CS-Local-Long	12.84	65.50	38.59	48.39	14

Table 5.9: Performance of transformation rule combinations

L-CS was 4.58% higher than that achieved by CS; the F measure achieved by L-CS was 31.37% higher than that achieved by CS. L-CS-Local achieved a marginally lower CER than L-CS and an F measure 7.54% higher than that achieved by L-CS.

The best performance was achieved by L-CS-Local-Long. Specifically, the CER achieved by L-CS-Local-Long was 12.30% higher than that achieved by CS and 7.02% higher than that achieved by L-CS-Local. In addition, the F measure achieved by L-CS-Local-Long was 53.62% higher than that achieved by CS and 8.74% higher than that achieved by L-CS-Local. Therefore, enlarging the scope of context for feature selection can lead to improved performance in error detection.

In Table 5.9, it is noteworthy that improvement in F measure results from improvement in recall and relatively stable precision. Recall achieved by L-CS was 43.77% higher than that achieved by CS. Recall achieved by L-CS-Local-Long was 75.57% higher than that achieved by CS. These improvements confirm the findings from the study reported in Section 5.3 that linguistic features can help finding more errors. Additionally, precision achieved by L-CS-Local-Long was 7.32% higher than that achieved by CS, which suggests that linguistic features can also help finding errors more precisely.

The average number of learned rules is listed in the last column of Table 5.9. When the number of user pre-defined candidate rules increased, the number of learned rules only increased moderately. For example, the maximum number of rules was 14 and was generated by L-CS-Local-Long and L-CS-Local.

Figure 5.3 illustrates the relationship between CER and the number of learned rules in one run of the L-CS-Local-Long setting. The three lines in the figure correspond to the three folds in a cross-validation. Because we did not use the number of rules as the cutting-off threshold, the training could stop once selected cutting-off thresholds were satisfied. As a result, different folds resulted in different numbers of learned rules.

Following application of the top few rules, CERs dropped significantly. This decreasing trend in CER

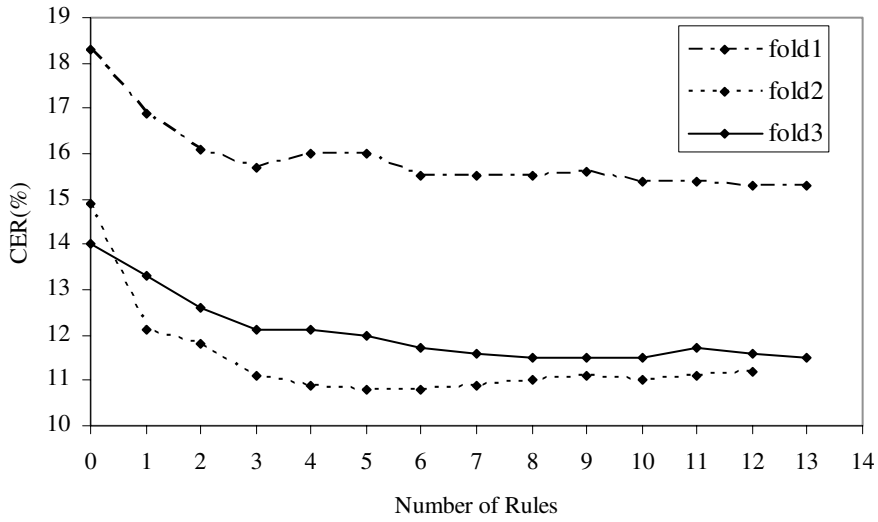


Figure 5.3: Relationship between CER and the number of rules applied

gradually tapered off as additional rules were applied. Both fold 1 and fold 3 reached the lowest CER after the last rule was applied, and fold 2 reached the lowest CER when less than half of the rules were applied. Thus, the top ranked rules were the most beneficial to the error detection.

One advantage of TBL is that learning results are easily interpreted. The following are the top six rules learned in fold 3, as shown in Figure 5.3.

Mark a word as an error, if it satisfies the following conditions:

- its confidence score is less than 0; it is in the middle of a sentence; it is a null-link word.
- its confidence score is less than -5; it is the middle of a sentence; it has links to preceding words.
- its confidence score is less than 0; it is the first word of a sentence; it is a null-link word.
- its confidence score is less than 2; it is in the middle of a sentence; it has 1 syllable; the word following it also has 1 syllabus and is an error.
- its confidence score is less than -1; both its preceding and following words are errors.

Mark a word as a correct word, if it satisfies the following conditions:

- its confidence score is greater than -1; both its preceding and following words are correct words.

All of the above six rules contain word confidence score as a feature. Rule 1 and Rule 3 suggest that null-link words are good indicators of errors, which supports our hypothesis. Rule 2 shows that a word with

a very low confidence score may be an error even if it is part of a linkage of the sentence. Rule 4 shows that consecutive short words are possible errors when they are in the middle of a sentence and their CS are not high enough. Rule 5 indicates that a word with a low confidence score may be an error if its surrounding words are errors. Rule 6 labels a word back to a correct word and may correct mistakes made by previous rules.

5.4.5 Summary

The goal of applying a rule-based method in this experiment was not to compare it to a statistical model but to provide user-friendly explanations of why and how the linguistic features are useful. The results of the rule-based method confirmed the findings of our SVM-based methods. Although linguistic features alone were not as useful as the confidence scores, when combined with word confidence scores they were effective in further improving error detection performance. Moreover, error detection performance can be improved by enlarging the scope of context. The best performance was achieved when sentence context was used. Specifically, enlarging the context of linguistic features improved error detection performance by revealing more errors while maintaining or even improving precision.

Chapter 6

CONFIDENCE MEASURE INCORPORATING WORDNET BASED SEMANTIC KNOWLEDGE

Because speech recognition systems try to generate output that maps original audio speech, a recognition error is always phonetically similar to its corresponding spoken word. In contrast, a recognition error may be semantically unrelated to its corresponding spoken word and hence may be semantically out of context. Language models embedded in an SR system are usually used for checking the semantic consistency of output words. Due to performance constraints, n-gram language models are widely used by SR systems to carry out consistency analysis. The most popular n-gram language models restrict the assessment of semantic compatibility in a small local context, such as two or three words. Therefore, analysis of semantic consistency of words within a larger context is a possible tool for error detection. To quantify semantic consistency, we can apply semantic relatedness of a word in its local context.

In this chapter, we investigate how error detection performance may be improved through analysis of semantic relatedness within an expanded context. Several semantic relatedness measures based on WordNet are first introduced. How to extract semantic features for a word based on its semantic relatedness to the expanded context then described. Finally, the usefulness of the extracted semantic features for error detection is evaluated.

6.1 Semantic Relatedness based on WordNet

There are two well known groups of approaches to computing the semantic relatedness of two words, and these two groups differ in source of background knowledge utilized:

- dictionary/thesaurus-based: semantic relatedness measures are constructed using the structured information embedded in a dictionary or thesaurus, such as WordNet and Roget's thesaurus.
- Corpus-based: semantic relatedness measures are derived by statistically analyzing word associations (e.g. co-occurrence of words) extracted from text corpora.

In this chapter, we focus on using semantic relatedness measures based on the dictionary/thesaurus, especially WordNet, to evaluate the semantic consistency of a word to its context. Dictionary/thesaurus-based measures are generally topic independent.

6.1.1 WordNet

WordNet [1] is an electronic lexical database of English words [28]. The basic unit of WordNet is not word but synset which includes a group word senses representing the same concept. In English, some words are polysemous with more than one sense. Each sense of a word corresponds to a synset. Like a regular dictionary, WordNet offers both definition and sample sentences for each synset. Synsets are inter-linked by semantic relations such as *hypernymy* and *meronymy* as well as lexical relations such as *antonym*. WordNet only covers open words, that is, nouns, verbs, adjectives, and adverbs. Each grammatical category has its own synset network, because synsets in different grammatical categories may have different semantic relations. In addition, there are some relations, such as *attribute* and *pertain*, which can across different networks to link synsets belongs to different grammatical categories.

Among the available dictionaries and thesauri, WordNet is the most popularly used lexical source due to its wide coverage of words and semantic relations and its programming APIs. Another popular thesaurus is Roget's thesaurus, but few measures such as Jarmasz and Szpakowicz's measure [46] have been proposed based on Roget's thesaurus due to its limited availability. In the rest of this section, we will introduce several semantic relatedness measures that use different knowledge sources provided by WordNet. Given the fact that semantic relations exist between concepts (synsets) not words, the semantic relatedness measures were designed to determine the semantic relatedness between concepts, or synsets, instead of between words.

[1] Available from `http://wordnet.princeton.edu/obtain`

6.1.2 Semantic Relatedness Measures based on WordNet Structure

In WordNet, synsets are semantically connected and are organized as a network. Therefore, the network structure of synsets is an obvious information source to measure the semantic relatedness between synsets. Edge counting is an obvious measure of semantic relatedness of two synsets by counting the edges connecting the two synsets. However, edge counting makes a strong but unrealistic assumption that all edges are equidistant from one another. Some variations on edge counting have been proposed to alleviate this problem as follows:

Leacock and Chodorow [61] measured the semantic similarity of two synsets using normalized path length. Only *hypernym-hyponym* relations between synsets were used when computing path length, which simplified WordNet as an IS-A hierarchy. For two concepts c_1 and c_2, the similarity between them was computed by Equation 6.1, where *Pathlength* is the length of their shortest path, and D is the maximum depth of the hierarchy.

$$sim(c_1, c_2) = -\log \frac{Pathlength(c_1, c_2)}{2 \times D} \tag{6.1}$$

Sussna [109] considered two factors when computing the semantic distance between two concepts: *depth-relative scaling* that considers the fact that low-level siblings are more closely related to one another than high-level siblings, and *type-specific fanout* that considers the number of the same type of relations leaving the node. Moreover, different types of relations were assigned different weights. The semantic distance between two concepts was computed by summing the weights of the edges in the shortest path between them. This semantic measure was applied to a word sense disambiguation task, and the results showed that (i) *depth-relative scaling* played an important role, and that (ii) both hierarchical and non-hierarchical relations helped.

6.1.3 Semantic Relatedness Measures Incorporating WordNet Structure and Corpus

WordNet is a static knowledge source. To make it adaptable, WordNet can be augmented by representing each concept c with the probability ($p(c)$) of observing an instance of it in a background corpus. Different measures have been proposed based on such augmented WordNet, and regard WordNet as an IS-A hierarchy. Given

two concepts c_1 and c_2, information content of their lowest subsumed concept were used by these measures which differed in whether information content of concepts themselves had been used or how information content was combined.

Resnik [88] proposed to measure the similarity between two concepts using the information content of their lowest subsumed concept $S(c_1, c_2)$, as shown in Equation 6.2.

$$sim(c_1, c_2) = -\log P(S(c_1, c_2)) \tag{6.2}$$

Jiang and Conrath [47] pointed out that Resnik's measure only takes into account the subsumed concept and does not differentiate concept pairs with the same lowest subsumed concept. They tackled this problem by assigning strength to each link and then adding the link strengths (LS) along the shortest path between two concepts to get the semantic distance. The *link strength* is defined as the information content of the conditional probability of a concept (c) given its parent (p), which in fact is the difference between the information content of c and p. Equation 6.3 defined the semantic distance between two concepts, which is the linear combination of the information content of both concepts themselves and their lowest subsumed concept.

$$\begin{aligned} Dist(c_1, c_2) &= \sum_{c \in \{path(c_1,c_2) - S(c_1,c_2)\}} LS(c, parent(p)) \\ &= (-\log P(c_1)) + (-\log P(c_2)) - 2 \times (-\log P(S(c_1, c_2))) \\ &= -(\log P(c_1) + \log P(c_2) - 2 \times \log P(S(c_1, c_2))) \end{aligned} \tag{6.3}$$

Lin [67] proposed an information-theoretic based measure that used the same information as [47] but a different combination method. The similarity between two concepts is defined as the rate between the information describing the commonality of the concepts and the information fully describing the concepts. The commonality of c_1 and c_2 could be described by the information content of their lowest subsumed concept; and the sum of their own information content can fully describe them given that the two concepts are independent. Thus, the semantic similarity of two concepts is defined in Equation 6.4:

$$sim(c_1, c_2) = \frac{-2 \times \log P(S(c_1, c_2))}{-(\log P(c_1) + \log P(c_2))} = \frac{2 \times \log P(S(c_1, c_2))}{\log P(c_1) + \log P(c_2)} \tag{6.4}$$

6.1.4 Semantic Relatedness Measures Based on Gloss

Word definitions (glosses) provided in the dictionary are another source of information to assess the semantic relatedness of words. Lesk [62] proposed to use the overlap of glosses to judge word senses. In his method, the sense of a target word is determined by comparing the glosses of its various senses to those of its surrounding words. A sense is assigned to the target word when the gloss of the sense shares the most common words with the glosses of its surrounding words.

Banerjee and Pedersen [2] extended Lesk's gloss overlapping method by overcoming several limitations. In stead of only considering glosses of the target concept, they also compared the glosses of the concepts that are related to the target concepts through certain semantic relations, such as hypernym and also-see, to address the insufficient information provided by short glosses. In addition, they differentiated single words and phrases and assigned common phrases a weight larger than the sum of weights of words in the phrase. The overlapping score between two glosses was then computed by summing the weights of their longest overlaps. The relatedness of two concepts could be further computed as the sum of the overlapping scores between all allowable concept pairs, which was confined by $RELPAIRS$, as shown in Equation 6.5,

$$relatedness(c_1, c_2) = \sum score(R_1(c_1), R_2(c_2)) \quad \forall (R_1, R_2) \in RELPAIRS \tag{6.5}$$

Patwardhan and Pedersen [82] proposed a measure based on second order co-occurrence vectors [98] to overcome the exact match problem present in the extended gloss overlap method. For a word in the gloss, instead of using the word itself, they used a co-occurrence vector extracted from a pre-computed word space to represent the word. The pre-computed word space is a sparse matrix, in which each cell holds the frequency of co-occurrence of two words in a certain distance. The gloss of a concept is then represented by the co-occurrence vectors of words in the gloss. Similar to the extended gloss overlap method, the gloss of a concept c is also augmented by adding glosses from the concepts that have certain direct relationships with c. The relatedness of the two concepts is the cosine similarity between the augmented gloss vectors.

Semantic relatedness measures have been used in many natural language applications. Budanitsky and Hirst [8] applies several semantic relatedness measures to the malapropism detection and found that Jiang and Conrath's measure achieved the best detection performance among five measures incorporating WordNet structure. McCarthy et al. [73] compared semantic relatedness measures on the task of finding the predominant noun senses and discovered that Jiang and Conrath's measure and Banerjee and Pedersen's measure were the best measures among six measures, and the same findings were also observed by Patwardhan and Pedersen [82] in their task for word sense disambiguation. Mihalcea et al. [74] evaluated text similarity based on semantic relatedness measures on the task of detecting paraphrase, and observed that different semantic relatedness measures achieved similar results.

With those findings by other research, Jiang and Conrath's measure and Banerjee and Pedersen's measure were selected to be used in this chapter. Moreover, semantic relatedness measures belonging to the same categories as them were also used. To simplify the notation, we used *res*, *jcn*, *lin*, *E-lesk*, *vector* to refer to the measures proposed by Resnik, Jiang and Conrath, Lin, Banerjee and Pedersen, and Patwardhan and Pedersen respectively.

6.2 Features Extracted Based on Semantic Measures

Two groups of features were extracted for each word for error detection. One group of features was used to categorize words, and the other group of features was used to characterize the relatedness of a word to its contextual words.

6.2.1 Categorization Features

Three features were included in this group:

part-of-speech

Given that only content words can bear meaning, WordNet therefore only includes content words. However, SR output includes both content and function words, and this feature thus is used to represent the part-of-speech of a word. When only nouns are considered, this feature is represented by a binary variable the value of which indicates if a word is a noun. When nouns, verbs and adjectives are considered, this feature is represented by three binary variables, each corresponding to one grammatical category. The Stanford log-

linear part-of-speech tagger [2] [110] was used to parse the SR output.

stop word

The stop word feature is used to determine if a content word belongs to a pre-defined stop word list. Because not all content words have specific meanings, we used a stop word list to eliminate common words.

included in WordNet

This feature indicates if a content word is included in WordNet and it is useful for two reasons. First, because WordNet does not cover all words, it is useful in identifying the words covered. Second, existing part-of-speech taggers are not error free for grammatically correct input, and the ever present SR errors in SR output may worsen the tagging results. Therefore, a part-of-speech tagger may introduce possible errors by assigning wrong parts-of-speech to words, which makes words not included in the WordNet.

6.2.2 Semantic Relatedness Features

Word Semantic Relatedness

The semantic relatedness measures introduced in section 6.1 compute the semantic relatedness between two concepts instead of two words. In WordNet 3.0, 17.32% of words are polysemous. Given that the SR output is presented in words instead of concepts, we need to further measure the relatedness between two words. Given two words, w_1 and w_2, where each has a set of concepts, the relatedness between them were quantified by the highest relatedness achieved by their possible concepts as shown in Equation 6.6.

$$sim(w_1, w_2) = \max_{c_1 \in S(w_1), c_2 \in S(w_2)} (sim(c_1, c_2)) \tag{6.6}$$

where $S(w)$ is a set of concepts that w may have.

Word Semantic Relatedness to Context

To measure the fitness of a word to an output document, we scoped the context with a window. The window size limits the number of content words immediately before and after the current word. For example, a

[2] Available from `http://nlp.stanford.edu/software/tagger.shtml`

window size of three means the context consists of three content words immediately before and three content words immediately after the current word. The type of content word depends on the experiment setting.

Three statistical features were extracted for each word, w, based on its similarity relatedness to every word, w_i, within a context window.

- **Maximum relatedness (**$max\_rel(w)$**)**: the highest relatedness value w has with its neighboring words. It is a measure of the single best fitness of w to the context.
- **Average relatedness (**$average\_rel(w)$**)**: the average of all the relatedness values w has with its neighboring words. It is a measure of leveraging the overall fitness of w to the context.
- **Average relatedness of top 3 (**$max3\_rel(w)$**)**: the average of the top three relatedness values w has within its neighbors. It is a trade-off between the above two features.

The definition of the three features is shown in Equation 6.7, in which $C(w)$ represents the context of w with a certain window size.

$$
\begin{aligned}
max\_rel(w) &= \max_{w_i \in C(w)} (sim(w, w_i)) \\
avg\_rel(w) &= \sum_{w_i \in C(w)} sim(w, w_i) \Big/ |C(w)| \\
max3\_rel(w) &= \sum_{i \in \arg\max 3(sim(w, w_i))} sim(w, w_i) \Big/ 3
\end{aligned}
\tag{6.7}
$$

6.3 Experiment

6.3.1 Experiment Setting

The experiment setup described in this chapter is the same as that described in Chapter 5. The semantic features introduced in this chapter were appended to the feature sets used in Chapter 5. Therefore, the confidence measure combined non-linguistic features, syntactic features, and semantic features. To make the annotation simple, we used the symbol representing a measure to also stand for the combination of CSSYN and semantic features derived from the measure. For example, *jcn* also represented the combination of CSSYN and semantic features some of which were derived by using *jcn*. The complementary effect of semantic features to the CSSYN was evaluated.

