Word and Phone Level Acoustic Confidence Scoring for Speech Understanding Systems

by

Simo O. Kamppari

S.B., Massachusetts Institute of Technology, 1999

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

September, 1999

Signature of Author ....
Department of Electrical Engineering and Computer Science
September 1, 1999

Certified by ....
Timothy J. Hazen
Research Scientist
Department of Electrical Engineering and Computer Science

Accepted by ....
Arthur C. Smith
Chair, Department Committee on Graduate Students

Word and Phone Level Confidence Scoring for Speech Understanding Systems

by

Simo O. Kamppari

Submitted to the Department of Electrical Engineering and Computer Science in September, 1999 in partial fulfillment of the requirements for the Master of Engineering

Abstract

This thesis discusses the development and potential applications of acoustic based phone and word level confidence scores in a segment-based recognizer. The implementation of the theories in this thesis was performed in the JUPITER [10] weather information domain, using the SUMMIT [7, 26] recognizer and the TINA [11, 22] natural language understanding environment. The phone level confidence scores are derived from features based solely on the acoustic observations. The basic phone level confidence feature is a measure of how well a proposed model accounts for the acoustic observations relative to a generic *catch-all* model. The word level confidence scores are derived by combining the phone level scores in various manners. The basic word level confidence score is a simple arithmetic mean of the phone level scores within a word. The performance of the confidence scores was analyzed based on the content value of words. The results were encouraging, as the confidence scores performed significantly better on words with high content than words with low content value. This result, along with the fact that the estimation of the confidence scores was made computationally tractable by using compact approximates of the *catch-all* model, makes the confidence scores viable for use in practical applications. Based on limited experiments, using confidence scores to re-score word graphs used in the understanding component TINA yields slight increases in performance. In addition to improving the performance of existing components in a speech understanding system, robust confidence scores may enable entirely new functionality.

Keywords: speech recognition, confidence scoring, confidence score applications

Thesis Supervisor: Timothy J. Hazen
Title: Research Scientist

Acknowledgments

The last year and change I spent with the Spoken Language Group has been a wonderful experience for me. I feel like I've learned a great deal and truly enjoyed my time here. The members of the group have made this a very warm and happy place to work, and I thank them all. There are several people I would like to especially thank for their help and support above and beyond the call of duty. This thesis would not have been possible without the infinite patience, help and understanding of my advisor TJ Hazen. His ability and willingness to describe technical matters in such a fashion that even I could understand was invaluable. I would also like to thank my officemate Michelle Spina for her insight into the practical issues surrounding the systems at SLS, and for putting up with my venting about my car and the Boston police. Without her help this thesis would surely not be finished for yet another year, and chances are that I would have gone insane. I am also very grateful for the help I received on natural language material from Joe Polifroni and Stephanie Seneff. I would also like to thank Ed Hurley, Lee Hetherington and Christine Pao for helping me the many times I managed to break my computer. On the less academic side, but certainly not less important, I must thank Issam Bazzi and Karen Livescu for the countless hours of entertaining discussion over hundreds of cups of coffee. I will truly miss those coffee breaks.

This research was supported by DARPA under contract DAAN02-98-K-003, monitored through U.S. Army Natick Research, Development and Engineering Center.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Definition

The goal of an automatic speech understanding system is to correctly recognize words uttered by humans and then extract meaning from them. However, even the current state of the art speech recognition systems make errors on a regular basis. The errors are in large part due to errors in the recognition process. These recognition errors are in the form of substitutions, deletions and insertions. Furthermore, these errors may lead to undesirable misunderstandings. Eliminating these errors completely maybe impossible, thus a more reasonable goal is to derive robust methods for figuring which words might be wrong. Humans are quick to analyze utterances and ask people to repeat parts they may have not heard well. If reliable word level confidence metrics can be generated automatically, they can enable a natural manner for asking users to repeat or clarify parts of utterances. They can also be used to aid in partially parsing utterances when a complete parse is not possible. Word level confidence scores could play a key role in making the user feedback from automatic speech understanding systems more natural and useful.

To this end, this thesis attempts to define a set of robust confidence metrics which will enable improvements in parsing and possibly user feedback. The metrics are analyzed under various conditions and methods for deriving an optimum indicator of confidence are defined. Furthermore, the confidence indicator is incorporated into the parsing process and is analyzed for the process of user feedback.

1.2 Background

This thesis focuses on world level confidence scoring within GALAXY, an architecture for spoken language systems designed and developed by the Spoken Language Systems (SLS) group at the MIT Laboratory for Computer Science [12]. The GALAXY architecture connects together the various components necessary to perform language understanding tasks. The component in GALAXY responsible for generating the recognition hypotheses is called SUMMIT [7, 26]. The work in this thesis revolves around SUMMIT and, to a smaller degree, the natural language component TINA [11, 22]. Errors in SUMMIT are caused by misclassification of words, the use of out-of-vocabulary words in a user request, the existence of poor channel conditions and/or various other spontaneous speech effects (such as false starts and partial words).