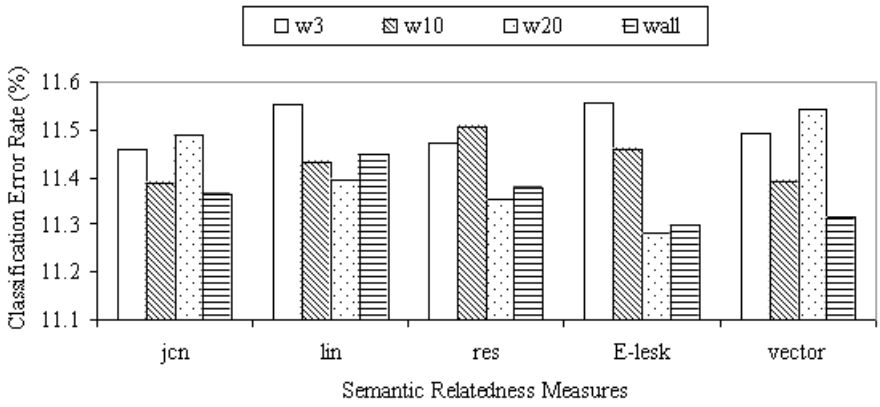


Figure 6.1: Effect of window size on classification error rates when semantic relatedness measures were used to nouns on the Study3 corpus

Because measures such as *res*, *jcn* and *lin* treated WordNet as an IS-A hierarchy, the measures could only be used for nouns and verbs which are hierarchically organized in WordNet. Measures such as *E-lesk* and *vector* were based on glosses and thus could be used for any part-of-speech categories. Given that nouns always play an important role in conveying content information and that they are better constructed than other grammatical categories in WordNet, nouns were served as the basis of the analysis.

To quantify the context, we used different window sizes, including 3 (*w3*), 10 (*w10*), 20 (*w20*), and *all* (*wall*). *all* represents to use all the words in a document as context.

WordNet::Similarity [3] [83] was used to obtain the semantic relatedness between concepts. WordNet 3.0 was used as the dictionary.

6.3.2 Experiment Results on the Study3 Corpus

Classification Error Rate

Figure 6.1 shows the classification error rates of five measures when they were applied to nouns under different window size. Each bar represents one window size. On average, a better CER was achieved under *wall*. Compared to *w3*, every measure achieved better CER under *wall*. Compared to *w10*, *res*, *E-lesk*, and *vector* achieved better CER, and *jcn* and *lin* achieved similar CER under *wall*. Compared to *w20*, although *lin*, *res*, and *E-lesk* achieved a slightly worse CER, *jcn* and *vector* achieved substantially better CER under *wall*. Under *w10* and *w20*, all measures achieved better CER than did *w3*, with only a few exceptions. Therefore, we used *wall* as the window setting in the following analyses.

[3] Available from `www.d.umn.edu/~tpederse/similarity.html`

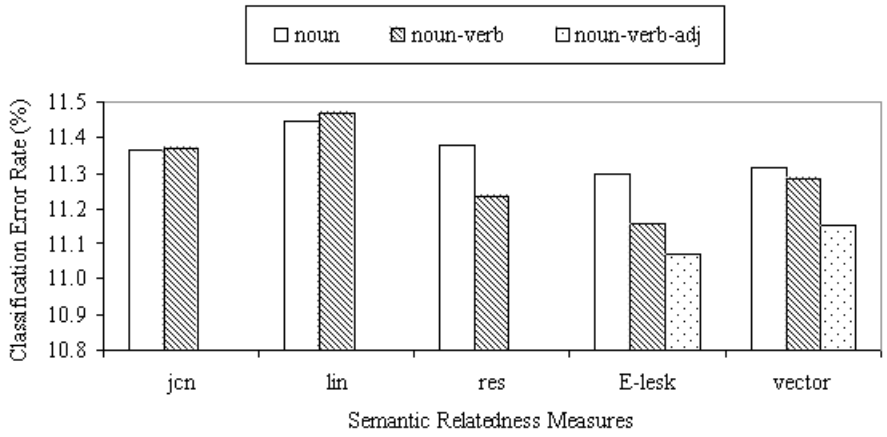


Figure 6.2: Classification error rates of semantic relatedness measures under nouns, noun-verbs, and noun-verb-adjectives settings on the study3 corpus

All measures could be used on verbs, and *E-lesk* and *vector* could also be used on adjectives. The CER of measures after taking verbs and adjectives into consideration under *wall* are shown in Figure 6.2. After verbs were considered, *E-lesk* and *res* achieved lower CER than that achieved when only nouns were considered, while other measures only underwent marginal changes on CER. When adjectives were further considered, both *E-lesk* and *vector* improved the CER over that achieved when nouns and verbs were considered.

Table 6.1 reports the experiment results of the CER achieved under selected settings and by selected measures on the study3 corpus, in which Table 6.1(a) reports descriptive statistics and Table 6.1(b) shows relative improvements. Two word settings are included: nouns alone and content words (noun-verb-adjective). Results of the top two measures from each group are reported, including *jcn* and *res* in the corpus-enhanced group and *E-lesk* and *vector* in the gloss-based group.

(a) Descriptive statistics of CERs achieved by different measures under two settings

			Nouns				Content Words	
	CS	CSSYN	*jcn*	*res*	*E-lesk*	*vector*	*E-Lesk*	*vector*
Mean (%)	12.57	11.54	11.37	11.38	11.30	11.32	11.07	11.15
Std.	6.36	5.71	5.97	5.69	5.87	5.84	5.97	5.74

(b) Relative improvement (%) of CERs achieved by different measures under two settings

	Nouns				Content Words	
	jcn	*res*	*E-lesk*	*vector*	*E-Lesk*	*vector*
CS	9.55	9.47	10.10	9.94	11.93	11.30
CSSYN	1.47	1.39	2.08	1.91	4.07	3.38

Table 6.1: Classification error rates of confidence measures on the Study3 corpus when non-linguistic, syntactic and semantic features were included

Compared to the CER achieved by *CSSYN*, more relative improvements were achieved in the content

(a) Descriptive statistics of Fs, RECs, and PREs achieved by different measures under two settings

				Nouns				Content Words	
		CS	CSSYN	*jcn*	*res*	*E-lesk*	*vector*	*E-Lesk*	*vector*
F	Mean (%)	38.08	46.52	49.20	49.10	50.03	50.14	51.54	50.75
	Std.	15.07	16.62	14.75	13.92	13.64	13.08	12.65	13.10
REC	Mean (%)	27.96	37.11	39.73	39.70	40.44	40.57	41.80	40.99
	Std.	12.76	16.53	15.30	14.83	14.71	14.20	13.93	13.86
PRE	Mean (%)	62.91	69.56	71.93	69.76	71.26	71.13	72.67	71.06
	Std.	17.97	15.18	14.82	14.06	13.91	13.57	13.25	13.62

(b) Relative improvement (%) of Fs, RECs, and PREs achieved by different measures under two settings

		Nouns				Content Words	
		jcn	*res*	*E-lesk*	*vector*	*E-Lesk*	*vector*
F	CS	29.21	28.94	31.38	31.67	35.35	33.27
	CSSYN	5.76	5.55	7.55	7.78	10.79	9.09
REC	CS	42.10	41.99	44.64	45.10	49.50	46.60
	CSSYN	7.06	6.98	8.97	9.32	12.64	10.46
PRE	CS	14.34	10.89	13.27	13.07	15.51	12.96
	CSSYN	3.41	0.29	2.44	2.26	4.47	2.16

Table 6.2: F measure, REC, and PRE of confidence measures on the Study3 corpus when non-linguistic, syntactic and semantic features were included

words setting than those in the nouns setting, and the largest relative improvement was achieved by *E-lesk* at 4.07% over that achieved by *CSSYN*, as shown in Table 6.1(b).

In comparison to the CER achieved by CS, the relative improvements ranged from 9.47% to 11.93%, with *E-lesk*'s being the largest in the content words setting.

F, PRE, and REC

The results in F measure, PRE, and REC, including the descriptive statistics (Table 6.2(a)) and relative improvement (Table 6.2(b)) under the nouns alone and content words settings by the selected measures are reported in Table 6.2.

Similar to the results on CER, the combination of semantic features from every semantic relatedness measure and *CSSYN* achieved better F measure than that achieved by *CSSYN* alone. As observed from CER, the largest relative improvement (10.79%) in F measure over that of *CSSYN* was achieved by *E-lesk* under the content words setting, as evidenced in Table 6.2(b).

Further analysis of F measure in terms of REC and PRE reveals how F measure was impacted by REC and PRE. As shown in Table 6.2(b), compared to REC obtained by *CSSYN*, the relative improvements in REC achieved by all measures ranged from 6.98% to 12.64%. However, the improvements in PRE over that of *CSSYN* were weaker, and only ranged from 0.29% to 4.47%.

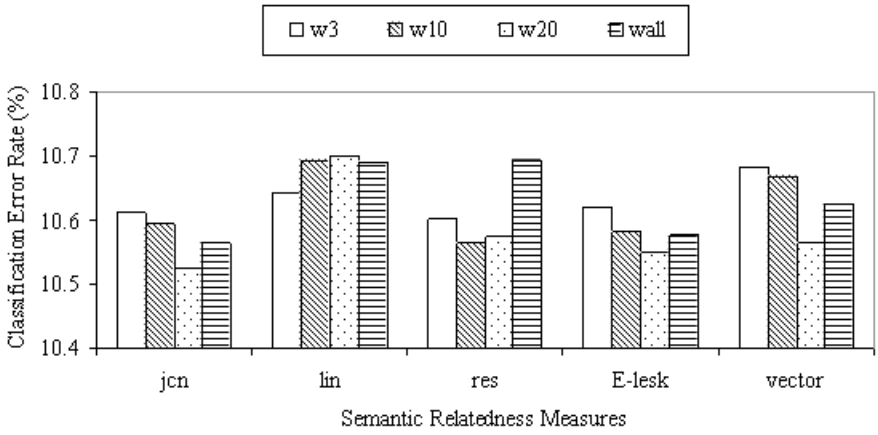


Figure 6.3: Effect of window size on classification error rates when measures were used for nouns on the WSJ corpus

The relative improvement in REC showed trends similar to the relative improvement in PRE. The largest improvement in REC over that achieved by *CSSYN* was 12.64%, which is much higher than the largest improvement in PRE (4.47%). Both the largest improvements in REC and PRE were achieved by *E-lesk* under the content words setting. *E-lesk* achieved a 49.50% relative improvement in REC and a 15.51% relative improvement in PRE over the corresponding results for CS.

6.3.3 Experiment Results on the WSJ Corpus

Classification Error Rate

Figure 6.3 shows the classification error rates of the five semantic relatedness measures under different window size when only nouns were considered. Each bar represents a specific window size. *lin* showed a different trend than did other measures, and CER achieved by *lin* worsened as window size increased. On average, all the measures except *lin* achieved the best CER under *w20*. Compared to *w3* and *wall*, all the measures except *lin* achieved better CERs under *w20*. Compared to *w10*, *jcn*, *E-lesk*, and *vector* achieved better CER, and *res* achieved similar under *w20*. Therefore, we chose *w20* as the window setting in the following analyses.

CERs of the five semantic relatedness measures following the incorporation of verbs and adjectives are shown in Figure 6.4. After verbs were considered, all the measures showed worse CERs than those achieved by using nouns alone. When adjectives were further considered, *E-lesk* and *vector* achieved worse CERs than those achieved when only nouns and verbs were considered. Therefore, only the results of the four measures under the nouns setting were selected for comparison to results obtained by *CSSYN* and CS in the following analysis.

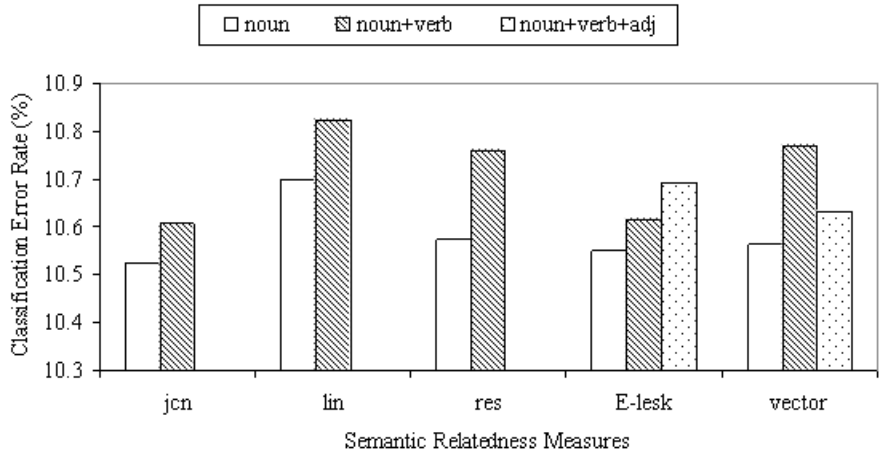


Figure 6.4: Classification error rates of measures for nouns, noun-verbs, and noun-verb-adjectives on the WSJ corpus

(a) Descriptive statistics of CERs achieved by different measures under nouns setting

	CS	CSSYN	Nouns			
			jcn	*res*	*E-lesk*	*vector*
Mean (%)	11.41	10.75	10.52	10.57	10.55	10.56
Std.	4.38	3.87	3.76	3.92	3.71	3.78

(b) Relative improvement (%) of CERs achieved by different measures under nouns setting

	Nouns			
	jcn	*res*	*E-lesk*	*vector*
CS	7.80	7.36	7.54	7.45
CSSYN	2.14	1.67	1.86	1.77

Table 6.3: Classification error rates of confidence measures on the WSJ corpus when non-linguistic, syntactic and semantic features were included

In Table 6.3, descriptive statistics (Table 6.3(a)) and relative improvement (Table 6.3(b)) of CERs achieved under the selected setting and by the selected measures on the WSJ corpus are reported.

jcn achieved the largest relative improvement in CER (2.11%) over *CSSYN*, as evidenced in Table 6.3(b). Compared to the corresponding CER achieved by CS, all the selected measures yielded relative improvements of CER over 7%.

F, REC and PRE

In Table 6.4, descriptive statistics (Table 6.4(a)) and relative improvement (Table 6.4(b)) of F measure, REC, and PRE achieved under the selected settings and by the selected measures on the WSJ corpus are reported.

The finding in the improvement in F measure over that achieved by *CSSYN* was similar to the improvement in CER over that achieved by *CSSYN*. The four measures achieved similar results in F and *jcn* had the

(a) Descriptive statistics of Fs, RECs, and PREs achieved by different measures under nouns setting

				Nouns			
		CS	CSSYN	*jcn*	*res*	*E-lesk*	*vector*
F	Mean (%)	16.35	24.19	26.29	25.93	25.58	25.89
	Std.	7.96	10.70	11.33	9.60	12.09	11.56
REC	Mean (%)	9.63	15.64	17.03	16.57	16.58	16.78
	Std.	4.92	9.33	9.77	8,32	10.23	9.84
PRE	Mean (%)	58.20	69.45	71.61	73.03	70.74	70.41
	Std.	22.38	17.89	19.86	15.82	15.92	19.64

(b) Relative improvement (%) of Fs, RECs, and PREs achieved by different measures under noun setting

		Nouns			
		jcn	*res*	*E-lesk*	*vector*
F	CS	60.80	58.59	56.45	58.35
	CSSYN	8.68	7.19	5.75	7.03
REC	CS	76.84	72.07	72.17	74.25
	CSSYN	8.89	5.95	6.01	7.29
PRE	CS	23.04	25.48	21.55	20.98
	CSSYN	3.11	5.15	1.86	1.38

Table 6.4: F measure, REC, and PRE of confidence measures on the WSJ corpus when non-linguistic, syntactic and semantic features were included

largest relative improvement (8.68%), as shown in Table 6.4(b).

Further investigations of F measure into REC and PRE revealed that REC showed the same improvement pattern as F. Compared to corresponding results achieved with *CSSYN*, the largest relative improvement in REC was 8.89%, while the largest relative improvement on PRE was 5.15%, which were achieved by *jcn* and *res* respectively. Compared to corresponding results achieved with CS, a 76.84% relative improvement was achieved by *jcn* in REC and a 25.48% relative improvement was achieved by *res* in PRE.

6.3.4 Discussion

In light of the above results from our experiments on the semantic relatedness measures using the study3 corpus and the WSJ corpus, we can find that, with a few exceptions, larger context is generally better than smaller context for error detection. On average, context size of all words for the study3 corpus and of 20 words for the WSJ corpus yielded the best performances. A smaller window size of 3 yielded the worst performance in most settings on both corpora. Of course context should not be expanded without any limit, because additional noises may overwhelm the actual signals, as evidenced by the superior performance of *w20* over *wall* on the WSJ corpus. Determination of the appropriate size of context is a research issue for further investigation.

	noun	verb	adjective
study3	20.36	14.90	5.73
WSJ	30.16	16.75	7.44

Table 6.5: Distribution of content words in both the study3 corpus and the WSJ corpus

The study3 corpus and the WSJ corpus achieved their best performance in two different settings. The difference could be caused by the different speech styles of the two corpora. The WSJ corpus consists of news style dictation delivered by professional journalists, and the study3 corpus contains common office communication style speech delivered by college students. Speech utterances in the WSJ corpus are more topic-oriented and compact than the utterances of the study3 corpus which are relatively loose in topic. Table 6.5 shows the distribution of all content words for different grammatical categories in both the study3 corpus and the WSJ corpus. The table also shows the difference between the distributions of each grammatical category in the two corpora. The percentage of nouns in the WSJ corpus is much higher than that in the study3 corpus, which has a relatively balanced noun-verb proportion.

6.4 Summary

In this chapter, we evaluated the performance of various WordNet-based semantic relatedness measures. It was found that *jcn* had the best performance when only nouns were used, and this observation confirms other research findings. *E-lesk* showed a better performance than *jcn* on the study3 corpus under larger window size. On both corpora, the incorporation of word relatedness features led to small but consistent improvement. Compared to the CER obtained by CS features alone, the combination of all linguistic features (including both syntactic and semantic features) and CS features achieved improvements in CER: an 11.93% relative improvement on the study3 corpus and a 7.80% relative improvement on the WSJ corpus in their respective settings for best performance. Moreover, the relative improvements in F measure by the combination of all features are also big in their respective settings for best performance: 35.35% on the study3 corpus and 60.80% on the WSJ corpus.

Chapter 7

LEXICAL COHESION ANALYSIS

A text or a document is typically composed of sentences. The arrangement of sentences is not random but is semantically dependent. The semantics of text is conveyed through the cohesion of sentences. Halliday and Hasan [36] defined cohesion as "relations of meaning that exist within the text, and that define it as a text." Cohesive relations unify different parts of the text, and connect a sentence to surrounding sentences to make the text coherent. An SR error may break the flow of semantic relatedness and therefore not cohere with other words in the text. In this chapter, we explore the detection of SR errors through analysis of text cohesion, and more specifically, lexical cohesion.

7.1 Lexical Cohesion

7.1.1 Cohesion

Halliday and Hasan [36] classified cohesion into five categories based on the linguistic form of which cohesion may take: reference, substitution, ellipsis, conjunction, and lexical cohesion. We will briefly introduce the first four categories and will focus on lexical cohesion in the remainder of this chapter.

Reference

Reference is a type of cohesion in which the interpretation of an element is not based on its own meaning but on the meaning of other items it references. Reference itself is a semantic relation between words. However, because reference is always represented through close words such as pronouns, it thus is grammatical in

representation form. In example 1, several reference relations exist: *his* refers to *Jim*, *they* refers to *Jim and his friends*, and *it* refers to *dinner*.

Example 1: Jim went out for dinner with his friends. They all enjoyed it.

Substitution and Ellipsis

Substitution and ellipsis represent grammatical relations between items. Substitution is the replacement of one item with another, and ellipsis is a special case of substitution and is the omission of an item. Substitution and ellipsis are also grammatical in representation form, and their boundary to reference is sometimes ambiguous. Example 2 demonstrates substitution because *did* in *Mary did* substitutes *clean*. Example 3 demonstrates ellipsis because *clean* is omitted after *did* in *he did*.

Example 2: Did Jim clean the room? No, Mary did.

Example 3: Did Jim clean the room? Yes, he did.

Conjunction

Conjunction is a type of cohesion which connects semantic meanings between components indirectly through specific meanings held by words or phrases which grammatically connect components. From the point of representation, conjunction is mainly grammatical and may be lexical. In example 4, *therefore* connects the meaning of the two sentences through a consequent relation.

Example 4: Jim works hard. Therefore he succeeds in the competition.

7.1.2 Lexical Cohesion

Lexical cohesion, by definition, is a lexical relation based on the selection of vocabulary, and it includes various kinds of semantic relations existing between two items. Unlike other types of cohesion, lexical cohesion is represented through open class words such as nouns. Another unique characteristic of lexical cohesion is that it need not be grammatically analyzed like the other four types of cohesion because it can be directly observed on the surface of the text.

Lexical cohesion is the most common type of cohesion. Hoey [41] summarized the frequencies of five types of cohesion on seven different types of text analyzed by Halliday and Hasan [36], and found that lexical

Repetition	[1.a] Jim is reading a *novel*.
	[1.b] He thinks it is a good *novel*.
Synonym/Near-Synonym	[2.a] A new *bicycle* is what Jim wants for the gift.
	[2.b] He hopes to ride the new *bike* to school.
Superordinate	[3.a] Jim eats a *banana* everyday.
	[3.b] Doctor told him that this *fruit* is high in potassium.
General Words	[4.a] Jim plans a trip to *Antarctic*.
	[4.b] This will be the first time he visit that *place*

Table 7.1: Example of reiteration relations

cohesion has the highest frequency and comprises 42% of all cohesion relations. Therefore, lexical cohesion is the most important type of cohesion and is the focus of our research in this chapter.

Halliday and Hasan [36] also classified lexical cohesion into two groups, reiteration and collocation, based on the nature of the cohesive relation, be it reference- or form-related.

Reiteration

Reiteration is the action of repeating and has other variations in addition to simple repetition. Table 7.1 lists and illustrates possible relations in the reiteration category. As listed first, reiteration is the exact repetition of a lexical item, which is the most restrictive case. *novel* in sentence 1.b is the repetition of *novel* in sentence 1.a. Secondly, reiteration can be represented though the usage of a synonym or near-synonym of a lexical item. *bicycle* in sentence 2.a and *bike* in sentence 2.b are synonyms. Third, reiteration can be realized by using a superordinate of a lexical item. As shown in sentence 3.b, *fruit* is more general than and is the hypernym of *banana*. Finally, the most loosely defined type of reiteration is the replacement of a lexical item with a general word. In sentence 4.b, *place* is a general word for locations such as *Antarctic*.

Collocation

Lexical cohesion in the collocation group has a large variation, but it follows the basic principle that cohesion is expressed through the association of regularly co-occurring items. Collocations can be further divided into two groups based on whether or not the relation of lexical items is systematically semantic.

Several types of systematically semantic relationships can be used to represent lexical cohesion. For example, antonym describes the relationship of a pair of lexical items with their opposite meanings, such as *man* and *woman*, *love* and *hate*, and *beautiful* and *ugly*. Meronym/holonym describes the part-whole relationship between a pair of items, such as *house* and *porch*, and *computer* and *memory*. Coordinate items

are those items sharing the same hypernyms, such as *novel* and *story*, and *run* and *jog*.

There also exist item pairs associated with each other through relations which are difficult to categorize under any of the above systematically semantic terms. Instead these relations could be established by identifying item pairs in similar lexical contexts. Then the lexical cohesion exists between items occurring in the same lexical environment.

7.1.3 Applications by Analyzing Lexical Cohesion

In a text, cohesion does not only exist between a pair of words but also can span across multiple words and form cohesive chains [36]. Typically, many interactive cohesive chains exist in a text. When only lexical cohesion is considered, cohesive chains can simply be referred to as lexical chains [78]. Lexical chains have been used in various natural language applications, such as text segmentation [78], information retrieval [106], text summarization [3], topic detection [107], question answering [76], correction of malapropisms [40], etc. In these applications, the cohesive relationships between words were extracted from knowledge sources such as WordNet or Roget's thesaurus.

Hirst and St-Onge's Correction of Malapropisms Based on Lexical Chains

Hirst and St-Onge [40] applied lexical chains to detect and correct malapropisms in written text. Malapropisms are unintentionally misused words that are similar, often phonetically, to but semantically different from intended words. The idea underlying Hirst and St-Onge's method is that the more semantically different a word is from other words in a text, the higher probability the word is a malapropism. Lexical chains can be used to connect semantically related words. It was assumed that if a word does not belong to any lexical chains and is orthographically similar to a word that can be incorporated into a lexical chain, the word is likely a malapropism. The construction of lexical chains involved three factors: word choice, relation choice, and chain construction.

word choice Words that can be found in the noun index of WordNet or can be morphologically transformed into the words contained therein were candidate words. If possible, compound words and phrases included in WordNet were extracted first from the text. All these words must not appear in a stop-word list, which contains close-class words and high-frequency words.

relation choice Twelve semantic or lexical relations defined in WordNet were used. They were also-see,

antonym, attribute, cause, entailment, holonym, hypernym, hyponym, meronym, pertain, similar, and synonym. Word repetition and word as a part of phrase and compound were also used.

chain construction To build chains, relations between two words were divided into three groups by relation strength:

- Extra-strong: one word is a literal repetition of the other
- Strong: a strong relation can exist in the following conditions:
  - one sense of a word is synonymous of one sense of the other word
  - one sense of a word is connected to one sense of the other word by a horizontal link (e.g. antonymy, similarity, and see also)
  - one word is a compound word or a phrase that contains the other word
- Medium-strong: a medium strong relation exists if at least one allowable path connects one sense of each word. A path was defined as a sequence of at most five links between two synsets. Several rules were defined to determine the existence of a path.

Different weights were assigned to relations in different group. Unlike the extra-strong and strong relations which were assigned to a constant weight, the weights of the medium-strong relations depended on path length and number of direction changes.

When a word initiated a new chain, all of its synsets attached to it because no contextual information was available to discriminate them. Upon a new word w, the decision of which chain w would be inserted were firstly made. To make this decision, the relations between w and words in existing chains were judged. Distance restrictions between w and words in chains were enforced for each group of relations: 1) no limitation for extra-strong relation; 2) distance of seven sentences for strong relation; 3) distance of three sentences for medium-strong relations. When selecting a chain to add w, an order of relations between words to be considered was enforced: extra-strong, strong, and medium-strong. If w can not be inserted into any existing chains, a new chain was initiated.

After a new word was added to a chain, the synsets of the new word and words in the chain were updated according to the relations that connect the new word and words in the chain. The new word's unconnected synsets were then eliminated, and the entire chain was scanned to delete unconnected synsets of other words in the chain. This greedy strategy of removing synsets when inserting new words gradually narrowed down the context and disambiguated words by eliminating unused senses.

After lexical chains were constructed, words with no connection to other words were candidates for errors. Possible spelling variations of a candidate error from a spelling checker could be added to existing chains. If a variation can belong to any of the existing chains, the word was marked as an error.

Hirst and Budanitsky's Correction of Malapropisms Based on Lexical Cohesion

Hirst and Budanitsky [39] also proposed a method to detect and correct malapropisms based on the analysis of lexical cohesion. Instead of explicitly constructing lexical chains, they treated the text as a bag of words and linked words that may have certain lexical relations.

Hirst and Budanitsky used the same criteria to choose candidate words as [40], though they chose different relations. They used the semantic measure proposed by Jiang and Conrath [47] to measure the semantic similarities between words. Correspondingly, WordNet was treated as a hierarchical structure such that only semantic relations such as synonym and hypernym/hyponym, a subset of the relations used in [40], were used. Minimal word sense disambiguation was performed, and every possible sense of a word was kept as long as it connected to one sense of any other words. Therefore, a word linked to another word if one of its senses was semantically similar to a sense of the other word. Given that the similarity computed by Jiang and Conrath's measure is not binary but numerical, they set a threshold to convert the numerical value into a binary value. Two concepts were considered related if their semantic similarity was above the threshold and unrelated if their semantic similarity was under the threshold. The threshold was set as the value under which Jiang and Conrath's measure achieved the best classification accuracy on 65 word pairs in Rubenstein and Goodenough's experiment [94].

The detection and correction procedures were the same as those in Hirst and St-Onge's experiment. The same corpus used by Hirst and St-Onge which includes 500 articles from Wall Street Journal corpus was used to test the proposed method. One word in every 200 words was replaced by one of its spelling variations. After eliminating articles without malapropizable words according to their criteria, on average there were 2.8 malapropisms per article.

To analyze the effect of context length on detection performance, they used paragraph as the basic unit and tested contexts consisting of one paragraph, three paragraphs, five paragraphs, and the entire article. Precision, recall, and F measure were used to test the detection performance. While no significant difference in detection performance was found between different context lengths, the best F measure (25.4%) was achieved when one paragraph was used as the context; and the corresponding precision was 18.4% and

recall was 49.8%. Results were compared with those of Hirst and St-Onge. Hirst and St-Onge's lexical chain method achieved an overall single-point performance of 17.4% F measure, which was lower than the performance of 23.7% F measure achieved with a context length of one paragraph.

7.2 Detecting Errors through Lexical Cohesion Analysis

We treated the speech recognition output as a bag of words and linked each word with potentially related words as Hirst and Budanitsky did. Two issues should be addressed prior to creating a linked graph: the selection of candidate words and the measurement of semantic relatedness between words.

7.2.1 Candidate Words

Only content words bear meaning, so we selected nouns, verbs, and adjectives as candidates. However, only verbs that have derivation relations in WordNet and are derivationally related to nouns were included. For adjectives, only those pertainym to nouns were included. Words were not selected if they belong to the stop words list, which consists of high frequency words and close words. Instead of referring to WordNet in determining the parts-of-speech of words, an English part-of-speech tagger was used to parse the speech recognition output. When nouns alone were considered, 19.99% of words in the study3 corpus and 30.05% of words in the WSJ corpus were selected as candidate words. When all selected content words were consider, the selected candidate words were 31.75% and 43.89% of output words in the study3 and WSJ corpus respectively.

7.2.2 Choice of Semantic Measure

Two different groups of semantic relatedness measures were used. One included WordNet-based measures, and the other included corpus-based measures. Measures based on WordNet can help capture reiteration relations and systematic collocation relations. Measures based on statistical analysis of corpora are able to identify non-systematic collocation relation to a certain extent.

Measures Based on WordNet

We chose Jiang and Conrath's measure (*jcn*), and Extended Lesk measures (*E-lesk*) as WordNet-based measures. Both of them showed better performance than other WordNet-based measures in Chapter 6. In addition,

Jiang and Conrath's measure achieved the best performance in [39].

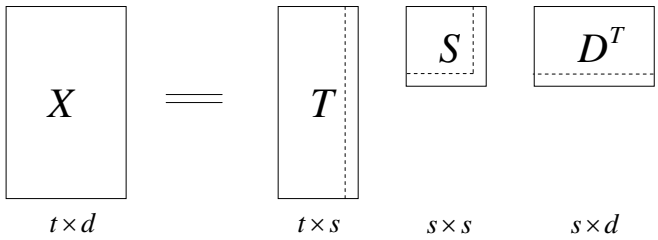
Corpus-Based Measures

We used latent semantic analysis to perform word co-occurrence analysis. Words that co-occur in one document were assumed to be semantically related. As will be discussed below, the degree of semantic relatedness between a pair of words was measured by the cosine similarity between their vectors in the semantic space. To simplify annotation, we refer to this measure as *LSA-based*.

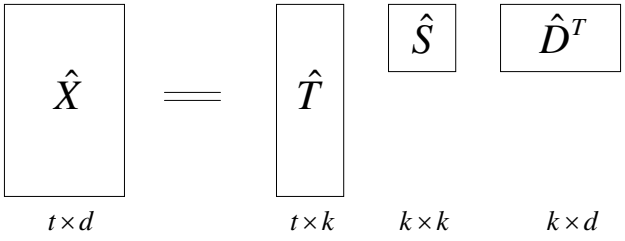
Latent Semantic Analysis LSA was first developed for information retrieval and was referred to as latent semantic indexing [23]. It is a fully automatic method and does not dependent on any pre-defined knowledge. It is a technique for extracting the contextual usage of words by statistically analyzing a text corpus [57]. LSA tries to discover deep rather than surface relations between words and context and to infer the underlying meaning of text. In general, many words can represent the same meaning. A word may have more than one meaning in different contexts. Therefore synonym and polysemy may introduce noise into text and pose major problems for word-based methods. To tackle this problem, LSA first represents the text in a high dimension then reduces the number of dimensions to only the most important.

A matrix X $t \times d$ of term and document can be built on a given text corpus. In X, rows stand for words, and columns correspond to documents. The dimension of X depends on the term selection criteria and the number of documents in the text corpus. Each cell (i, j) represents term i in document j, and the cell's value is derived using certain weighting schema. After a singular value decomposition is performed on X, X is factored into the production of three matrices as shown in Figure 7.1(a). S is the diagonal matrix of singular values with rank r, which is less than or equal to the minimum of t and d. The singular values, representing the scaling factor of each dimension, are sorted in descending order along the diagonal line. T is the matrix with orthonormal unit-length columns ($T^T T = I$) and represents the original term vectors by left singular vectors. Similarly, D has the orthonormal unit-length columns ($D^T D = I$) and represents the original document vectors by right singular vectors. The product of these three matrices perfectly restores the original matrix, as shown by Equation 7.1.

$$X = TSD \tag{7.1}$$



(a) Original matrix



(b) Reduced matrix

Figure 7.1: Latent Semantic Analysis

An approximate matrix that can maintain important relations while eliminating the noise is used to reduce noise in the original matrix. Singular value matrix S is reduced to $\hat{S}$ with rank k by only keeping the top k singular values which are expected to represent the most important dimensions. Correspondingly, both T and D are reduced into $\hat{T}$ and $\hat{D}$ by keeping the first k columns, as shown in Figure 7.1(b). The production of the three reduced matrices (see Equation 7.2) is an approximate of the original matrix and a least-square best fit.

$$X \approx \hat{X} = \hat{T}\hat{S}\hat{D} \tag{7.2}$$

With the reconstructed matrix, the similarity or distance between documents, terms, or terms and documents can be computed by using appropriate measures such as dot product, cosine, and Euclidean distance. The most commonly used measure is the cosine between two vectors [57], which measures the overlaps along each dimension and has the dimension-matching properties [56]. Given two terms t_i and t_j represented as vectors of length n, their cosine similarity is computed according to Equation 7.3, which is the angle between two vectors.

$$cosine(t_i, t_j) = \frac{t_i \cdot t_j}{|t_i| * |t_j|} = \frac{\sum_{d=1}^{n} t_{i,d} * t_{j,d}}{\sqrt{\sum_{d=1}^{n} t_{i,d}^2} * \sqrt{\sum_{d=1}^{n} t_{j,d}^2}} \tag{7.3}$$

Term weight As mentioned above, the value of each cell (i, j) in X is determined by certain weighting schema. Generally, the weight of the term is the product of two parts, namely local weight and global weight, as shown in Equation 7.4. Local weight is specific to each cell in X, and global weight is the same for every cell in a row.

$$termweight(i, j) = localweight(i, j) \times globalweight(i) \tag{7.4}$$

Local weight is used to measure the importance of a term in the document. Generally, frequency of a term positively correlates with its importance. The commonly used local weighting schema is term frequency $freq$ or the logarithm of $freq$. $freq(i, j)$ is the number of times term i appears in document j. Global weight calibrates the importance of a term across all the documents in a corpus. The more general a term is across an entire corpus, the less importance the term is to a document. Various global weighting functions have been proposed, such as normalization, inverse document frequency, and entropy.

Dumain [26] compared several combinations of different local and global weighting methods on information retrieval tasks and found that the combination of the logarithmic term frequency and entropy [69] produced the best performance. We used this combination as the weighting schema in the implementation of LSA in this chapter, which is represented by Equation 7.5.

(7.5a) $$localweight(i,j) = \log(freq(i,j)+1)$$

(7.5b) $$\begin{aligned} globalweight(i) &= 1 - entropy(i) = 1 - \frac{H(d|i)}{H(d)} \\ &= 1 - \frac{-\sum_{j=1}^{N} p(i,j)\log p(i,j)}{\log(N)} \\ &= 1 + \frac{\sum_{j=1}^{N} p(i,j)\log p(i,j)}{\log(N)} \\ where, p(i,j) &= \frac{freq(i,j)}{\sum_{j} freq(i,j)} \end{aligned}$$

Logarithmic term frequency decreases the effect of large differences in term frequency by using a log function transformation. In Equation 7.5b, N is the number of documents in the corpus and $H(d)$ is the entropy of document distribution. $H(d|i)$ is the entropy of a term across documents and measures the distribution of the term over documents. When using a constant (1 in this case) to subtract the entropy as shown in Equation 7.5b, a term is assigned a minimum global weight when it is equally distributed across all the documents and has the maximum entropy, and is assigned the maximum global weight when it appears only in several documents.

7.2.3 Connecting Words

Word Sense Disambiguation

WordNet-based measures compute the semantic relatedness between concepts rather than words. Therefore, to add a link between a pair of words, the senses of the words should be disambiguated. To identify the sense for each word, we adopted the assumption that each discourse has one sense [31]. We revised the three-step procedure proposed by Galley and McKeown [32] for disambiguating word sense during lexical chain construction to fit into our case, as described below:

1. All possible links between potential senses of words were maintained, forming a big concept graph.

2. The senses of all words were selected all at once. The number of links for each sense of a word was computed for all words. The sense with the largest number of links was selected. If two or more senses held the same number of links, the number of types of links for each sense of a word was computed. The

sense with the largest number of types of links was selected. If a tie existed, the semantic relatedness of their links was analyzed. The total weight (semantic relatedness) for each type of links was computed. If the weights were comparable, the sense with the highest weight was selected. If the weights were not comparable or a tie still existed, the sense appearing first in WordNet was selected.

3. When the sense of every word was identified, all other senses and their links were deleted from the graph. The resulting graph was a word graph.

We gave the number of links and number of types of links precedence over the sum of semantic relatedness when selecting word sense. The reason is that we used different methods to judge semantic relatedness between concepts. In [32], WordNet relations were used and they were assigned different weights based on the type of relation and allowed distance. These weights were comparable. However, we used the *jcn* and *E-lesk* measures, and the semantic relatedness values computed by them could not be compared directly. We could only compare weights computed by the same type of measure. If two competing senses had both types of links, the selection decision could be made only when the comparison results on the weight of links for both types had the same direction. Moreover, only links that had semantic relatedness over a predefined threshold were assumed to be relevant.

Setting the Thresholds for WordNet Based Measures

To set the threshold for the WordNet-based measures, we followed the method used by Hirst and Budanitsky [39] by setting the threshold which can obtain the best classification accuracy on 65 word pairs whose relatedness were judged by human subjects.

Rubenstein and Goodenough [94] selected 48 ordinary English nouns and formed 65 word pairs which had relations ranging from highly synonymous to semantically unrelated. The task of the subjects was to rank the word pairs on a scale of 0 to 4 based on the "similarity of meaning". The higher the number, the more semantically similar the two words are. Table 7.2 shows the 65 word pairs and their corresponding similarity values perceived by subjects.