As errors occur on different levels in the speech understanding process, the meaning of the confidence scores varies accordingly. There are generally 4 different levels on which the scores are considered: understanding, sentence, word and phone.

The highest level errors occur at the understanding level [17]. Understanding refers to the system's ability to extract meaning from a spoken phrase. For example, the system must know that a person uttering: "*Hello, Please tell me about the weather in the city of Boston tomorrow*," is conveying a 1) *a request* 2) *for weather* 3) *in Boston* 4) *tomorrow.* Generally, misrecognizing any of these four key components leads to a mistake in understanding. For a domain specific system, it may be sufficient to recognize only a subset of these four. For example, JUPITER, a weather domain information system, needs only parts *3)* and *4)* for a proper understanding. A confidence score at this level describes the certainty that a system's interpretation of a phrase is correct [17].

The next level down from speech understanding is sentence level recognition. At this level, a confidence score reflects the word and phone level performance across the entire utterance. Scores at this level are not widely used, but they may indicate general channel characteristics and be useful in conjunction with the understanding level scoring [17].

Word level recognition is next [6, 23, 24]. At this level, the correctness of individual words in a recognition hypothesis is considered. Correct recognition entails finding and classifying each word regardless of context and meaning. Returning to the previous example, the correct word recognition corresponds to recognizing the utterance as the string of words: "HELLO PLEASE TELL ME ABOUT THE WEATHER IN THE CITY OF BOSTON TOMORROW." As hinted above, it is possible to make an error at this level without causing an error in understanding; specifically, if a non-critical word is incorrectly recognized, it may still be possible to extract the full meaning from the phrase [17].

Phone level confidence scores are defined at the lowest level. At this level the confidence in the classification of the lowest level measurements in a speech understanding system is considered. In the version of SUMMIT used in this thesis these measurements correspond to landmarks, where as in a Hidden Markov Model (HMM) based system the lowest level measurements correspond to frames. While the phone level scoring in this thesis is implemented on the landmarks of a segmentation based system, the techniques used in conjunction with the landmark level measurements carry over to the corresponding frames in a HMM based system.

This thesis focuses on the lowest two levels mentioned above, word and phone level, with the primary emphasis on word level scores. The SLS group is currently working on several domains which may benefit from access to confidence scores. These systems include JUPITER, PEGASUS, VOYAGER which cover weather, flight and traffic information respectively [8, 9, 12]. Currently, the user feedback of these systems is relatively limited. For example, if a system is unsuccessful in understanding just a portion of a request it may reject the entire utterance. Also at the times of rejection, the user is given no clue to why a request may have failed. Confidence scores may allow better behavior in both situation.

A robust confidence measure maybe useful in picking the right path through a proposed N-best word graph. The current natural language component TINA utilizes an *ad hoc* method, described further in Section 6.1.1, for weighing nodes of a N-best word graph which represent some measure of confidence in the words. The *ad hoc* method for weighting the words can be replaced with the actual word level acoustic confidence scores. Replacing the ad hoc scoring with the actual scores provides slight performance increases in parsing. Because the language model scores are also incorporated at this point, they are not used as features for word level confidence.

In addition to boosting parsing performance, the confidence metrics can be used for identifying weak points in a hypothesized output. Pinpointing weak points in a hypothesis may allow more informative feedback. For example, if most words in an utterance are recognized with a high confidence, but a critical word is recognized with a low one, the system may prompt the user to repeat that critical word. Prompting users for more information can increase the likelihood that the user's request is ultimately correctly understood.

1.2.1 Previous Work

As the value of robust confidence metrics has been obvious for quite some time, much work has already been done in the field [6, 23, 24]. Although each approach has been somewhat different in their specific implementations, many of the methods have much in common. Some work has been done at language understanding [17] and phonetic level [1], while most of the existing work has focussed on the problem at the word level.

The approaches are based on finding key features indicative of a correct recognition or understanding and then deriving a confidence metric from them. The primary differences in the approaches can be accounted for in the specific set of confidence features proposed, as well as the methods with which the features are combined.

The number of useful features available for confidence scoring is potentially very large. Anything correlated with correctness of the system output can be used as a feature. Past research efforts have used features extracted from a variety of different intermediate results of recognition [1, 14] or understanding. Some of the more commonly used features include number of competing hypotheses for a word, the number of hypotheses in the N-best list and other values indicative of uncertainty in output. Also, features based on language model information have been found to work very well. However, this thesis does not use language model based features because the language model information is incorporated in the parsing process where the confidence scores are used to complement the language model. Generally, the set of proposed features is based on both intuition and empirical relationships demonstrated by the data.