Figure 7.2(a) shows the average performance of subjects on 65 word pairs. The word pairs were listed in ascending order of average similarity assigned by subjects. The x-axis represents word pairs, and the y-axis represents average similarity values. Figure 7.2(a) reveals a big gap between pair 37 and 38. Pair 37 is the "magician oracle" with similarity of 1.82, and pair 38 is the "crane implement" with similarity of 2.37. Therefore, pair 37 can be treated as a dividing line separating word pairs into two groups. Pairs having a

Word Pair		Similarity	Word Pair		Similarity
cord	smile	0.02	car	journey	1.55
rooster	voyage	0.04	cemetery	mound	1.69
noon	string	0.04	glass	jewel	1.78
fruit	furnace	0.05	magician	oracle	1.82
autograph	shore	0.06			
automobile	wizard	0.11	crane	implement	2.37
mound	stove	0.14	brother	lad	2.41
grin	implement	0.18	sage	wizard	2.46
asylum	fruit	0.19	oracle	sage	2.61
asylum	monk	0.39	bird	crane	2.63
graveyard	madhouse	0.42	bird	cock	2.63
glass	magician	0.44	food	fruit	2.69
boy	rooster	0.44	brother	monk	2.74
cushion	jewel	0.45	asylum	madhouse	3.04
monk	slave	0.57	furnace	stove	3.11
asylum	cemetery	0.79	magician	wizard	3.21
coast	forest	0.85	hill	mound	3.29
grin	lad	0.88	cord	string	3.41
shore	woodland	0.90	glass	tumbler	3.45
monk	oracle	0.91	grin	smile	3.46
boy	sage	0.96	serf	slave	3.46
automobile	cushion	0.97	journey	voyage	3.58
mound	shore	0.97	autograph	signature	3.59
lad	wizard	0.99	coast	shore	3.60
forest	graveyard	1.00	forest	woodland	3.65
food	rooster	1.09	implement	tool	3.66
cemetery	woodland	1.18	cock	rooster	3.68
shore	voyage	1.22	boy	lad	3.82
bird	woodland	1.24	cushion	pillow	3.84
coast	hill	1.26	cemetery	graveyard	3.88
furnace	implement	1.37	automobile	car	3.92
crane	rooster	1.41	midday	noon	3.94
hill	woodland	1.48	gem	jewel	3.94

Table 7.2: Synonymy of theme pairs judged by participants [94]

higher similarity than pair 37 were considered relevant, while pairs having a lower or an equal similarity to pair 37 were treated as irrelevant. The thresholds for WordNet-based measures were set to values that led to the best classification accuracy on the 65 word pairs.

Figure 7.2(b) and Figure 7.2(c) show the similarities on 65 word pairs computed using both Jiang and Conrath's measure and Banerjee and Pedersen's extended lesk measure. For each word pair, the similarity was the maximum similarity between any sense combinations of the word pair. As in Chapter 6, the similarity value was computed using WordNet::Similarity.

Originally, Jiang and Conrath's measure computed semantic distance instead of the semantic similarity. To derive semantic similarity, WordNet::Similarity used the reciprocal of the distance, which resulted in a very large similarity value for synonyms. To make it easy to plot on a graph, we simply replaced the large value with a value higher than all other values (3 was used here). Figure 7.2(b) indicates that the threshold can be set to around 0.25, because any lower value would result in inferior discriminatory ability.

In Figure 7.2(c), there is a gap between the value around 1 and the value around 0.5. Thus the threshold can then be selected from the range between these two values. For the sake of simplicity, we selected 1 as the threshold.

Linking Words

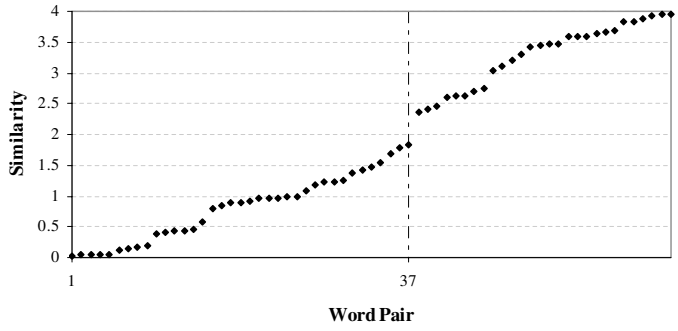
There are always repeated words in a document. A separate lexical relation "repeated" was used to link different occurrences of the same word.

When WordNet-based measures were used, the procedure for word sense disambiguation was used to build the final word graph. When *LSA-based* measure was used, links were discarded for word pairs having similarity less than 0.1. The threshold of 0.1 was used because random word pairs have the cosine similarity of 0.02+/-0.06 [55]. Since a low threshold is likely to introduce much noise and a high threshold is likely to eliminate useful relations, the threshold for cosine similarity would be re-evaluated later.

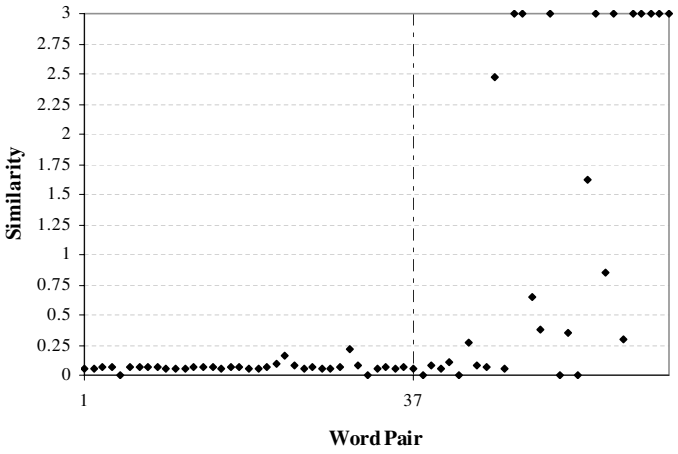
To merge links from different relations, generally a word graph based on WordNet measures was first constructed, and then the repeated links and links from *LSA-based* measure were added.

7.2.4 Error Detection

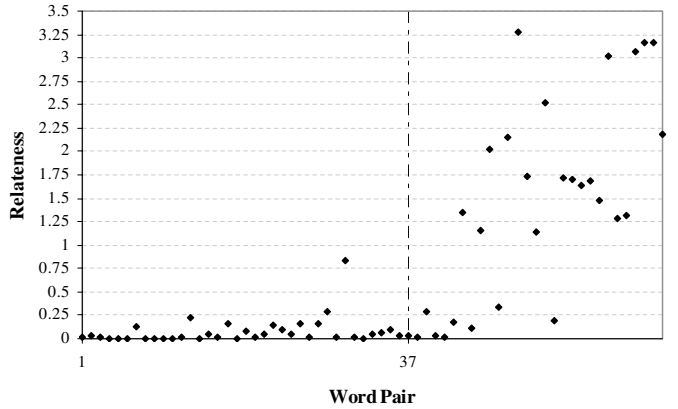
Once the word graph was built, it was augmented with the prediction probabilities from the output of the confidence measure that combined non-linguistic features and syntactic features (*CSSYN*) as described in



(a) human



(b) Jiang and Conrath's measure



(c) Banerjee and Pedersen's extended lesk

Figure 7.2: Similarities of 65 word pairs determined by humans, Jiang and Conrath's measure and Banerjee and Pedersen's extended lesk measure

Chapter 5. As a result, every word w in the word graph had an initial label ($label(w)$), which was the label set by the confidence measure. When confidence measure set the label for a word, it used a threshold of 0.5. Given a word with probability ($prob(w)$), if $prob(w) \geq 0.5$, w was set to be a correct word ($label(w) = 1$); and if $prob(w)$ was less than 0.5, w was set to be an error ($label(w) = -1$). With this setting, the experiment results achieved by (*CSSYN*) in Chapter 5 serve as the baseline for the experiment in this chapter.

A word in the graph can exist in one of three states: with no edge, with repeated edges, and with relation edges only. These three states were disjointed. For a word with both repeated edges and relation edges, it was classified into repeated edges state. Different strategies were used for words in different states. Let C/w refer to the set of words in the word graph except w, A_w refer to the word alternative hypotheses of w generated by a speech recognition system, and a refer to an alternative hypothesis of w.

words with repeated edges (Repeat) For repeated words, we assumed that they should have the same label. When speaking to computers, it is not uncommon that when SR mistakenly recognizes a word to another word, the same error will happen again. If applicable, the majority vote was used to determine the label. All words with the same or morphologically same lexical forms were considered. If more than half of the words shared the same label, then the labels of all the other words were changed. If same number of words sharing two different labels, nothing could be changed because no decision could be made.

words with relation edges only (Edge) The words in this group were related to some words in C/w. We assume that if a word was semantically related to some words that were correct with high confidence and it was not an error with high confidence by itself, it was possibly correct. Words in this state can only be reset to correct if possible, with the hope to correct some mistakenly predicted errors which were related to the context. Words were categorized into three groups based on their probability: correct words with high confidence (CH), errors with high confidence (EH), and others. A word w belonged to the CH group if $prob(w)$ was above an upper-threshold; w belonged to the EH group if $prob(w)$ was below a lower-threshold. If a word was related to a word in CH, we considered it to have a support. For a word w not belonging to EH, its number of supports was counted. If the number of supports of w was not below the support-threshold, it was set to be a correct word.

words with no edge (NoEdge) If a word w does not relate to any other words in C/w, it is a potential error, because it seems out of context. Instead of simply labeling all singletons as errors, alternative

hypotheses of singletons and their probabilities were considered. For a possibly erroneous word, its correct output word may be one of its alternative hypotheses. Therefore, alternative hypotheses of singletons were judged to see if they were related to other words in the context. Only the top alternative hypotheses were selected to use because more noise would be introduced when more hypotheses were used. When any one hypothesis of a singleton was related to some other words and also satisfied certain conditions, the singleton may be an error. Word probability is also an important factor. Although a word does not relate to any other words in the context, it may also be correct if other information has a high confidence to judge it is correct because the semantic analysis is far from perfect by itself. To determine if a singleton w is an error, several steps were followed as shown in Figure 7.3:

1. The words in A_w were sorted in decreasing order of their original scores assigned by a speech recognition system. If w belonged to the top two alternative hypotheses, the hypothesis other than w was selected. If w did not belong to the top two alternative hypotheses, both of the top two hypotheses were selected. If there were alternative hypotheses shared the same score, they were all selected. If the selected set was empty, step 5 was performed. In SR systems, the grammatical category of an alternative hypothesis may not be the same as the output word. It is possible that the selected hypotheses did not satisfy the criteria of candidate word. In such case, selected set was empty and step 5 was performed.
2. Selected alternative hypotheses were each compared with other words in the word graph. If there was one selected alternative hypothesis a that was the same as one non-singleton word w_r, and $label(w_r) = 1$, w was set to be an error, and then the procedure was concluded.
3. If w is the top alternative hypothesis, step 5 was performed.
4. If there was one selected alternative a that was related to one non-singleton word w_r through the semantic relatedness measure, w was set to be an error under two conditions: $label(w_r) = 1$, and $prob(w_r)$ was higher than that of $prob(w)$. If w was set to be an error, and then the procedure was concluded.
5. If $prob(w) < prob - threshold$, w was set to be an error. *prob-threshold* was the threshold set for probabilities of words, and was evaluated as a parameter.

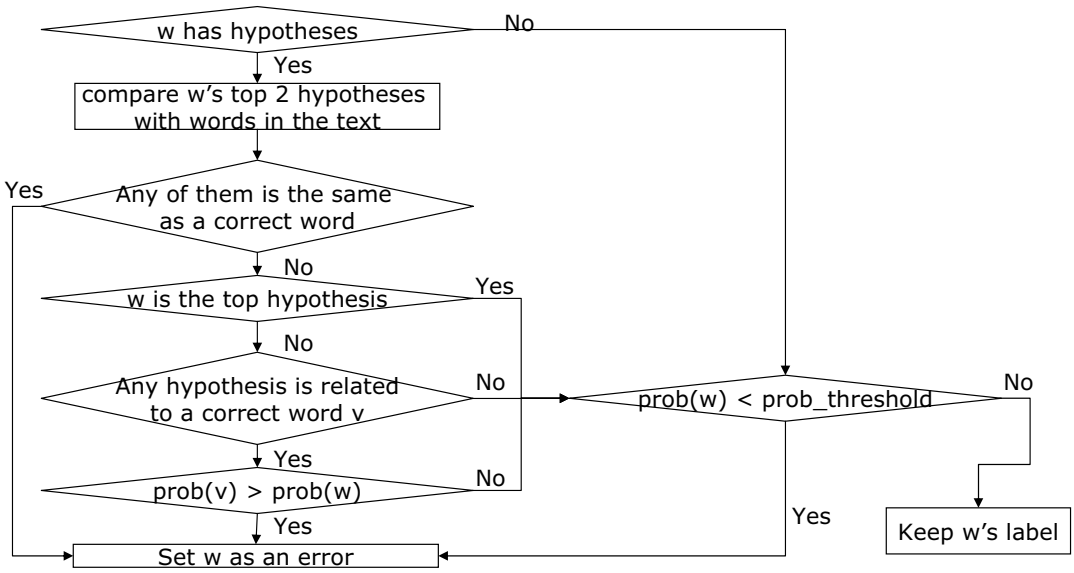


Figure 7.3: Flow chart for detecting SR errors on singleton words

7.3 Experiment

7.3.1 Experiment Setting

Latent Semantic Analysis

To determine word relatedness from latent semantic analysis, a training corpus that is homogenous to test corpus is needed. The reference transcripts, not the SR output, were used for the training purpose.

study3 For the study3 corpus, two dictation corpora as training data were used. One corpus was collected by the Interactive System Research Center at UMBC under the same setting of the study3 corpus with the exception that more scenarios were used and participants were allowed to make the inline corrections during dictation [99]. It includes 121 dictated texts from 15 speakers. Each dictation has about 400 words. The other corpus was collected by Dr. Feng at Towson University. It includes 37 dictations from 37 speakers. Each dictation has about 600 words. The collection procedure was similar to that of the study3 corpus. The dictation software used was Microsoft Vista, and the dictation scenarios used were a subset of those used in the first corpus.

WSJ For the WSJ corpus, the 8000 spontaneous utterances from both the training and development set in the CSR-II corpus were used. The 8000 utterances were dictated by 40 journalists and had the same spontaneous dictation style as the WSJ corpus. The 200 utterances dictated by each journalist cover several

topics. To group the utterances based on topic, utterances from a journalist were merged together in one document and then were automatically segment into topic chunks by C99 text segmentation [19] software [1]. There were totally 1274 topic chunks were generated.

The term-document matrix was built with each non-stop word as a row and a document/topic chunk as a column. A stop word list was used and no stemming was done to the words. JAMA [2], a Java matrix package, was used to get the singular value decomposition of the term-document matrix. For the study3, the matrix had 2213 rows and 158 columns; for the WSJ corpus, the matrix had 6700 rows and 1274 columns. The top 50 dimensions were used.

Word Relatedness Computation

As in Chapter 6, we used WordNet::Similarity as the implementation of WordNet-based measures to compute semantic relatedness between words.

Threshold Setting

To correct the labels of words in the Edge group, the upper-threshold was set to 0.9, and the lower-threshold was set to 0.1. The support-threshold was set to be the smaller number of three and $(|C/w| + 1)/5$.

Experiment Settings

The proposed method was first tested on the constructed word graphs by considering each WordNet-based measure separately. Then the *LSA-based* measure was combined with each WordNet-based measure to find out if the incorporation of collocation relations extracted from statistical analysis can help improve error detection performance. Two types of word choice settings, nouns only and content words, were tested. The *prob-threshold* and the threshold for cosine similarity (*cosine-threshold*) were the two parameters needed to determine.

7.3.2 Experiment Result on the Study3 Corpus

Upperbound

In lexical cohesion analysis, only selected content words may change their labels, and other words keep their labels unchanged. The performance upperbound achieved by lexical cohesion analysis is generally lower

[1] Available from `http://myweb.tiscali.co.uk/freddyyychoi/software/software.htm`
[2] Available from `http://math.nist.gov/javanumerics/jama/`

	CER(%)	Pre (%)	Rec (%)	F (%)
nouns	8.96	82.43	51.66	62.19
content words	7.51	86.19	60.42	70.40

Table 7.3: Upperbounds of CERs, PREs, RECs, and Fs of lexical cohesion analysis on the study3 corpus under both noun words and content words settings

	Nouns		Content Words	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	11.35	5.66	11.33	5.67
PRE	72.38	13.97	72.74	13.39
REC	36.98	16.56	36.86	16.59
F	46.75	16.48	46.70	16.45

Table 7.4: Descriptive statistics of CERs, PREs, RECs, and Fs on the study3 corpus for words with repeated edges in lexical cohesion analysis

than the perfect performance. The upperbounds of CERs, PREs, RECs, and Fs achieved by lexical cohesion analysis were obtained by setting the labels of all selected words to their true labels, which represents the situation that all selected words were correctly labeled. Table 7.3 shows the upperbounds under noun words and content words settings.

Words with Repeated Edges

All the measures had the same performance on words with repeated edges because the repeated relation is independent of relatedness measures. The CER, PRE, REC, and F measure after the majority vote on those words are shown in Table 7.4. Compared to the corresponding results by CSSYN in Chapter 5, there was a slight improvement in CER and F.

Words with Relation Edges Only

Table 7.5 shows the CERs, PREs, RECs, and F measures when words with relation edges were processed. Compared to the corresponding results by CSSYN in Chapter 5, improvement on the four metrics, if any, was rather small.

Words with no Edge

A range of *prob-threshold*s from 0.5 to 1.0 was tested in increments of 0.1. The value of 0.5 was chosen as the starting value because words with probabilities lower than 0.5 were errors by default. Therefore, the performance achieved at a threshold lower than 0.5 is equivalent to that achieved with a threshold of 0.5.

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	11.50	5.70	11.52	5.70	11.50	5.70	11.48	5.68
PRE	69.86	15.13	69.72	15.05	69.86	15.13	70.07	15.13
REC	36.98	16.56	36.98	16.56	36.98	16.56	36.98	16.56
F	46.49	16.65	46.46	16.63	46.49	16.65	46.58	16.74

Table 7.5: Descriptive statistics of CERs, PREs, RECs, and Fs on the study3 corpus for words with relation edges only in lexical cohesion analysis

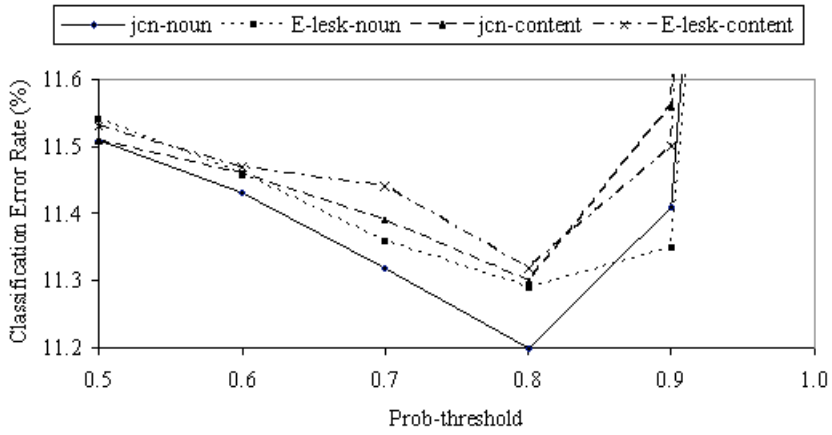


Figure 7.4: CERs under different *prob-threshold*s on the study3 corpus for words without edges in lexical cohesion analysis

When the threshold was 0.5, a word could be set to an error only when one of the word's selected alternative hypotheses was related to some other words in certain way in the document. When the threshold was 1.0, all words without edges were set to errors regardless of their hypotheses. When the threshold was between these two values, both alternative hypotheses and word *prob-threshold* played role.

Figure 7.4 shows the changes of CER with different *prob-threshold*s by *jcn* and *E-lesk* measures under both noun words and content words settings. Each line represents one combination setting of the measure and word setting. When *prob-threshold* was 0.5, there was little change in the CER on all settings. When *prob-threshold* was 1.0, the CERs greatly increased and did not show on the figure: under noun words setting, *jcn* and *E-lesk* achieved a CER of 14.07% and 13.83% separately; and under content words setting, *jcn* and *E-lesk* achieved a CER of 17.28% and 16.15% separately. The results achieved when *prob-threshold* was 1.0 suggest that more words were set to incorrect labels than to correct labels. Lines in figure 7.4 clearly indicated that a threshold of 0.8 led to the best CER under all combinations of settings, and the CERs were decreasing and then increasing. Therefore, 0.8 was chosen as one *prob-threshold* for the rest of the experiments.

(a) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.5

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	11.32	5.72	11.38	5.78	11.30	5.71	11.28	5.66
PRE	72.67	13.93	72.41	13.85	71.08	11.36	72.01	11.75
REC	37.75	16.14	37.79	16.14	38.57	15.82	38.26	15.91
F	47.55	15.84	47.50	15.76	48.26	15.46	47.99	15.29

(b) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.8

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	11.01	5.86	11.12	5.70	11.09	6.13	11.07	5.74
PRE	72.81	12.63	72.46	12.02	69.81	10.22	69.84	9.67
REC	43.08	13.98	41.43	14.28	46.33	13.60	44.41	13.77
F	52.35	12.68	50.97	13.05	54.34	11.55	52.99	11.97

Table 7.6: Performance of lexical cohesion analysis by WordNet-based measures on the study3 corpus under two selected *prob-threshold*s

Combination of All Three Groups of Words

The CERs, PREs, RECs, and F measures achieved by the combination of all the above three groups of word are shown in Table 7.6. Results from two *prob-threshold*s, 0.5 (Table 7.6(a)) and 0.8 (Table 7.6(b)), are reported. The 0.5 *prob-threshold* represented the baseline threshold under which word probabilities did not play roles for words without edges. The 0.8 *prob-threshold* was the threshold under which the best CERs were achieved by all measures and settings for words without edges.

Since words in the three groups are disjoint, the overall performance is simply the aggregation of the performance of the individual groups. Under both *prob-threshold*s, the difference in performance between different measures in both word choice settings was small. However, there was some improvement in CER and F over those of the CSSYN under both thresholds, particularly the 0.8 threshold.

Incorporating the *LSA-based* Measure

Performances reported thus far were achieved by using WordNet-based measures alone. Figure 7.5 shows the CERs achieved by integrating the *LSA-based* measure with WordNet-based measures. When both *LSA-based* and WordNet-based measures were integrated, selected words could be connected through relations determined either by a WordNet-based measure or by a *LSA-based* measure. Words were only classified into the no edge group when they were not related to any other words by any of the possible relations.

Two *prob-threshold*s were continued to be used in the following experiment. Figure 7.5(a) presents the

variations in CER with different thresholds of cosine similarity values when the *prob-threshold* is 0.5, and Figure 7.5(b) plots CERs for different cosine similarity values when the *prob-threshold* is 0.8. In both figures, there are four lines, denoting four types of combination of WordNet measures and word choice settings: noun–*jcn*, noun–*E-lesk*, content–*jcn*, and content–*E-lesk*.

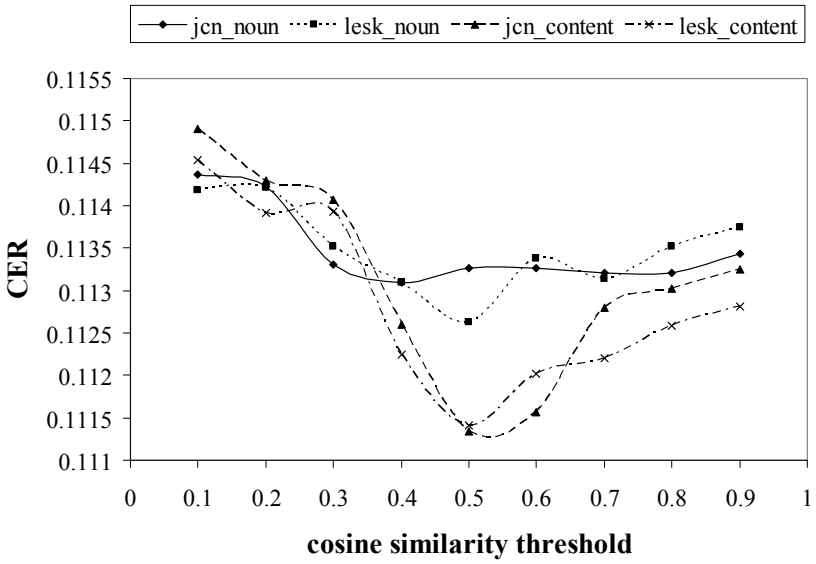
It can be observed from both figures, when *cosine-threshold* is increasing, CER decreases then increases. When the *cosine-threshold* is set too low, more words may relate to one another, and more noise will be introduced. When the *cosine-threshold* is set too high, too few relations will exist between words under the *LSA-based* measure, and the CER will be similar to that achieved without use of it. After integrating the *LSA-based* measure with WordNet-based measures, in general when *cosine-threshold* was larger than 0.3, more gains were obtained when all content words were used than when only nouns were used. The only exception is the combination of *jcn* and *LSA-based* measure when *prob-threshold*s was 0.8, under which when *cosine-threshold* was too high noun words setting had the better CER than content words setting did. These results demonstrate that distributed measure could detect useful semantic relations across grammatical categories.

Table 7.7 shows the CERs, PREs, RECs, and F measures at the *cosine-threshold*s when most of the CER lines reach their lowest points. Figure 7.5 indicates the *cosine-threshold* of 0.5 for *prob-threshold* 0.5 and the *cosine-threshold* of 0.6 for *prob-threshold* 0.8. Compared to CERs presented in Table 7.6, more improvement in CER was achieved under the content words setting than that achieved under the noun words setting. In addition, the comparisons of PREs, RECs and F measures to the corresponding results in Table 7.6 show that F measures stayed almost the same, while PRE increased and REC decreased.

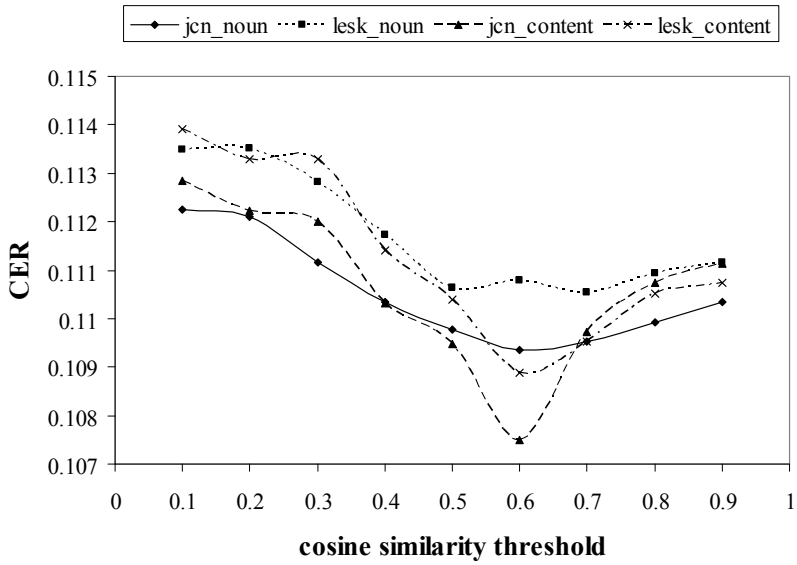
Performance of the *LSA-based* Measure

Under the content words setting, the combination of the *LSA-based* measure and WordNet-based measures helped to improve the CERs and precision of error detection over those achieved by WordNet-based measures alone. To verify if the improvement indeed resulted from the combination, the performance of using *LSA-based* measure alone was evaluated. To this end, the performance of the *LSA-based* measure on different word choice settings under the same threshold settings as presented in Table 7.6(b) were evaluated. For words with repeated edges, same performance was achieved by *LSA-based* measure as that of WordNet measures, which can be referred back to Table 7.4.

Table 7.8 shows the CERs, PREs, RECs, and F measures of the *LSA-based* measure for the combination of all groups of words and for the two individual groups. *NoEdge* is the word group in which words have no



(a) *prob-threshold* is 0.5



(b) *prob-threshold* is 0.8

Figure 7.5: CERs of lexical cohesion analysis on the study3 corpus when integrating WordNet-based measures and *LSA-based* measure with varied cosine similarity thresholds

(a) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.5 and *cosine-threshold* is 0.5

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	11.33	5.66	11.26	5.56	11.13	5.71	11.14	5.55
PRE	72.89	13.38	73.15	13.11	74.91	12.68	74.58	12.00
REC	37.08	16.19	37.29	16.02	37.45	15.83	37.37	15.60
F	46.96	15.73	47.26	15.63	47.84	15.28	47.76	15.10

(b) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.8 and *cosine-threshold* is 0.6

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	10.94	5.76	11.08	5.64	10.75	5.90	10.89	5.68
PRE	73.3	11.89	72.87	11.58	74.15	12.60	72.87	11.98
REC	42.66	13.91	40.93	14.08	45.53	14.09	43.84	14.12
F	52.23	12.50	50.69	12.82	54.86	12.48	53.39	13.02

Table 7.7: Performance of lexical cohesion analysis on the study3 corpus when integrating WordNet measures with *LSA-based* measure under two selected *prob-threshold*s and *cosine-threshold*s

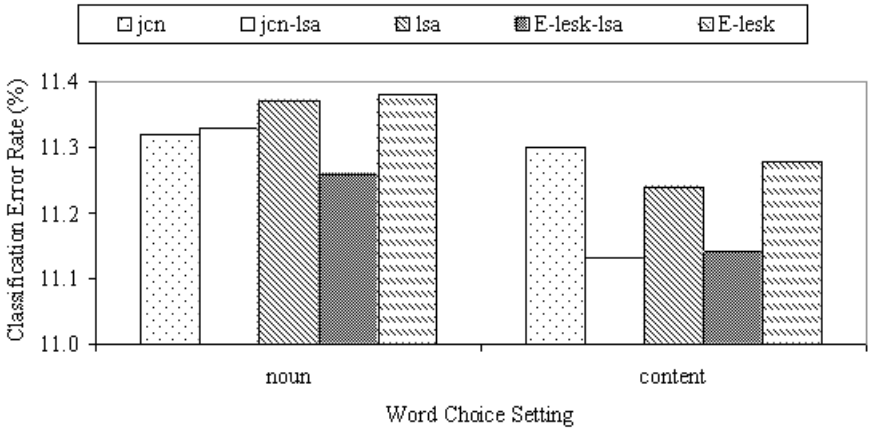
edges. Correspondingly, *Edge* denotes words with relation edges only. *ALL* represents the combination of all three groups of words. As shown in Table 7.8, the *LSA-based* measure by itself can achieve comparable performance to that achieved through measures based on WordNet.

Do WordNet-based measures and *LSA-based* measure provide complementary information?

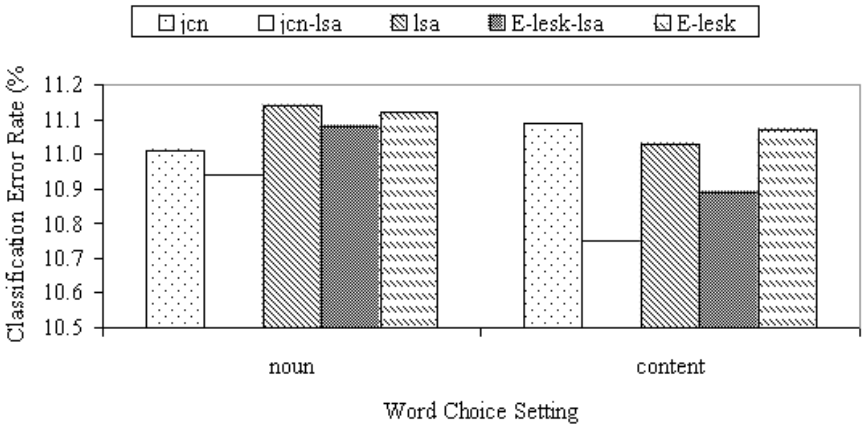
Figure 7.6 shows the CERs achieved by WordNet-based measures, *LSA-based* measure, and their combinations under two selected threshold settings for both noun words and content words. The combinations of WordNet-based measures and the *LSA-based* measure were annotated by connecting their names with a dash ("-"). For example, *jcn-lsa* stands for the combination of the *jcn* measure and the *LSA-based* measure. Because two WordNet-based measures (*jcn* and *E-lesk*) and one *LSA-based* measure were used, five bars were used to represent the performance of individual measures and their combinations, specifically *jcn*, *jcn-lsa*, *lsa*, *E-lesk-lsa*, and *E-lesk*. These five bars corresponded to two comparisons: one was *jcn-lsa*, *jcn*, *lsa*; and the other was *E-lesk-lsa*, *E-lesk*, *lsa*.

Under both threshold settings the combination of WordNet-based measures and *LSA-based* measure had a relatively bigger improvement on CER for content words than for noun words, which is evidenced from Figure 7.6(a) and Figure 7.6(b).

To find out if the improvement in CER achieved by the combination of WordNet-based measures and the *LSA-based* measure over those achieved by individual measure alone were significant or not, repeated



(a) *prob-threshold* is 0.5, and *cosine-threshold* is 0.5



(b) *prob-threshold* is 0.8, and *cosine-threshold* is 0.6

Figure 7.6: CERs of lexical cohesion analysis on the study3 corpus by WordNet-based measures, *LSA-based* measure, and their combinations under two selected threshold settings for both noun words and content words

(a) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.5 and *cosine-threshold* is 0.5

	Nouns						Content Words					
	NoEdge		Edge		All		NoEdge		Edge		All	
	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)
CER	11.54	5.71	11.49	5.70	11.37	5.65	11.49	5.77	11.44	5.58	11.24	5.64
PRE	68.77	15.18	70.74	14.15	71.46	13.10	69.37	14.83	70.67	14.01	72.54	12.21
REC	37.20	16.87	36.60	16.26	36.81	16.59	38.08	16.17	36.27	16.59	37.18	16.19
F	46.50	16.86	46.25	16.27	46.54	16.44	47.44	15.68	45.96	16.90	47.17	15.76

(b) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.8 and *cosine-threshold* is 0.6

	Nouns						Content Words					
	NoEdge		Edge		All		NoEdge		Edge		All	
	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)
CER	11.35	5.86	11.46	5.72	11.14	5.82	11.26	6.01	11.46	5.66	11.03	5.92
PRE	69.29	11.88	70.90	14.17	71.68	11.80	68.67	13.96	70.55	14.23	70.61	12.89
REC	43.14	13.76	36.98	16.56	43.14	13.76	46.56	13.52	36.79	16.89	46.24	13.92
F	54.44	12.58	46.58	16.52	52.15	12.31	51.86	12.62	46.29	17.10	54.64	12.73

Table 7.8: Performance of lexical cohesion analysis on the study3 corpus when using *LSA-based* measure alone under two selected *prob-threshold*s and *cosine-threshold*s

measure analyses were conducted for the content words setting. CER was the dependent variable for repeated measure analyses. The independent variable for each repeated measure analysis was the measures it used, and had three levels. The values of the independent variables corresponded to the two comparisons introduced above. For each possible value of the independent variable, two threshold settings were considered, which resulted in four repeated measure analyses in total. Besides the main effect of the independent variable, the multiple comparisons were also conducted to compare the difference in CER between different measures.

Table 7.9 shows the analysis results, in which the numbers in the second column represent the threshold setting, and the first and second numbers represent the *prob-threshold* and the *cosine-threshold*, respectively.

Results from all four repeated measure analyses revealed that there were no significant changes in CER when different measures were used. However, the multiple comparison results showed some difference. *E-lesk-lsa* did not significantly improve the CERs achieved by *E-lesk* and *lsa* alone under both threshold settings. *jcn-lsa* significantly improved the CER achieved by *lsa* alone ($p < 0.05$) and weakly improved the CER achieved by *jcn* alone ($p = 0.088$) under the threshold setting with *prob-threshold* of 0.8 and *cosine-threshold* of 0.6. Under the threshold setting with *prob-threshold* of 0.5 and *cosine-threshold* of 0.5, *jcn-lsa* significantly improved the CER achieved by *jcn* alone ($p < 0.05$).

The effects of the combination of *jcn* and *lsa* on individual measures is different from those of the combination of *E-lesk* and *lsa*. One possible explanation is that *jcn* can only measure the semantic relatedness

Level of variables	Threshold	Main effect		Multiple comparisons		
		$F(2, 22)$	p		Mean difference	p
jcn-lsa, jcn, lsa	0.5, 0.5	2.656	0.122	jcn-lsa to jcn	-0.002	0.042 *
				jcn-lsa to lsa	-0.001	0.258
	0.8, 0.6	3.538	0.069	jcn-lsa to jcn	-0.003	0.088
				jcn-lsa to lsa	-0.003	0.027 *
E-lesk-lsa, E-lesk, lsa	0.5, 0.5	0.826	0.451	E-lesk-lsa to E-lesk	-0.001	0.384
				E-lesk-lsa to lsa	-0.001	1.000
	0.8, 0.6	0.614	0.550	E-lesk-lsa to E-lesk	-0.002	0.670
				E-lesk-lsa to lsa	-0.001	1.000

* Significant at 0.05 level, Bonferroni adjustment

Table 7.9: Repeated measure analysis results of CERs between the combination of semantic relatedness measures and individual semantic relatedness measures on the study3 corpus under content word setting and two selected *prob-threshold*s and *cosine-threshold*s

	CER(%)	Pre (%)	Rec (%)	F (%)
noun	8.21	86.48	37.98	51.46
content	6.83	90.42	48.25	62.04

Table 7.10: Upperbounds of CERs, PREs, RECs, and Fs of lexical cohesion analysis on the WSJ corpus under both noun words and content words settings

between words within the same grammatical categories, specifically between nouns or between verbs, and *jcn* cannot measure semantic relatedness between words belonging to different grammatical categories. However, *lsa* is able to measure the semantic relatedness between any pair of words, which could provide additional information to *jcn*. *E-lesk* is a WordNet-based measure that can measure word relatedness between words each belonging to different grammatical category, and thus the *lsa* may provide a certain level of redundant information to *E-lesk*. Further evidence is that all three measures produced similar results on nouns and their combinations did not improve the CERs.

7.3.3 Experiment Result on the WSJ Corpus

Upperbound

Same as on the study3 corpus, Table 7.10 reports the upperbounds of CER, PRE, REC and F measure for lexical cohesion analysis on the WSJ corpus.

Words with Repeated Edges

Similar to the results on the study3 corpus, all the measures on the WSJ corpus shared the same performance on words with repeated edges. The CERs, PREs, RECs, and Fs after performing the majority vote on those

	Nouns		Content Words	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	10.74	3.86	10.69	3.86
PRE	69.72	17.79	70.56	17.21
REC	15.64	9.33	15.64	9.33
F	24.21	10.70	24.26	10.68

Table 7.11: Descriptive statistics of CERs, PREs, RECs, and Fs on the WSJ corpus for words with repeated edges in lexical cohesion analysis

words are reported in Table 7.11. The performance improvement, if any, was small.

Words with Relation Edges only

When processing words with relation edges only, neither *jcn* nor *E-lesk* affected error detection performance. The achieved performance, specifically CER, PRE, REC, and F was the same as that of the CSSYN.

Words with no Edge

Figure 7.7 shows the changes of CER with different *prob-threshold*s by *jcn* and *E-lesk* measures under both noun words and content words settings. When *prob-threshold* was 0.5, there was a very small improvement in CER over the corresponding results of CSSYN in Chapter 5. When *prob-threshold* was 1.0, more words were wrongly labeled as errors and CERs increased greatly and did not show on the figure: under noun words setting, *jcn* and *E-lesk* achieved a CER of 21.99% and 20.65% separately; and under content words setting, *jcn* and *E-lesk* achieved a CER of 30.06% and 27.36% separately.

Similar to that observed for CER on the study3 corpus, a decreasing and then increasing trend is also observed on the WSJ corpus. Figure 7.7 reveals that the best CER across both measures and across both nouns and content words settings was achieved when the *prob-threshold* was set to 0.7. Therefore, 0.7 was chosen as one *prob-threshold* for the rest of experiments. The CERs achieved by *jcn* and *E-lesk* on both word choice settings were similar under *prob-threshold* of 0.7.