The correlation between any single feature and correctness is simple to calculate. These correlations allow for a relative evaluation of individual features. However, more interesting than the correlation between a single feature and correctness is the correlation between a complete set of features and correctness. To incorporate information from all the proposed features various methods have been proposed. One such method is via the use of neural networks [21]. The inputs to the neural net are the various proposed features and the output is a decision whether or not the input values correspond to a correctly recognized word. The output decision is then compared against a truth value and the nodes of the net are adjusted so as to maximize agreement between the decision output the truth value. Ideally this approach allows features to complement each other to achieve better performance than a single feature could achieve.

1.2.2 The SUMMIT Speech Recognition System

The various spoken language applications developed by SLS use a recognition engine called SUMMIT [7, 26]. SUMMIT is a segment-based speech recognition system developed by the Spoken Language System group at MIT. The segment-based approach differs from the more common *frame-based* speech recognition systems in that a segmentation network is used for recognition instead of a sequence of frames.

The recognition process begins in the same fashion as in *frame-based* recognizers. First a sequence of observation frames are measures. The observation frames contain spectral information in the form of Mel Frequency Cepstral Coefficient (MFCC) averages [19]. Potential segment boundaries, referred to as landmarks, are next proposed based on changes in spectral content in between frames. Using a set of heuristic rules, various segmentations are proposed based on the proposed landmarks. Figure 1.1 illustrates four possible segmentations $S1$,$S2$,$S3$ and $S4$ for four landmarks $L1$,$L2$,$L3$ and $L4$, where each solid bar corresponds to a segment. As a segmentation hypothesis is made, some of the landmarks become segment internal while others remain transitional boundaries. For example, in the Figure 1.1, if the segmentation $S2$ is hypothesized then landmarks $L1$,$L2$ and $L4$ become transitional boundaries and $L3$ becomes an internal boundary. In SUMMIT segments roughly correspond to individual phonetic units.

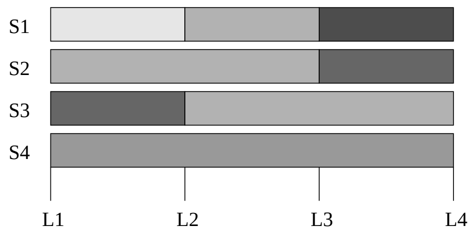

Figure 1.1: Possible segmentations for a given set of landmarks in the SUMMIT recognition system.

Based on the segmentation, word pronunciation models and language model information, various words are hypothesized. Each utterance is defined to have at least two words. Each utterance begins with the word <*pause1*> and ends with the word <*pause2*>. The start boundary <*pause1*> matches up with the first segment boundary in the utterance and the end boundary of <*pause2*> matches up with the last segment boundary of the utterance. Words <*pause1*> and <*pause2*> may also occur in the middle of an utterance where they account for silences. In between <*pause1*> and <*pause2*> lies the hypothesized string of words.

Figure 1.2 illustrates the relationship between the landmarks, segments, phones and words. The work in this thesis revolves primarily around the landmarks and the words.

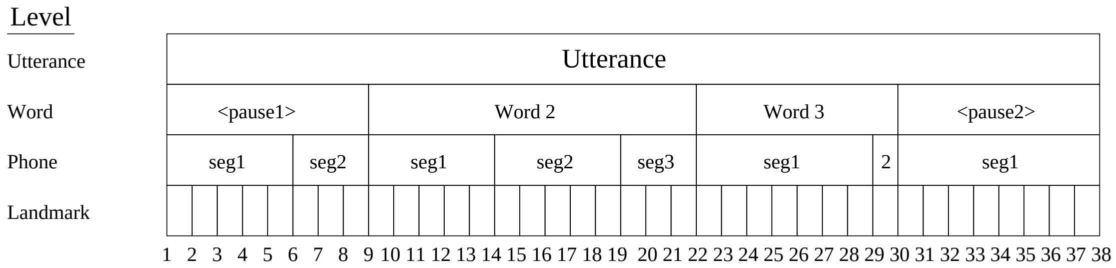

Figure 1.2: A possible segmentation of an utterance in the SUMMIT recognition system.

1.3 Thesis Overview

The goal of this thesis is to develop confidence scoring metrics within SUMMIT recognition environment, and analyze methods for utilizing the scores within the GALAXY architecture. The concepts of this work are general to all of the domains, but the experiments and the research itself was conducted in the JUPITER weather information domain. JUPITER was a natural choice for the experiment domain, as it is the oldest and most robust of the several domains currently under development by SLS. A complete description of the corpora used in this thesis and details regarding the JUPITER recognizer, can be found in Appendix A. In addition to the readily available data, the behavior of the JUPITER system is well documented and stable.