Combination of All Three Groups of Words

Table 7.12 shows the CERs, PREs, RECs, and Fs achieved by processing all three groups of words together. Same as do for the study3 corpus, results from two *prob-threshold*s are reported, one *prob-threshold* is 0.5 (Table 7.12(a)) and the other *prob-threshold* is 0.7 (Table 7.12(b)). Because words in three groups are disjointed and the process of words with relation edges did not make any changes on the performance, results

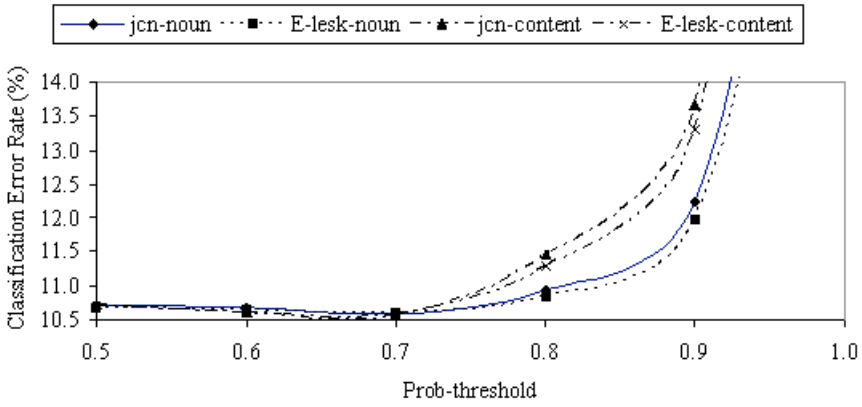


Figure 7.7: CERs under different *prob-threshold*s on the WSJ corpus for words without edges in lexical cohesion analysis

in Table 7.12 in fact were the joint performance achieved on the other two word groups. When the *prob-threshold* was 0.7, both *jcn* and *E-lesk* achieved better F measures under content words setting than those under noun settings. Comparisons of the PREs and RECs between two word choice settings reveal that RECs were higher under content words setting and PREs were higher under noun words setting for both *jcn* and *E-lesk* when the *prob-threshold* was 0.7.

Incorporating the *LSA-based* Measure

Figure 7.8 shows the CERs achieved by integrating WordNet-based measures with the *LSA-based* measure. Following integration, the selected words were connected by the WordNet-based measures, the *LSA-based* measure, or both. Words belonged to the no edge group only when they did not share any possible relation to any other words.

The two selected *prob-threshold*s were kept in the experiment for the integration of the Wordnet-based measures and the *LSA-based* measure. Figure 7.8(a) plots the variations of CERs with different thresholds of cosine similarity values when the *prob-threshold* is 0.5. Figure 7.8(b) presents CERs for different cosine values when the *prob-threshold* is 0.7. There are four lines in each figure, each denoting one kind of combination of the WordNet measures and word choice settings.

Compared to Figure 7.5(a) and Figure 7.5(b), the trend of the lines in Figures 7.8(a) and 7.8(b) is somewhat different from that of the study3 corpus. In both Figures 7.8(a) and 7.8(b), when the *cosine-threshold* is increasing, the general trend is that CERs continue to drop and reach the lowest point at very high *cosine-threshold*s. However, the general trend of CERs on the study3 corpus is decreasing then increasing with the

(a) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.5

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	10.70	3.90	10.68	3.88	10.69	3.82	10.65	3.84
PRE	69.33	16.71	70.98	16.16	66.56	13.40	71.17	17.04
REC	16.53	8.93	16.26	8.98	16.76	8.93	16.56	9.49
F	25.46	10.12	25.14	10.10	25.69	10.22	25.43	10.56

(b) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.7

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	10.59	3.96	10.59	3.94	10.52	3.76	10.54	3.82
PRE	67.84	11.96	69.28	12.20	64.68	12.94	67.41	14.64
REC	19.64	8.23	18.95	8.83	22.14	8.72	21.24	9.86
F	29.33	8.58	28.37	9.46	31.97	9.00	30.83	10.16

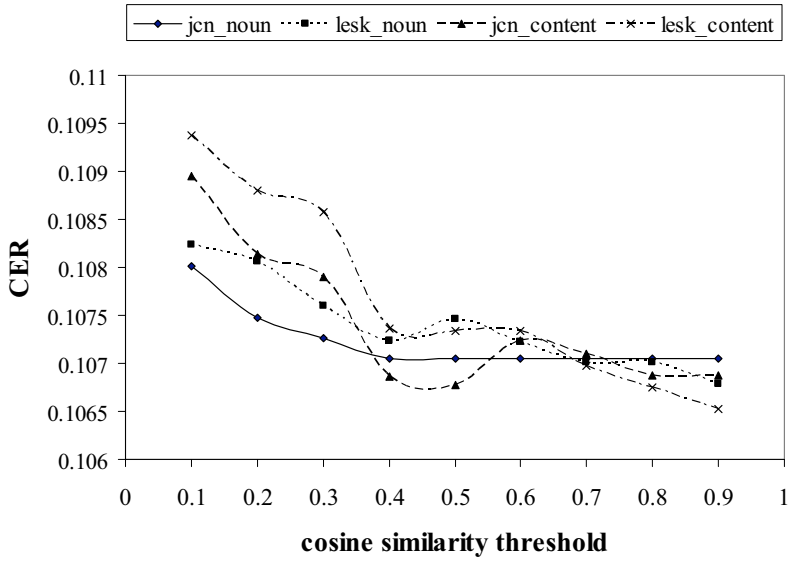
Table 7.12: Performance of lexical cohesion analysis by WordNet-based measures on the WSJ corpus under two selected *prob-threshold*s

	Nouns				Content Words			
	Jcn		E-lesk		Jcn		E-lesk	
	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)	Mean (%)	Std.(%)
CER	10.56	3.93	10.59	3.89	10.49	3.73	10.53	3.77
PRE	67.98	11.75	68.08	11.38	64.81	12.79	66.35	13.50
REC	19.64	8.23	18.95	8.83	22.14	8.72	21.24	9.86
F	29.38	8.67	28.38	9.59	32.02	9.08	30.83	10.29

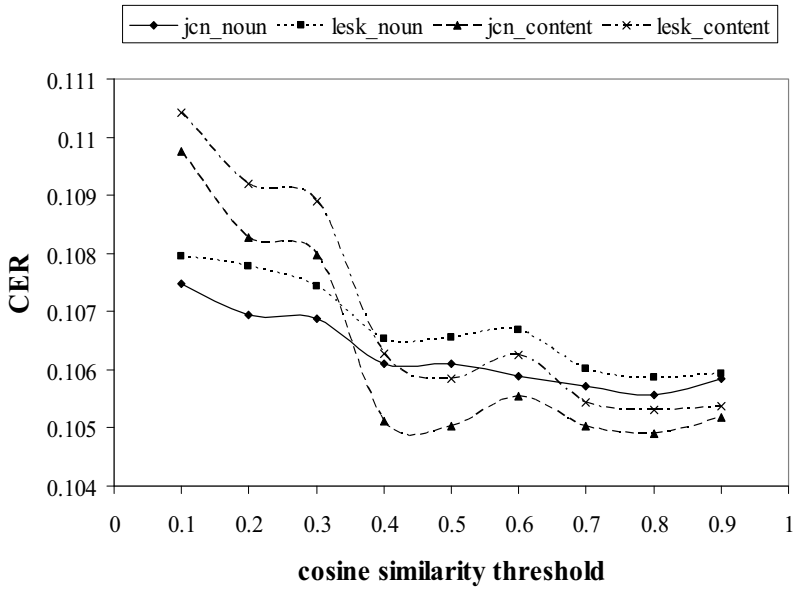
Table 7.13: Performance of lexical cohesion analysis on the WSJ corpus when integrating WordNet measures with *LSA-based* measure under *prob-threshold* of 0.7 and *cosine-threshold* of 0.8

increase of *cosine-threshold*s. When the *prob-threshold* was set to 0.5, the best CERs were achieved when the *cosine-threshold* was 0.9. When the *prob-threshold* was set to 0.7, the best CERs were achieved when the *cosine-threshold* was 0.8.

Integration of the *LSA-based* measure with WordNet-based measures did not provide further improvement over that achieved by the WordNet-based measures, which can be observed from the fact that best performance was achieved under the high *cosine-threshold*. When the *prob-threshold* was set to 0.5 and *cosine-threshold* was set to 0.9, there was no performance change when compared to the performance reported in Table 7.12. When the *prob-threshold* was set to 0.7 and *cosine-threshold* was set to 0.8, there were only small changes in performance, as shown in Table 7.13.



(a) *prob-threshold* is 0.5



(b) *prob-threshold* is 0.7

Figure 7.8: CERs of lexical cohesion analysis on the WSJ corpus when integrating WordNet-based measures and *LSA-based* measure with varied cosine similarity thresholds

(a) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.5 and *cosine-threshold* is 0.9

	Nouns				Content Words			
	NoEdge		All		NoEdge		All	
	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)
CER	10.72	3.88	10.70	3.87	10.72	3.82	10.66	3.80
PRE	70.08	17.80	70.34	17.72	66.79	14.50	67.86	13.90
REC	16.00	9.20	16.00	9.20	16.24	9.22	16.24	9.22
F	24.74	10.54	24.76	10.54	24.94	10.65	25.02	10.63

(b) Descriptive statistics of CERs, PREs, RECs, and Fs when *prob-threshold* is 0.7 and *cosine-threshold* is 0.8

	Nouns				Content Words			
	NoEdge		All		NoEdge		All	
	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)	Mean (%)	Std. (%)
CER	10.63	3.96	10.62	3.95	10.59	3.79	10.53	3.78
PRE	67.18	13.37	67.40	13.32	63.58	13.73	64.43	13.74
REC	19.48	8.46	19.48	8.46	21.98	9.09	21.98	9.09
F	29.02	8.87	29.04	8.87	31.62	9.54	31.72	9.52

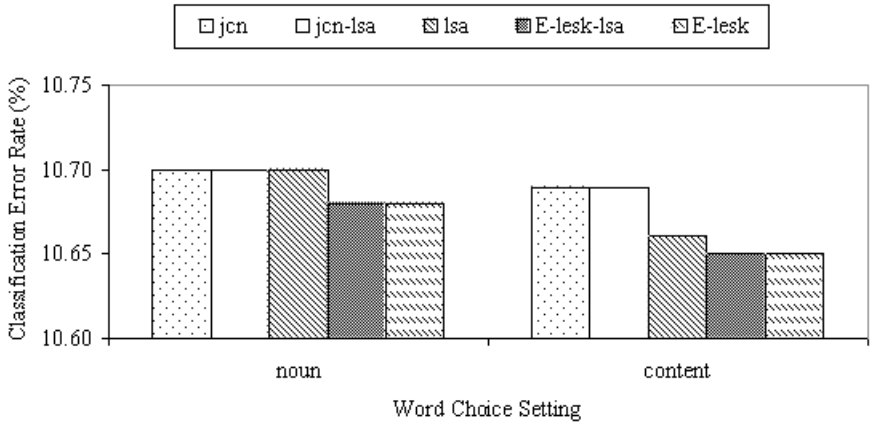
Table 7.14: Performance of lexical cohesion analysis on the WSJ corpus when using *LSA-based* measure alone under two selected *prob-threshold*s and *cosine-threshold*s

Performance of the *LSA-based* Measure

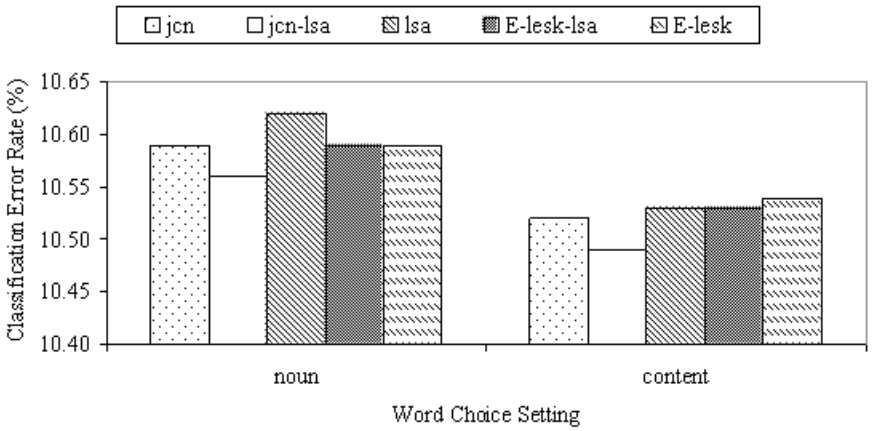
Although the incorporation of the *LSA-based* measure with the WordNet-based measures did not improve the performance of WordNet-based measures, evaluation of the performance of the *LSA-based* measure is still valuable. Table 7.14 shows the performance of the *LSA-based* measure on words without edges and on all words in three groups under the threshold settings which yielded the best combination performance. For words with repeated relations, the performance of the *LSA-based* measure was the same as that of WordNet measures. Likewise, for words with relation edges only, the performance of the *LSA-based* measure was the same as those of *jcn* and *E-lesk*, indicating there was no change in performance when compared to CSSYN. Table 7.14 shows that under both word choice settings and threshold settings the *LSA-based* measure alone can achieve performances comparable to those achieved by WordNet-based measures reported in Table 7.12.

Do WordNet-based measures and *LSA-based* measure provide complementary information?

Figure 7.9 shows the CERs achieved by WordNet-based measures, *LSA-based* measure, and their combinations under two selected threshold settings for both noun words and content words. It is evidenced from Figure 7.9(a) and Figure 7.9(b) that the combination of WordNet-based measures and *LSA-based* measure did not bring improvement on CER on almost all settings. The only exception is the *jcn-lsa* when *prob-threshold* is 0.7 and *cosine-threshold* is 0.8.



(a) *prob-threshold* is 0.5, and *cosine-threshold* is 0.9



(b) *prob-threshold* is 0.7, and *cosine-threshold* is 0.8

Figure 7.9: CERs of lexical cohesion analysis on the WSJ corpus by WordNet-based measures, *LSA-based* measure, and their combinations under two selected threshold settings for both noun words and content words

			LSA								
	jcn	E-lesk	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
noun	10.75	10.75	10.70	10.70	10.72	10.72	10.72	10.72	10.75	10.75	10.75
content	10.75	10.69	10.61	10.61	10.70	10.72	10.72	10.72	10.75	10.75	10.75

Table 7.15: Upperbounds of CERs by lexical cohesion analysis on words with relation edges only on the WSJ corpus

Explanations for the Lack of Effect on Words with Relation Edges Only

All the measures tested failed to produce any performance change on words with relation edges only. To look for explanations, the constructed word graph for lexical cohesion analysis was investigated. The guidance for the method to process words with only relation edges was that a word was designated as correct if the word was related to several words which were correct words with high probabilities. Therefore, the method took effect only on words that were related to other words but were wrongly labeled as errors. The upperbounds of this method were then computed by setting all mistakenly labeled errors that were non-singleton as correct, and the upperbounds of CER were reported in Table 7.15. The CERs in the Table 7.15 reveal that if a word with true label "correct" was related to some other words by *jcn* relations, the word was never labeled as an error (10.75 is the CER by CSSYN) by the method introduced in Chapter 5. The same finding held for the *LSA-based* measure on both word choice settings under high *cosine-threshold*s and for *E-lesk* on noun words. For most of the settings, the upperbounds only slightly differed from the baseline.

For the *LSA-based* measure, it can be observed from Table 7.15 that the upperbounds of CER increased with the increase of the *cosine-threshold*. When the *cosine-threshold*s of the *LSA-based* measure were low, more words could be related to other words because more relations could be incorporated, and then more truly correct words that related to other words could be mistakenly labeled as errors.

7.4 Summary

Lexical cohesion analysis can help improve error detection. WordNet-based measures and *LSA-based* measure deal with reiteration and collocation relations respectively. All individual measures showed comparable performances on each data corpus, while the combination of WordNet-based measures and *LSA-based* measure had a different effect on the two corpora. For the study3 corpus, the integration of *jcn* and *LSA-based* measure led to additional improvement when all content words were used and did not cause additional improvement when only nouns were used. Therefore, the *LSA-based* measure could provide useful collocation

	noun	verb	adjective
study3	19.99	8.33	3.43
WSJ	30.05	9.22	4.62

Table 7.16: Distribution of content words in both the study3 corpus and the WSJ corpus when lexical cohesion analysis was used

information across the parts-of-speech. For the WSJ corpus, the integration of measures did not improve the performance on either nouns or content words.

Similar as in Chapter 6, one alternative explanation for the above difference is the different composition of parts-of-speech of content words in the two corpora. Table 7.16 shows the percentage of each part-of-speech group that had been used in lexical cohesion analysis. The percentages in Table 7.16 are smaller than those in Table 6.5 because some words are stop words and some verbs/adjectives are not related to any noun through any of the concerning relations. Compared to the results shown in Table 6.5, the ratio between nouns and verbs/adjectives increases from 1.25 to 2.17 for the WSJ corpus, and is still higher than that of the study3 corpus which increases from 0.99 to 1.70.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Summary

This dissertation investigates the effect of syntactic and semantic information on the performance of speech recognition error detection in monologue applications, specifically, in spontaneous dictation. New methods were proposed to extract and employ syntactic and semantic information for error detection. The proposed methods were evaluated with two spontaneous dictation corpora, which differed in both style and topic. One corpus concerns daily correspondence in the office environment, and the other follows the topic and the style of Wall Street Journal. The methods used to recognize the two speech corpora were also different: one was inline recognition, and the other was offline recognition.

Two approaches were designed to incorporate linguistic information into error detection. The first one was a feature combination based confidence measure, in which Support Vector Machines was used as the classifier. Linguistic features were incorporated as part of the input to the Support Vector Machines. The second approach focused on analyzing the lexical cohesion of text and encoding the semantic information through lexical relations among words. Because lexical cohesion analysis ignores sentence syntactic structures, predictions of the confidence measure that had incorporated syntactic features were used to initialize probabilities of words in the analysis. To evaluate the overall effect of syntactic and semantic information on error detection, a confidence measure that combined non-linguistic features was built to serve as the baseline.

Syntactic features were extracted from the output of parsers. Lexicalized syntactic parsers were selected due to their natural fit to the task of detecting misrecognized words. Two sets of syntactic features were used. The first set concerns whether a word had any links to other words in the same sentence. The other set is

related to word associations based on the syntactic dependency between words. Experiment results showed that neither sets of syntactic features alone could improve the performance of error detection. However, when combined with non-linguistic features, the syntactic features improved the performance of error detection. In addition, even if these two sets of syntactic features rely on different information sources, they provide a certain degree of redundant information and their combination did not lead to significant improvement in the performance of error detection.

WordNet is a lexical database that provides information for determining the semantic relatedness between words. Five measures (*jcn*, *res*, *lin*, *E-lesk*, and *vector*) based on WordNet were used, and they were grouped into two categories (corpus-enhanced measures and gloss-based measures) based on the kinds of information being exploited. Since a word may relate to many other words in same document, three statistics representing the semantic relatedness of a word to its surrounding words were selected as features: maximum, average, and the average of the top three. The surrounding words were constrained by window size. Experiment results show that a larger window size achieved a better performance, and the choice of window size may vary with the data. The performances achieved by these measures were comparable with some slight variation, and *jcn* and *E-lesk* were two measures yielding relatively better performance. Improvements achieved through use of semantic features were generally consistent despite being small.

Lexical cohesion analysis is the second type of technique selected to model the semantic relatedness among words. Instead of encoding semantic information into several features, a word graph was build for the speech recognition output. Words were linked through certain relations. Semantic relations were extracted based on two types of measures: WordNet-based and corpus-based. *jcn* and *E-lesk* are WordNet-based measures and concern reiteration relations. The latent semantic analysis-based measure is corpus-based measure and concern collocation relations. Words in a graph were processed separately based on their states: having no edges, having repeated edges, or having relational edges. Two threshold parameters were evaluated: prob-threshold and cosine threshold. Experiment results show that setting the prob-threshold higher than the default value of 0.5 for words with no edges improved the performance. The best prob-thresholds for the study3 corpus and WSJ corpus were 0.8 and 0.7 separately, both of which were not too high as 1.0 or 0.9 and also not too low as 0.5 or 0.6. Therefore, a prob-threshold in the middle range of higher values may be a good choice. Changing the cosine threshold for the LSA-based measure also had a large impact on performance, and the choice of cosine threshold was corpus dependent. For the study3 corpus, the selected cosine thresholds for two prob-thresholds were 0.5 and 0.6; but for the WSJ corpus, the selected cosine thresholds were 0.9

and 0.8. The cosine thresholds for one corpus were relatively stable. Individual measures showed comparable error detection performance when used alone. When content words were considered, the combination of *jcn* and LSA-based measures outperformed the individual measures on the study3 corpus, but the combination of any of WordNet-based measures and the LSA-based measure did not improve the performance of individual measures on the WSJ corpus. One possible explanation for the difference in performance is the different distribution of parts-of-speech in the two corpora. Another possible explanation is the topic mismatch between training corpus and evaluation corpus. The study3 corpus only has one topic, which is also one of the topics in the training corpora. However, the WSJ corpus contains more topics, which may not appear in the training corpus given that the journalists were allowed to dictate on any topic they want.