Deriving word level scores from acoustic features is a two step process. First, phone level confidence scores are calculated, and then they are combined to derive a word level scores. The details of this process are discussed in Chapter 2.

The correlations with correctness vary greatly between the various word level features and working with a large set of features is cumbersome. Thus, various methods for combining these features into a single metric were proposed. Of these methods, discussed in Chapter 3, Fisher linear discriminant analysis provided the best results.

The performance of the word level confidence scoring methods varies as a function of the words for which they are being used. Chapter 4 describes the effects of word classes on the performance of the confidence scores.

Computation time is not a significant issue in laboratory experiments, however in building a usable real-time system it becomes a crucial one. The initial confidence metrics discussed in Chapter 2 were computationally expensive and inefficient, thus methods for improving the efficiency and speed were also explored. Chapter 5 discusses implementation details significant in making the confidence scoring system work in real time.

Applications for confidence scores are numerous. This thesis includes a brief analysis of the value of confidence scores for improving parsing and understanding performance. The confidence scores are used to replace a current ad hoc method for weighing nodes in a word graph used for understanding. Incorporating the confidence scores into this process results in mild gains in performance. The description of the parsing application, as well as a brief description of issues surrounding user feedback, can be found in Chapter 6.

Because of the broad range of issues addressed there is much room for future work and improvements at various levels. The confidence scores can be improved by various means, and their applications are too numerous to list. A reliable set of confidence scores is sure to have many uses, future direction and lessons learned in this thesis are discussed in Chapter 7, Conclusions and Future Work.

Chapter 2

Word and Phone Confidence Scoring

The value of confidence scores lies in their ability to indicate the certainty that a particular phone or word is correctly classified. An ideal confidence metric would precisely point to every error, however achieving such performance would imply perfect speech recognition. Thus, rather than try to solve the problem of speech recognition, the goal is to develop robust confidence scoring methods which express confidence in a probabilistic manner.

The process of developing confidence scores is akin to finding features correlated with correct recognition hypothesis and then combining these features to form a metric that conveys confidence. As mentioned, this thesis limits the features for confidence scoring to purely acoustic features, which means features like the language model scores are not directly used. However, the language model information is valuable, and it is generally known that features derived from the language model information work well. Thus, leaving the language model information completely unused is bad idea. Instead of using the language model scores to generate confidence scores, the confidence scores are used to complement the language model information at a later stage. If the language model scores were used in creating confidence scores, it would not be possible to use the confidence scores to complement the language model without suffering from circular reasoning.

2.1 Phone Level Confidence Scoring

There are many ways in which a word level confidence scoring problem can be approached. This thesis approaches word level scoring from the perspective of the underlying phone level scores. The phone level scores are used to calculate word level features that are used to derive a confidence measure. In this thesis, references to phone level scoring actually refer to the scoring of individual landmarks rather than phones themselves. Thus, *phone level* scoring refers to the *scale* at, rather than the units on which, the scoring takes places.

As described in Section 1.2.2, landmarks represent hypothesized phone boundaries which become either internal or transitional to the proposed segments [7]. An observation feature is associated with each landmark which contains MFCC averages, and other acoustic information, from frames around the landmarks.

Because each phone is generally defined by only a few landmarks, the landmark scoring is referred to as phone level. If one is interested in true phone level scores, the phone level scores can be derived from the landmark scores via methods analogous to deriving the word level scores from landmark scores as described in Section 2.2.

This thesis utilizes only boundary models for confidence scoring, although it is possible to utilize the corresponding segment models in addition to, or in the place of, the boundary models. Only the boundary models were used because the current JUPITER recognizer uses only boundary scores, making them readily accessible.

While this thesis chooses to derive the word level scores from phone level scores there is no reason it has to be this way. It is possible to perform word level confidence scoring without ever performing a phone level analysis. To do so, is to limit the word level confidence features to strictly word level metrics, such as: length of the word, language model score, etc [3].

2.1.1 Motivation

Deriving the word level scores from the phone level ones is a natural extension of the recognition process, as the word level recognition hypotheses are largely a function of the underlying phone level hypotheses. Similarly, this thesis bases its word level confidence scores largely on the phone level confidence scores, which makes the phone level scoring necessary.

In addition to being necessary for word level scoring there are potential benefits from the phone level scores themselves. In general the phone level confidence scores lend further insight into the confidence features. Better understanding of the features will then help in building more robust confidence metrics in the future. It is also possible that average phone scores across an utterance may yield information about the channel. It is not clear how the average should be interpreted and it may not be very valuable in itself. However, the average may prove valuable when used in conjunction with other observations. This thesis computes phone level scores only as an intermediate step to calculating word level confidence metrics, and explores no phone level applications for them.

2.1.2 Theory

This thesis uses a ratio of acoustic scores as a fundamental feature of confidence from which many of the other features are derived. The most basic metric for phone level confidence is the maximum *a posteriori* (MAP) probability $\mathrm{P}(c_i|\vec{x})$ [16]. This metric, which is referred to as $C_{map}(c_i|\vec{x})$, is shown in Equation 2.1, where c_i corresponds to a proposed boundary class and $\vec{x}$ to the acoustic observation.

$$C_{map}(c_i|\vec{x}) = \mathrm{P}(c_i|\vec{x}) = \frac{\mathrm{p}(\vec{x}|c_i)\mathrm{P}(c_i)}{\mathrm{p}(\vec{x})} = \frac{\mathrm{p}(\vec{x}|c_i)\mathrm{P}(c_i)}{\sum_{j=1}^{N_c} \mathrm{p}(\vec{x}|c_j)\mathrm{P}(c_j)} \tag{2.1}$$

Equation 2.1 shows the Bayes expansion of the $\mathrm{P}(c_i|\vec{x})$ which is a ratio of two scores. The $C_{map}(c_i|\vec{x})$ score can be thought of as the ratio of a *proposed score* and *catch-all score*, $\mathrm{p}(\vec{x})$. The *proposed score*, $\mathrm{p}(\mathrm{\vec{x}}|\mathrm{c_i})\mathrm{P}(\mathrm{c_i})$, reflects how well a proposed boundary model c_i accounts for the acoustic observation $\vec{x}$, taking into account $\mathrm{P}(c_i)$, the prior probability for c_i.

The *catch-all score* reflects how well the JUPITER models in general account for the acoustic observation. Explicitly, it is the likelihood of the observation $\vec{x}$ occurring, also known as $\mathrm{p}(\vec{x})$. A mathematical formulation of the *catch-all score* is shown in Equation 2.2, where N_c is the number of specific models, $\mathrm{p}(\vec{x}|c_j)$ is the likelihood that acoustic observation $\vec{x}$ occurred given the class c_j, and $\mathrm{P}(c_j)$ is the prior probability of class c_j.

$$\mathrm{p}(\vec{x}) = \sum_{j=1}^{N_c} \mathrm{p}(\vec{x}|c_j)\mathrm{P}(c_j) \tag{2.2}$$

The range of values for $C_{map}(c_i|\vec{x})$ is between 0 and 1. A value close to 0 indicates that there are other models which score as well as, or better, than c_i, indicating low confidence and potentially high confusability. A value close to 1, on the other hand, signifies that the proposed model c_i scores considerably better than any other model

$c_j; i \neq j$. This indicates a region of low confusability and thus high confidence.

A slight variation of C_{map}, the normalized log likelihood score C_{nll}, is equal to the log of $C_{map}(c_i|\vec{x})$ normalized by the prior probability $\mathrm{P}(c_i)$. This score, as shown in Equation 2.3, performs slightly differently from $C_{map}(c_i|\vec{x})$ since it is based purely on the acoustic observation, as it ignores the prior probability.

$$C_{nll}(c_i|\vec{x}) = \log\left(\frac{\mathrm{p}(\vec{x}|C_i)}{\mathrm{p}(\vec{x})}\right) = \log\left(\frac{\mathrm{p}(\vec{x}|c_i)}{\sum_{j=1}^{N_c}\mathrm{p}(\vec{x}|c_j)\mathrm{P}(c_j)}\right) \tag{2.3}$$

Since the prior probability $\mathrm{P}(c_i)$ in the numerator is removed, the ratio before the log operation is no longer constrained to be between 0 and 1. Thus the range of values for the C_{nll} range from $-\infty$ to $\log \mathrm{P}(\mathrm{c_i})$. The more positive the C_{nll} score is, the higher the confidence.

In addition to the C_{map} and C_{nll} scores, the $\mathrm{p}(\vec{x})$ score as seen in Equation 2.2 can also be used as a confidence feature. This feature is indicative of how well the acoustic models of JUPITER are able to account for the acoustic observation $\vec{x}$. This feature is not very useful in determining phone level confidence, however, it can be indicative of how much the C_{map} and C_{nll} scores can be trusted. It can be thought of as *confidence* on the confidence scores.