In previous chapters, performance of error detection when utilizing syntactic and semantic knowledge was reported. In this section, statistical analyses were conducted to test whether improvements achieved by the addition of linguistic knowledge in error detection were significant or not. Because of the incremental integration of linguistic knowledge in our proposed methods, three levels of knowledge conditions were compared during the statistical analyses: 1) only non-linguistic knowledge was used, 2) both non-linguistic and syntactic knowledge were used, and 3) non-linguistic, syntactic, and semantic knowledge were all used. Corresponding to the two approaches incorporating semantic information, two sets of statistical analyses were conducted and they differed in the third knowledge condition only. Specifically, one set of statistical analyses compared error detection performance by confidence measure combing non-linguistic features, confidence measure combining non-linguistic and syntactic features, and confidence measure combining non-linguistic, syntactic and semantic features; the other set of statistical analyses compared error detection performance by confidence measure combing non-linguistic features, confidence measure combining non-linguistic and syntactic features, and lexical cohesion analysis.

Because error detection performances under each knowledge condition were evaluated on the same corpora using the same leave-one-out schema, repeated measure analyses were selected as the method for statistical analyses. In each set of the analyses, separate analysis was conducted for each of the performance metrics, including CER, F, PRE, and REC, which served as dependent variables in the analyses. The independent variable was the knowledge condition for error detection and consisted of three levels. The significant level was set to 0.05. When reporting the main effect of knowledge condition, if the Sphericity assumption was violated, the significant level achieved by Greenhouse-Geisser adjustment was used. When conducting multiple comparisons between different knowledge conditions, Bonferroni adjustment was used.

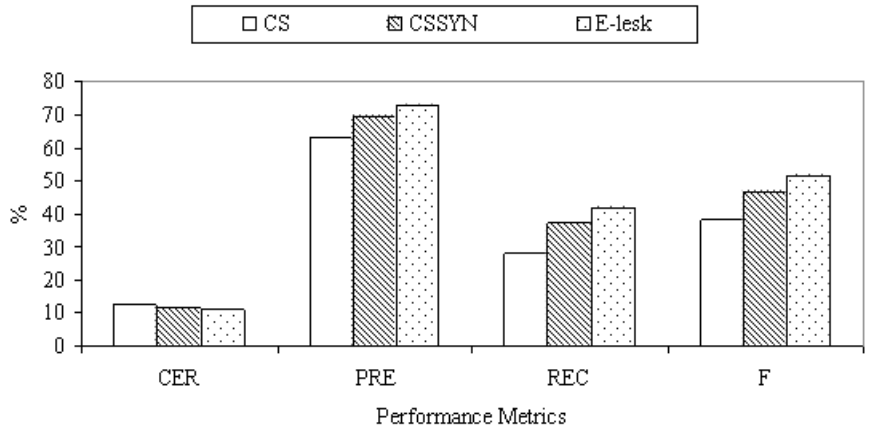


Figure 8.1: Performance of confidence measures combining different kinds of knowledge on the study3 corpus

8.1.1 Statistical Analyses of Error Detection Performance by Confidence Measures Incorporating Non-linguistic, Syntactic, and Semantic knowledge

For the confidence measure incorporating non-linguistic knowledge, feature set *CS* was its representative. For the confidence measure incorporating non-linguistic and syntactic knowledge, feature set *CSSYN* included all syntactic features and was the corresponding representative. For the confidence measure incorporating non-linguistic, syntactic, and semantic knowledge, feature set having the best performance was used as the representative. For the study3 corpus, *E-lesk* under the content words setting was selected; for the WSJ corpus, *jcn* under the noun words setting was selected.

Study3 corpus

Figure 8.1 shows the difference in CER, PRE, REC and F measures between *CS*, *CSSYN* and *E-lesk*. With the incremental incorporation of syntactic and semantic knowledge, CER decreased, and PRE, REC and F measure increased. When compared to *CS*, *E-lesk* under content words setting improved the F measure from 30.08% to 51.54% with a 71.34% relative improvement, and also improved the CER from 12.57% to 11.07% with a 10.11% relative improvement. Both the improvements were significant as shown in the following statistical analyses.

Table 8.1 reports repeated measure analyses results, in which Table 8.1(a) reports the main effect of knowledge condition, and Table 8.1(b) reports the multiple comparisons results.

Analyses results revealed that knowledge condition did significantly affect the CER, F measure, and REC.

(a) Main effect of knowledge condition

	CER	F	REC	PRE
$F(2, 22)$	11.783	19.167	24.080	3.234
p	0.000 *	0.000 *	0.000 *	0.092

* Significant at 0.05 level

(b) Multiple comparisons between knowledge conditions (Bonferroni adjustment)

		CER	F	REC	PRE
CS CSSYN	mean difference	1.029	-8.439	-9.150	-6.647
	p	0.064	0.018 *	0.011 *	0.371
CS - E-lesk	mean difference	1.500	-13.468	-13.841	-9.756
	p	0.001 *	0.000 *	0.000 *	0.252
CSSYN - E-lesk	mean difference	0.470	-5.029	-4.691	-3.109
	p	0.250	0.033 *	0.018 *	0.420

* Significant at 0.05 level

Table 8.1: Repeat measure analyses results on knowledge condition (*CS*, *CSSYN*, and *E-lesk*) for CER, F, REC, and PRE on the Study3 corpus

The comparison between *CS* and *CSSYN* showed that the combination of *CS* features and all syntactic features significantly improved the F measure and REC over those achieved by *CS* features alone. It highlights that syntactic features can help finding more errors.

The comparison between *CSSYN* and *E-lesk* also showed that the addition of semantic features significantly improved the F measure and REC over those achieved by *CSSYN* features alone, which shows that the usage of semantic information on top of syntactic information can further help finding more errors.

The comparison between *CS* and *E-lesk* shows us the total effect of linguistic information. Similarly, F measure and REC were significantly improved by *E-leask* over *CS*. Although both the improvements of CER by *CSSYN* over *CS* and by *E-lesk* over *CSSYN* were not significant, the improvement in CER by *E-lesk* over *CS* was significant. Therefore, the usage of linguistic information can improve the CER.

WSJ corpus

Figure 8.2 shows the difference in CER, PRE, REC and F measures between *CS*, *CSSYN* and *jcn*. Same as pattern found on the study3 corpus, with the incremental incorporation of syntactic and semantic knowledge, CER decreased, and PRE, REC and F measure increased. When compared to *CS*, *jcn* under noun words setting improved the F measure from 16.35% to 26.29% with a 60.81% relative improvement, and also improved the CER from 11.41% to 10.52% with a 7.79% relative improvement. Both of the improvements were also significant as shown later.

Table 8.2 shows repeated measure analyses results, in which Table 8.2(a) reports the main effect of knowledge condition, and Table 8.2(b) reports multiple comparisons results between different levels of knowledge

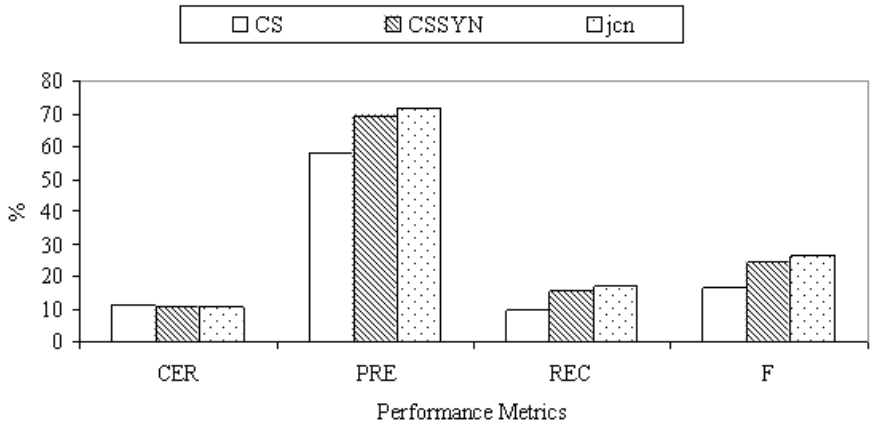


Figure 8.2: Performance of confidence measures combining different kinds of knowledge on the WSJ corpus

(a) Main effect of knowledge condition

	CER	F	REC	PRE
$F(2, 18)$	11.715	10.704	6.931	9.328
p	0.003 *	0.005 *	0.021 *	0.005 *

* Significant at 0.05 level

(b) Multiple comparisons between knowledge conditions (Bonferroni adjustment)

		CER	F	REC	PRE
CS CSSYN	mean difference	0.662	-7.843	-6.008	-11.255
	p	0.036 *	0.067	0.137	0.067
CS - jcn	mean difference	0.889	-9.940	-7.399	-13.417
	p	0.012 *	0.007 *	0.043 *	0.009 *
CSSYN - jcn	mean difference	0.227	-2.098	-1.391	-2.162
	p	0.161	0.394	0.383	1.000

* Significant at 0.05 level

Table 8.2: Repeat measure analyses results on knowledge condition (*CS*, *CSSYN*, *jcn*) for CER, F, REC, and PRE on the WSJ corpus

condition with Bonferroni adjustment.

Analyses results revealed that knowledge condition did significantly affect all the four performance metrics. The comparison results between *CSSYN* and *CS* showed that the combination of *CS* features and all syntactic features significantly improved the *CER* over that achieved by *CS* features alone. The improvement in F measure, REC and PRE by *CSSYN* over *CS* was not significant. Although the relative improvements in F measure, REC and PRE were bigger than CER, their improvement was not significant. One possible explanation was that the changes of F measure, REC and PRE between different runs were not consistent: some of runs improved them but other runs hurt them.

The comparison between *CSSYN* and *jcn* showed that the *jcn* did not significantly improve any perfor-

mance metric over CSSYN.

When compared to *CS*, *jcn* significantly improved the CER. Although improvement in F measure, REC and PRE by *CSSYN* over *CS* and that by *jcn* over *CSSYN* were not significant, *jcn* yielded significant improvement in F measure, REC and PRE over that achieved by *CS*. Therefore, the semantic information can help improve the error detection performance.

8.1.2 Comparison among Confidence Measures Incorporating Non-linguistic and Syntactic Knowledge, and Lexical cohesion analysis

Similar as the last set of analyses, feature set *CS* represented the confidence measure incorporating non-linguistic knowledge and feature set *CSSYN* represented the confidence measure incorporating non-linguistic and syntactic knowledge. For the lexical cohesion analysis, the measure combination having the best performance was used as the representative. For both the study3 corpus and the WSJ corpus, *jcn-lsa* under the content words setting was selected. Because two threshold settings were considered during lexical cohesion analysis, one set of repeated measure analyses was then conducted for each threshold setting separately.

Study3 corpus

Figure 8.3 shows the difference in CER, PRE, REC and F measures between *CS*, *CSSYN* and *jcn-lsa* under two threshold settings. The two threshold settings were the same as those used in Chapter 7, and the first number is the *prob-threshold* and the second number of the *cosine-threshold*. With the incremental incorporation of syntactic and semantic knowledge, CER decreased, and PRE, REC and F measure increased. In general, better performance was achieved by threshold setting $(0.8, 0.6)$ than by threshold setting $(0.5, 0.5)$, with the only exception PRE. When compared to *CS*, *jcn-lsa* under threshold setting $(0.8, 0.6)$ improved the F measure from 30.08% to 54.86% with a relative improvement of 82.38%, and also improved CER from 12.57% to 10.75% with a relative improvement of 14.48%. Both the improvements were significant, which is evidenced from Table 8.3(b).

Table 8.3 reports the repeated measure analyses results, in which Table 8.3(a) reports the main effect of the knowledge condition and Table 8.3(b) reports the multiple comparison results between different knowledge conditions with Bonferroni adjustment. Given that the comparison results between *CS* and *CSSYN* for the four performance metrics were the same as those in the section 8.1.1, we do not repeat them in this subsection.

Under both threshold settings, the knowledge condition affected all the four performance metrics sig-

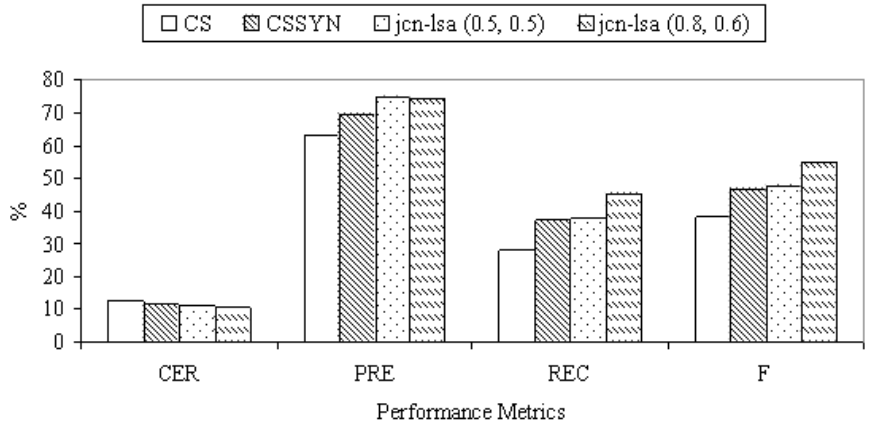


Figure 8.3: Performance of confidence measures combining non-linguistic, syntactic knowledge and lexical cohesion analysis on the study3 corpus

(a) Main effect of knowledge condition

Threshold		CER	F	REC	PRE
0.5, 0,5	$F(2, 22)$	13.242	11.883	13.402	4.879
	p	0.006 *	0.003 *	0.002 *	0.043 *
0.8, 0.6	$F(2, 22)$	20.057	22.970	33.505	3.484
	p	0.000 *	0.000 *	0.000 *	0.076

* Significant at 0.05 level

(b) Multiple comparisons between knowledge conditions (Bonferroni adjustment)

Threshold			CER	F	REC	PRE
0.5, 0,5	*CS* to *jcn-lsa*	mean difference	1.435	-9.766	-9.488	-12.000
		p	0.012 *	0.012 *	0.009 *	0.111
	CSSYN to *jcn-lsa*	mean difference	0.386	-1.382	-0.467	-5.190
		p	0.014 *	0.408	1.000	0.045 *
0.8, 0,6	*CS* to *jcn-lsa*	mean difference	1.821	-16.779	-17.563	-11.238
		p	0.004 *	0.000 *	0.000 *	0.215
	CSSYN to *jcn-lsa*	mean difference	0.772	-8.396	-8.541	-4.428
		p	0.024 *	0.008 *	0.001 *	0.417

* Significant at 0.05 level

Table 8.3: Repeat measure analyses on knowledge condition in error detection (CS, CSSYN, *jcn-lsa*) for CER, F, REC, and PRE on the Study3 corpus

nificantly, with the exception of PRE under the threshold setting with the *prob-threshold* of 0.8 and the *cosine-threshold* of 0.6.

When the *prob-threshold* was set to 0.5 and the *cosine-threshold* was set to 0.5, *jcn-lsa* improved CER and PRE significantly over those achieved by *CSSYN*. When the *prob-threshold* of 0.8 and the *cosine-threshold* of 0.6 were used, significant improvements were achieved in CER, F measure, and REC by *jcn-lsa* over those achieved by *jcn-lsa*. The improvement in F measure mainly came from the significant improvement in REC. These findings evidence that semantic information exploited through lexical cohesion analysis can provide additional information to CS related and syntactic knowledge in error detection task, especially when thresholds were properly set.

When compared to *CS*, *jcn-lsa* achieved significant improvement in CER, F measure and REC under both threshold settings, which evidenced that linguistic information helped to improvement the error detection.

Figure 8.3 and Table 8.3(b) clearly reveals the difference in PRE and REC between two selected threshold settings. When compared to *CSSYN*, under *prob-threshold* of 0.8, REC improved significantly. The higher the *prob-threshold*, more words could be labeled as errors, which resulted in larger REC improvement of *prob-threshold* of 0.8 over that of *prob-threshold* of 0.5. Under *prob-threshold* of 0.5, the word probabilities did not affect the labeling of word with no edges. The rules to label the words with edges could correct some erroneous labeled errors, which resulted in improvement in PRE.

WSJ corpus

Figure 8.4 shows the difference in CER, PRE, REC and F measures between *CS*, *CSSYN* and *jcn-lsa* under two threshold settings. With the incremental incorporation of syntactic and semantic knowledge, CER decreased, REC and F measure increased, and PRE increased first then decreased. After incorporating syntactic knowledge, PRE increased, but it decreased after further incorporating semantic knowledge. Similar as observed for the study3 corpus, better performance was achieved by threshold setting (0.7, 0.8) than by threshold setting (0.5, 0.9), with the only exception PRE. When compared to *CS*, *jcn-lsa* under threshold setting (0.7, 0.8) improved the F measure from 16.35% to 32.02% with a relative improvement of 95.84%, and also improved CER from 11.41% to 10.49% with a relative improvement of 8.06%. The improvement in F was significant, but the improvement in CER was not significant, as shown in Table 8.4(b).

Table 8.4 reports the repeated measure analyses results, in which Table 8.4(a) reports the main effect of the knowledge condition and Table 8.4(b) reports the multiple comparison results between different knowledge

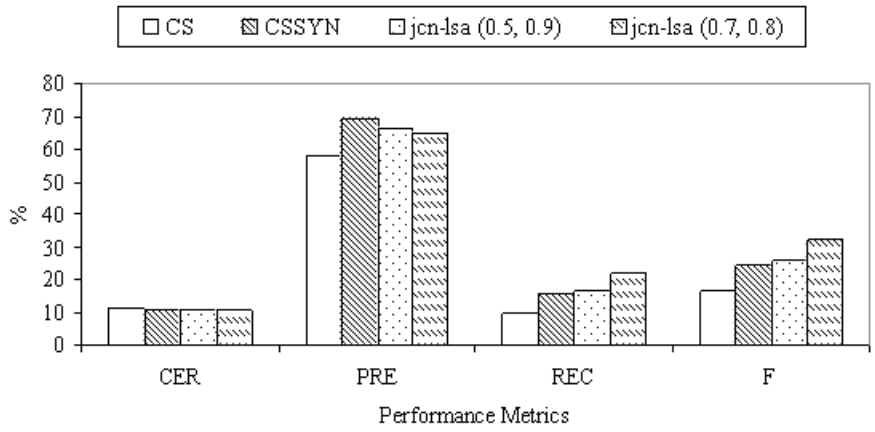


Figure 8.4: Performance of confidence measures combining non-linguistic, syntactic knowledge and lexical cohesion analysis on the WSJ corpus

conditions with Bonferroni adjustment. Under both threshold settings, the knowledge condition affected CER, F measure and REC significantly, but not PRE.

When the *prob-threshold* was set to 0.5 and the *cosine-threshold* was set to 0.9, *jcn-lsa* did not significantly improve any of the four performance metrics over those achieved by *CSSYN*. Although *CSSYN* did not significantly improve the F measure over that achieved by *CS*, *jcn-lsa* significantly improved F measure over that achieved by *CS*,

When the *prob-threshold* of 0.8 and the *cosine-threshold* of 0.6 were used, significant improvement was achieved in F measure and REC by *jcn-lsa* over that achieved by *CSSYN*. Compared to F measure and REC achieved by *CS*, significant improvement also was obtained by *jcn-lsa*.