A low $\mathrm{p}(\vec{x})$ score indicates poor coverage by the JUPITER models, which can be a sign of non-speech sounds, noise, or other abnormal conditions. A low $\mathrm{p}(\vec{x})$ lowers the significance of C_{map} and C_{nll}, because these scores are only meaningful if the acoustic data falls in the range of JUPITER's acoustic models. For example, a non-speech sound not previously observed in the training data may score poorly for both the proposed model and the *catch-all* model. This may lead to a C_{map} score which is artificially high, which would incorrectly indicate a high confidence in the proposed model. Since the actual observed phenomena is not accounted for by JUPITER's acoustic models it can not possibly be correct. Similarly, a high $\mathrm{p}(\vec{x})$ score indicates good acoustic coverage by the JUPITER acoustic models, and thus the features C_{map} and C_{nll}, have more meaning.

2.2 Word Level Confidence Scoring

This thesis approaches word level confidence scoring as a function of the underlying phone scores described above in Section 2.1. Because of this, the performance of the word level scores is greatly dependent on how good the phone level scores are. This thesis makes the assumption that the phone level scores are good enough that word level scores can be successfully based on them.

2.2.1 Theory

As mentioned, the word level scores are derived from the phone level scores C_{nll}, p($\vec{x}$), and C_{map} described in Section 5.2.1. There are several ways to analyze the phone level scores and derive a word level score from them. The basic idea in this thesis is that the collection of the phone level scores which fall within a hypothesized word are reflective of how well the hypothesized word can account for the acoustic evidence. The better the acoustic evidence is supported by a hypothesized word, the higher the confidence that the word was in fact correct. The corresponding word score can be expressed mathematically as Equation 2.4, where C_{word} is the confidence score for the word. $F_c(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n | word)$ is a function of the acoustic measurements $\vec{x}_i$, the number of phones within the hypothesized word n, and the hypothesized phone classes c_i.

$$C_{word} = F_c(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n | c_1, c_2, \ldots, c_n) \tag{2.4}$$

By varying the function F_c, different methods for combining the phone level scores can be explored. Several F_c functions are explored in this thesis, primarily focusing on arithmetic and geometric means. The mathematical description of the arithmetic mean C_{am} is shown in Equation 2.5, where N_L is the number of landmarks within the word, $C_p(c_i|\vec{x}_k)$ is the phone level confidence score for the kth landmark given the acoustic evidence $\vec{x}_k$ and a proposed model c_k.

$$C_{am}(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k, c_1, c_2, \ldots, c_k) = \frac{1}{N_L} \sum_{k=1}^{N_L} C_p(c_k|\vec{x}_k) \tag{2.5}$$

Similarly, Equation 2.6 describes the mathematical description of the geometric mean C_{gm}.

$$C_{gm}(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k, c_1, c_2, \ldots, c_k) = e^{\left(\frac{1}{N_L} \sum_{k=1}^{N_L} \log[C_p(c_k|\vec{x}_k)]\right)} \tag{2.6}$$

As some scores are initially in *non-logarithmic* space (C_{map}), and others in *logarithmic* space (C_{nll}), it is important that the meaning of the arithmetic and geometric means is well defined. This thesis refers to all averages in the *non-logarithmic* space as *arithmetic means*, and all averages in the *logarithmic* space as *geometric means*.

The two means have distinct behaviors depending on the underlying scores. The geometric mean is well suited for emphasizing small individual values. The geometric mean allows a small value to pull down the average for an entire word more so than an arithmetic mean. For example, a word which contains one or two low phone level scores will have a resulting geometric mean which is very low. On the other hand, an arithmetic mean would be less sensitive to one or two low value scores. In an arithmetic mean a single low score has only a small impact, especially if the number of phone level scores to be averaged is large. A benefit of the arithmetic mean is that it is less sensitive to small outliers, and thus will be more indicative of the average ability of the JUPITER models to account for the acoustic observations. Because of their unique behaviors both the arithmetic and geometric means of C_{map} and C_{nll} are used as features.

In addition to the above means several other features were proposed. Standard deviations σ_{map} and σ_{nll}, for the C_{map} and C_{nll} scores, are used as indicators to how the means should be interpreted. These standard deviations, shown in Equation 2.7 and Equation 2.8, are only useful in conjunction with the mean scores. For example, a high arithmetic mean along with a low standard deviation would have a higher confidence than a high arithmetic mean with a high standard deviation. In the former case, the low standard deviation translates to scores being close to each other, the mean being high translates to the scores being high, which means that all the scores are high. In the latter case, the high standard deviation means that the scores are widely dispersed, thus it is difficult to say if the mean is high because of a few outliers or if the scores were indeed consistently high. Generally, a low standard deviation makes the word level mean scores more reliable.

$$\sigma_{map} = \sigma(C_{map}) = \mathrm{E}\left[C_{map}^2\right] - \mathrm{E}\left[C_{map}\right]^2 \tag{2.7}$$

$$\sigma_{nll} = \sigma(C_{nll}) = \mathrm{E}\left[C_{nll}^2\right] - \mathrm{E}\left[C_{nll}\right]^2 \tag{2.8}$$