8.2 Contributions

The research contribution of the dissertation is three-fold:

- New syntactic and semantic features were developed and incorporated into the confidence measure to detect speech recognition errors in monologue applications.
  - Syntactic features such as *number of links* and *word association based on dependency relation* from the output of lexicalized syntactic grammars were newly introduced into speech recognition error detection.

(a) Main effect of knowledge condition

Threshold		CER	F	REC	PRE
0.5, 0,9	$F(2, 18)$	8.883	8.226	6.375	2.601
	p	0.013 *	0.015 *	0.029 *	0.131
0.7, 0.8	$F(2, 18)$	6.869	21.004	17.230	2.203
	p	0.020 *	0.000 *	0.001 *	0.139

* Significant at 0.05 level

(b) Multiple comparisons between knowledge conditions (Bonferroni adjustment)

Threshold			CER	F	REC	PRE
0.5, 0,9	*CS* to *jcn-lsa*	mean difference	0.727	-9.343	-7.137	-8.360
		p	0.048 *	0.043 *	0.069	0.756
	CSSYN to *jcn-lsa*	mean difference	0.064	-1.500	-1.128	2.895
		p	1.000	0.348	0.231	1.000
0.7, 0,8	*CS* to *jcn-lsa*	mean difference	0.924	-15.674	-12.508	-6.618
		p	0.075	0.001 *	0.002 *	1.000
	CSSYN to *jcn-lsa*	mean difference	0.261	-7.832	-6.500	4.637
		p	0.586	0.000 *	0.000 *	0.998

* Significant at 0.05 level

Table 8.4: Repeat measure analyses on knowledge condition in error detection (CS, CSSYN, *jcn-lsa*) for CER, F, REC, and PRE on the WSJ corpus

  - A rule-based learning method was developed to conduct in-depth analysis of certain syntactic features.
  - The complementary effect of semantic features based on WordNet measures to other features was evaluated for the task of error detection.

- Lexical cohesion analysis was introduced into speech recognition error detection. Moreover, lexical cohesion analysis was improved by customizing it to the characteristics of speech recognition.
  - Word probability that indicates the correctness of word was revealed to be an important feature in lexical cohesion analysis for speech recognition error detection.
  - The ability to correct previously mislabeled errors based on their relations to other words proved its effectiveness to error detection on one corpus.
  - The combination of reiteration and collocation semantic relations led to improved performance over that achieved with individual relations under certain settings.

- New empirical evidence was provided by the proposed methods to error detection.
  - High-level linguistic analysis, including both syntactic and semantic analysis, can produce information generally useful to error detection. This finding achieved the goal of this dissertation.

- The distribution of grammatical categories of content words in a corpus affected error detection performance.

8.3 Future Work

The work in this dissertation paves the way for the usage of high-level linguistic analysis in error detection. We have achieved notable performance improvement by using linguistic information in error detection, and there is much room for further improvement. Several future directions related to this research are listed below.

A better model can be built for lexical cohesion analysis for error detection. In the current work, all relation edges from semantic relatedness measures were treated equally by ignoring their specific semantic relatedness values as long as they were above the preset thresholds for the individual measures. Consideration of the edges' semantic relatedness values may help to correct more false alarms for words with edges. In addition, the distance between words can also be useful to further discriminating different words. In the model presented, distance between words was not considered, and a model that discriminates the strength of edges by distance warrants exploration.

Another direction for further research is the application of other machine learning methods such as transformation-based learning that can explicitly model the relations between words to update the words with certain edges and to automatically learn threshold settings.

Proper nouns and noun phrases have a significant presence in a corpus like the Wall Street Journal. The performance of proper noun and noun phrase identification prior to semantic analysis should be investigated.

Given that the methods proposed in this dissertation were targeted to monologue applications, specifically spontaneous dictation, further experiments should be conducted to evaluate the generality of the proposed methods on other types of monologue applications such as lectures.

To evaluate the impact of the proposed error detection methods on manual error correction, user studies should be conducted. One possible user study is to compare user performance in error detection between conditions with and without the support of error predictions generated by the proposed methods. Another possible user study is to evaluate the effect of the proposed error detection methods on error navigation. Previous findings on applying confidence scores to select navigation anchors in support of hand-free error navigation [29] suggest that the accuracy of anchors is important to the efficiency of error navigation. Therefore, a better error prediction method is expected to improve the performance of anchor-based error navigation.

Automatic error detection can be a part of automatic error correction. When lexical cohesion analysis was used for malapropism detection in the text corpus [40, 39], a spelling variation of a suspected error, if related to other words in text, was assumed to be a correction. Similarly, in speech recognition output, the similarity in sound between alternative hypotheses and output words should be considered. Another factor for consideration is that the grammatical categories of recognition errors may differ from those of their reference words.

Appendix A

ACRONYMS

ASR Automatic speech recognition

ATIS Air Travel Information System

CER classification error rate

CFG context-free grammar

CS confidence score

CSR Continuous Speech Recognition

CTS conversational telephone speech

F F measure

LSA latent semantic analysis

MI mutual information

MLE Maximum likelihood estimation

OOV Out-Of-Vocabulary

PCFG Probabilistic context-free grammar

PMI Point-wise Mutual Information

POS Part-Of-Speech

PRE Precision

REC Recall

ROC Receiver-operating characteristic

RT Rich Transcription

SR Speech recognition

SVD singular value decomposition

SVM Support Vector Machine

TBL Transformation-based learning

TDT2 Topic Detection and Tracking Phase 2

WER Word Error Rate

WSJ Wall Street Journal

WSJCAM0 Wall Street Journal recorded at the University of CAMbridge (phase 0)

WWW World Wide Web

Bibliography

[1] comp.speech faq. `http://fife.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html`.

[2] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, Mexico, August 2003.

[3] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, Spain, July 1997.

[4] J. R. Bellegarda. A multispan language modeling framework for large vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6(5):456–467, September 1998.

[5] J. R. Bellegarda. Latent semantic language modeling for speech recognition. In M. Johnson, S. P. Khudanpur, M. Ostendorf, and R. Rosenfeld, editors, *Mathematical Foundations of Speech and Language Processing*, IMA Volumn in Mathematics and its Application, pages 73–103. Springer, 2004.

[6] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[7] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, December 1995.

[8] A. Budanitsky and G. Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13 – 47, March 2006.

[9] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998.

[10] P. Carpenter, C. Jin, D. Wilson, R. Zhang, D. Bohus, and A. I. Rudnicky. Is this conversation on track? In *Proceedings of the European Conference on Speech Communication and Technology*, Aalborg, Denmark, September 2001.

[11] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] E. Charniak. Immediate-head parsing for language models. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 124–131, Toulouse, France, July 2001.

[13] L. Chase. Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 815–818, Rhodes, Greece, September 1997.

[14] L. L. Chase. *Error-Responsive Feedback Mechanisms for Speech Recognizers*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 1997.

[15] C. Chelba. *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, Johns Hopkins University, Baltimore, MD, 2000.

[16] C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, and D. Wu. Structure and performance of a dependency language model. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 2775–2778, Rhodes, Greece, September 1997.

[17] C. Chelba and F. Jelinek. Exploiting syntactic structure for language modeling. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 225–231, Montreal, Quebec, Canada, August 1998.

[18] L. Chen, J.-L. Gauvain, L. Lamel, and G. Adda. Unsupervised language model adaptation for broadcast news. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–220–I–223, Hong Kong, April 2003.

[19] F. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, WA, 2000.

[20] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March 1990.

[21] S. Cox and S. Dasmahapatra. High-level approaches to confidence estimation in speech recognition. *IEEE Transactions on Speech and Audio Processing*, 10(7):460–471, October 2002.

[22] I. Dagan, S. Marcus, and S. Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 164–171, Columbus, OH, June 1993.

[23] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

[24] L. Deng and X. Huang. Challenges in adopting speech recognition. *Communications of the ACM*, 47(1):60–75, January 2004.

[25] J. Duchateau, K. Demuynck, and P. Wambacq. Confidence scoring based on backward language models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–221–I–224, Orlando, FL, May 2002.

[26] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

[27] R. M. Fano. *Transmission of information: A statistical theory of communications*. M.I.T. Press and John Wiley & Sons, Inc, 1961.

[28] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambbridge, Massachusetts, May 1998.

[29] J. Feng and A. Sears. Using confidence scores to improve hands-free speech based navigation in continuous dictation systems. *ACM Transactions on Computer-Human Interaction*, 11(4):329 – 356, December 2004.

[30] S. Furui. Toward spontaneous speech recognition and understanding. In W. Chou and B. H. Juang, editors, *Pattern Recognition in Speech and language Processing*, chapter 6, pages 191–227. CRC PRESS, 2003.

[31] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 233–237, Harriman, NY, February 1992.

[32] M. Galley and K. McKeown. Improving word sense disambiguation in lexical chaining. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1486–1488, Acapulco, Mexico, August 2003.

[33] D. Gibbon, R. Moore, and R. Winski, editors. *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, 1997.

[34] L. Gillick, Y. Ito, and J. Young. A probabilistic approach to confidence estimation and evaluation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 879–882, Munich, Germany, April 1997.

[35] D. Grinberg, J. Lafferty, and D. Sleator. A robust parsing algorithm for link grammars. Technical Report CMU-CS-95-125, Carnegie Mellon University, Pittsburgh, PA, August 1995.

[36] M. Halliday and R. Hasan. *Cohesion in English*. Longman Group Ltd, London, 1976.

[37] G. Hernández-Ábrego and J. B. M. no. Contextual confidence measures for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1803–1806, Istanbul, Turkey, June 2000.

[38] D. Hindle. noun classification from predicate-argument structures. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 268–275, Pittsburgh, PA, June 1990.

[39] G. Hirst and A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111, March 2005.

[40] G. Hirst and D. St-Onge. Lexical chains as representation of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. The MIT Press, Cambridge, MA, 1998.

[41] M. Hoey. *Patterns of lexis in Text*. Describing English language. Oxford University Press, 1991.

[42] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, New Jersey, 2001.

[43] M. hung Siu, H. Gish, and F. Richardson. Improved estimation, evaluation and applications of confidence measures for speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 831–834, Rhodes, Greece, September 1997.

[44] D. Inkpen and A. Désilets. Semantic similarity for detecting recognition errors in automatic speech transcripts. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 49–56, Vancouver, British Columbia, Canada, October 2005.

[45] R. M. Iyer and M. Ostendorf. Modeling long distance dependence in language: topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30 – 39, January 1999.

[46] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 212–219, Borovets, Bulgaria, September 2003.

[47] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research on Computational Linguistics*, pages 19–33, Taipei, Taiwan, August 1997.

[48] S. Jung, M. Jeong, and G. G. Lee. Speech recognition error correction using maximum entropy language model. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2137–2140, Jeju Island, Korea, October 2004.

[49] S. Kaki, E. Sumita, and H. Iida. A method for correcting errors in speech recognition using the statistical features of character co-occurrence. In *COLING-ACL*, pages 653–657, Montreal, Quebec, Canada, August 1998.

[50] T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 827–830, Rhodes, Greece, September 1997.

[51] O. P. Kenny, D. J. Nelson, J. S. Bodenschatz, and H. A. McMonagle. Separation of non-spontaneous and spontaneous speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 573–576, Seattle Washington, 1998.

[52] F. Kubala, J. Bellegarda, J. Cohen, D. Pallett, D. Paul, M. Phillips, R. Rajasekaran, F. Richardson, M. Riley, R. Rosenfeld, B. Roth, and M. Weintraub. The Hub and Spoke paradigm for CSR evaluation. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 37–42, Plainsboro, NJ, March 1994.

[53] J. Lafferty, D. Sleator, and D. Temperley. Grammatical trigrams: a probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural language*, Cambridge, MA, October 1992.

[54] T. Lager. The μ-TBL system: Logic programming tools for transformation-based learning. In *Proceedings of the international workshop on computational natural language learning*, pages 33–42, Bergen, Norway, June 1999.

[55] T. K. Landauer. On the computational basis of learning and cognition: Arguments from LSA. In B. H. Ross, editor, *The psychology of learning and motivation*, volume 41, pages 43–84. Academic Press, 2002.

[56] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[57] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[58] R. Lau, R. Rosenfeld, and S. Roukos. Trigger-based language models: a maximum entropy approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 45–48, Minneapolis, MN, April 1993.

[59] LDC. CSR-II (WSJ1) complete.

[60] W. A. Lea. The value of speech recognition systems. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, chapter 2, pages 39–46. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1990.

[61] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 11, pages 265–283. MIT Press, 1998.

[62] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the Annual International Conference on Systems Documentation*, pages 24 – 26, Toronto, Ontario, Canada, 1986.

[63] D. Lin. PRINCIPAR - an efficient, broad-coverage, principle-based parser. In *Proceedings of the International Conference on Computational Linguistics*, pages 482–488, Kyoto, Japan, August 1994.

[64] D. Lin. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 64–71, Madrid, Spain, July 1997.

[65] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*, pages 768–774, Montreal, Quebec, Canada, August 1998.

[66] D. Lin. Extracting collocations from text corpora. In *First Workshop on Computational Terminology*, Montreal, Quebec, Canada, August 1998.

[67] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning*, pages 296–304, Madison, WI, July 1998.

[68] D. Lin. Automatic identification of non-compositional phrases. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 317–324, College Park, MD, June 1999.

[69] K. E. Lochbaum and L. A. Streeter. Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing and Management*, 25(6):665–676, 1989.

[70] C. Ma, M. A. Randolph, and J. Drish. A support vector machines-based rejection technique for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 381–384, Salt Lake City, UT, May 2001.

[71] B. Maison and R. Gopinath. Robust confidence annotation and rejection for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 389–392, Salt Lake City, UT, May 2001.

[72] L. Mangu and M. Padmanabhan. Error corrective mechanisms for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 29–32, Salt Lake City, UT, May 2001.

[73] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Finding predominant word senses in untagged text. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 279–286, Barcelona, Spain, July 2004.

[74] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the National Conference on Artificial Intelligence*, Boston, MA, July 2006.

[75] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[76] D. Moldovan and A. Novischi. Lexical chains for question answering. In *Proceedings of the International Conference on Computational Linguistics*, pages 674–680, Taipei, Taiwan, 2002.

[77] R. Moore, D. Appelt, J. Dowding, J. M. Gawron, and D. Moran. Combining linguistic and statistical knowledge sources in natural language processing for ATIS. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, pages 261–264, Austin, TX, January 1995.

[78] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, March 1991.

[79] P. Pantel and D. Lin. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 101–108, 2000.

[80] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613 – 619, Edmonton, Alberta, Canada, July 2002.

[81] C. Pao, P. Schmid, and J. R. Glass. Confidence scoring for speech understanding systems. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[82] S. Patwardhan, , and T. Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, Trento, Italy, April 2006.

[83] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025, San Jose, CA, July 2004. Intelligent Systems Demonstration.

[84] S. D. Pietra, V. J. D. Pietra, J. Gillett, J. D. Lafferty, H. Printz, and L. Ures. Inference and estimation of a long-range trigram model. In *Proceedings of the Second International Colloquium on Grammatical Inference and Applications*, pages 78–92, 1994.

[85] S. S. Pradhan and W. H. Ward. Estimating semantic confidence for spoken dialogue systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–233–I–236, Orlando, FL, May 2002.

[86] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, Englewood Cliffs, New Jersey, 1993.

[87] M. Rayner, D. Carter, V. Digalakis, and P. Price. Combining knowledge sources to reorder N-Best speech hypothesis lists. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 217–221, Plainsboro, NJ, March 1994.

[88] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Quebec, Canada, August 1995.

[89] E. K. Ringger and J. F. Allen. Error correction via a post-processor for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 427–430, Atlanta, GA, May 1996.

[90] E. K. Ringger and J. F. Allen. A fertility channel model for post-correction of continuous speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, pages 897–900, Philadelphia, PA, October 1996.

[91] B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276, June 2001.

[92] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10(3):187–228, July 1996.

[93] R. Rosenfeld. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270 – 1278, August 2000.

[94] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October 1965.

[95] R. San-Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. M. Pardo. Confidence measures for spoken dialogue systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 393–396, Salt Lake City, UT, May 2001.

[96] R. Sarikaya, Y. Gao, and M. Picheny. Word level confidence measurement using semantic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–604–I–607, Hong Kong, April 2003.

[97] A. Sarma and D. D. Palmer. Context-based speech recognition error detection and correction. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: short papers*, pages 85–88, Boston, MA, May 2004.

[98] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, March 1998.

[99] A. Sears, J. Feng, K. Oseitutu, and C.-M. Karat. Hands-free, speech-based navigation during dictation: Difficulties, consequences, and solutions. *Human-Computer Interaction*, 18(3):229–257, 2003.

[100] A. Sears, C.-M. Karat, K. Oseitutu, A. Karimullah, and J. Feng. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society*, 1(1):4–15, June 2001.

[101] S. Sekine and R. Grishman. NYU language modeling experiments for the 1995 CSR evaluation. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, New York, NY, 1996.

[102] A. R. Setlur, R. A. Sukkar, and J. Jacob. Correcting recognition errors via discriminative utterance verification. In *Proceedings of the International Conference on Spoken Language Processing*, pages 602–605, Philadelphia, PA, October 1996.

[103] K. Seymore, S. Chen, and R. Rosenfeld. Nonlinear interpolation of topic models for language model adaptation. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[104] G. Skantze and J. Edlund. Early error detection on word level. In *Proceedings of the COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, Norwich, United Kingdom, August 2004.

[105] D. Sleator and D. Temperley. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, October 1991.

[106] M. A. Stairmand. *A Computational Analysis of lexical Cohesion with Applications in Information Retrieval*. PhD thesis, Department of Language Engineering, University of Manchester Institute of Science and Technology, 1996.

[107] N. Stokes. *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. PhD thesis, Department of Computer Science, University College Dublin, April 2004.

[108] B. Suhm. *Multimodal Interactive Error Recovery for Non-Conversational Speech User Interfaces*. PhD thesis, University of Karlsruhe, September 1998.

[109] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Conference on Information and Knowledge Management*, pages 67–74, Washington, DC, November 1993.

[110] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, pages 252–259, Edmonton, Canada, 2003.

[111] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the European Conference on Machine Learning*, pages 491–502, Freiburg, Germany, September 2001.

[112] W. Wang and M. P. Harper. The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 238–247, Philadelphia, PA, July 2002.

[113] W. Wang, A. Stolcke, and M. P. Harper. The use of a linguistically motivated language model in conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–261–4, Montreal, Quebec, Canada, May 2004.

[114] M. Weintraub, F. Beaufays, Z. Rivlin, Y. König, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 887–890, Munich, Germany, April 1997.

[115] A. Wendemuth, G. Rose, and J. Dolfing. Advances in confidence measures for large vocabulary. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 705–708, Phoenix, AZ, March 1999.

[116] F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, March 2001.

[117] F. Wessel, R. Schlüter, and H. Ney. Using posterior word probabilities for improved speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1587–1590, Istanbul, Turkey, June 2000.

[118] S. R. Young. Detecting misrecognitions and out-of-vocabulary words. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II/21–II/24, Adelaide, SA, April 1994.

[119] R. Zhang and A. I. Rudnicky. Word level confidence annotation using combinations of features. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 2105–2108, Aalborg, Denmark, September 2001.

[120] L. Zhou, Y. Shi, J. Feng, and A. Sears. Data mining for detecting errors in dictation speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(5):681–688, September 2005. Special Issue on Data Mining of Speech, Audio and Dialog.

[121] L. Zhou, Y. Shi, D. Zhang, and A. Sears. Discovering cues to error detection in speech recogntion output: A user-centered approach. *Journal of Management Information Systems*, 22(4):237–270, Spring 2006.

[122] Z. Zhou and H. Meng. A two-level schema for detecting recognition errors. In *Proceedings of the International Conference on Spoken Language Processing*, pages 449–452, Jeju Island, Korea, October 2004.

[123] Z. Zhou, H. M. Meng, and W. K. Lo. A multi-pass error detection and correction framework for Mandarin LVCSR. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1646–1649, Pittsburgh, PA, September 2006.