Other word level features proposed are various minimum scores. Namely, $C_{min-map}$, $C_{min-nll}$ and a slight variation of the $C_{min-map}$ the $C_{min-map-int}$. The mathematical representations of these scores are in Equation 2.9, Equation 2.10 and Equation 2.11, where i is the landmark and Ω is the set of all landmarks which are internal to segments.

$$C_{min-map} = \min_i \left[C_{map}(i)\right] \tag{2.9}$$

$$C_{min-nll} = \min_{i} \left[C_{nll}(i) \right] \tag{2.10}$$

The $C_{min-map-int}$ differs from $C_{min-map}$ only in that the $C_{min-map-int}$ only considers landmarks which are internal to segments. This feature was added because it gives insight into how well the segments themselves are scoring. Since only boundary scores are used, the segment internal boundary scores are the closest thing to segments scores.

$$C_{min-map-int} = \min_{i \in \Omega} \left[C_{map}(i) \right] \tag{2.11}$$

The minimum scores are similar to the standard deviation scores in that they may not have much meaning in themselves but in conjunction with the mean scores they can prove valuable. For example, having a high arithmetic mean value and a high "minimum" value is a sign that the acoustic evidence is well matched to the hypothesized word. Similarly a low geometric mean score along with a relatively high "minimum" score indicates that the low geometric mean is not due to a singular low phone level score but rather a set of systematically low phone level scores across the entire utterance.

The average *catch-all* model score p^{A} is also used as a feature. The p^{A} corresponds to the arithmetic mean of the $\mathrm{p}(\vec{x})$, which describes the average ability of the JUPITER models to account for the acoustic observations in a word.

The last two features which are used are N_{nbest} and N_{phones}. While these two are only indirectly a function of the acoustic evidence, they can be correlated with correctness. N_{nbest} is the number of competing hypothesis on the *n-best* list. The fewer hypotheses there are, the better the JUPITER models are doing at modeling the acoustic evidence. This is similar to the *catch-all* model score mentioned in Section 2.1.2. N_{phones} is actually the number of landmarks within each word. There is a correlation between the length of a word and the likelihood that the word is correctly hypothesized. Generally, longer words are more acoustically distinct than shorter ones, thus the chance of confusion is much smaller for longer words.

Table 2.1 is a complete list of features used for the word level confidence evaluation task.

Feature	Description
C^A_{map}	Arithmetic mean of the C_{map} scores
C^A_{nll}	Arithmetic mean of the C_{nll} scores
C^G_{map}	Geometric mean of the C_{map} scores
C^G_{nll}	Geometric mean of the C_{nll} scores
p^A	*Catch-all* model score
$C_{min-map}$	Minimum C_{map} score in the word
$C_{min-map-internal}$	Minimum internal C_{map} score in the word
$C_{min-nll}$	Minimum C_{nll} score in the word
N_{nbest}	Number of utterances on *n-best* list
σ_{map}	Standard deviation of C_{map}
σ_{nll}	Standard deviation of C_{nll}
N_{phones}	Number of landmarks in word

Table 2.1: A complete list of word level features used for confidence scoring.

2.2.2 Implementation

The actual implementation of the means utilizes a slightly modified form of the Equation 2.5. As shown in Figure 2.1, the first and last boundaries of each word are shared with adjacent words. Because these boundaries are shared, they are used in calculating the word level scores for a given word as well as the words preceding and following that word.

The first and last boundary of each word are weighted by a half to account for the shared bounds. This gives equal weight to each landmark score across the entire utterance. Since the two boundaries are weighted by a half, the denominator term of the mean must also be decremented by one. Equations Equation 2.12 and Equation 2.13 describe the actual implementation, where N_L is the number of landmarks within the, word including the boundary landmarks, and $C_{map}(c_{i,k}|\vec{x}_k)$ and $C_{nll}(c_{i,k}|\vec{x}_k)$ are the scores for landmark k.

$$C^A_{map} = \frac{1}{N_L - 1}\left[0.5\left(C_{map}(c_{i,1}|\vec{x}_1) + C_{map}(c_{i,N_L}|\vec{x}_{N_L})\right) + \sum_{k=2}^{N_L-1} C_{map}(c_{i,k}|\vec{x}_k)\right] \quad (2.12)$$

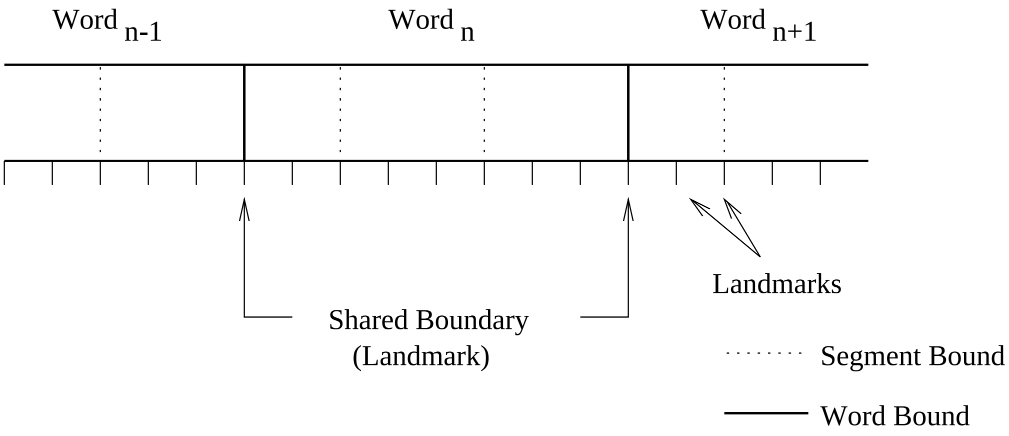

Figure 2.1: Boundary landmarks which fall between two segments are shared by the adjacent segments.

$$C^G_{nll} = \frac{1}{N_L - 1}\left[0.5\left(C_{nll}(c_{i,1}|\vec{x}_1) + C_{nll}(c_{i,N_L}|\vec{x}_{N_L})\right) + \sum_{k=2}^{N_L-1} C_{nll}(c_{i,k}|\vec{x}_k)\right] \tag{2.13}$$

The features C^G_{map} and C^A_{nll} were calculated next. Calculating the C^G_{map} follows the form of the Equation 2.6. The equation must be slightly altered to account for the weighting of the first and last boundary as described above, the new formula can be seen in Equation 2.15.

$$B = 0.5\left(\log[C_{map}(c_{i,1}|\vec{x}_1)] + \log[C_{map}(c_{i,N_L}|\vec{x}_{N_L})]\right) \tag{2.14}$$

$$C^G_{map} = e^{\left\{\frac{1}{N_L - 1}\left[B + \sum_{k=2}^{N_L-1} \log[C_{map}(c_{i,k}|\vec{x}_k)]\right]\right\}} \tag{2.15}$$

The calculation of C^A_{nll} also follows the form of Equation 2.6, however, the exponent and log are switched as shown in Equation 2.16.

$$C^A_{nll} = log\left\{\frac{1}{N_L - 1}\left[0.5\left(e^{C_{nll}(c_{i,1}|\vec{x}_1)} + e^{C_{nll}(c_{i,N_L}|\vec{x}_{N_L})}\right) + \sum_{k=2}^{N_L-1} e^{[C_{nll}(c_{i,k}|\vec{x}_k)]}\right]\right\} \tag{2.16}$$

The p^A was calculated via the arithmetic mean described in Equation 2.5, with the adjustments for the first and last boundaries as described above. The resulting form of the p^A is shown in Equation 2.17.

$$\mathrm{p}^A = \frac{1}{N_L - 1}\left[0.5\left(\mathrm{p}(\vec{x}_1) + \mathrm{p}(\vec{x}_{N_L})\right) + \sum_{k=2}^{N_L-1} \mathrm{p}(\vec{x}_k)\right] \tag{2.17}$$

A slight variation of the above equations is used to account for the very first and last words in the utterance. Because the first and last words of the utterance do not share their first and last bounds respectively with any other words they do not need to be weighted by a half.

Calculation of the minimum scores $C_{min-map}$ and $C_{min-nll}$, is a straight forward minimum calculation. The smallest landmark value within the word is picked out in each case. The calculation of the $C_{min-map-\mathrm{int}}$ varies in that only landmarks which correspond to internal phone boundaries are considered. If a word contains no internal bounds then the $C_{min-map}$ score is used for the $C_{min-map-\mathrm{int}}$ as shown in Equation 2.18.

$$C_{min-map-\mathrm{internal}} = \begin{cases} \min_{i\in\Omega}\left[C_{map}(i)\right] & \text{if } \Omega \neq \text{NULL} \\ C_{min-map} & \text{otherwise} \end{cases} \tag{2.18}$$

The standard deviation scores σ_{map} and σ_{nll} are calculated following the form of Equation 2.7 and Equation 2.8. The values $\mathrm{E}\left[C^2_{map}\right]$ and $\mathrm{E}\left[C^2_{nll}\right]$ are calculated using an arithmetic mean as described in Equation 2.5. Unlike in the calculation of C_{map} and C_{nll} means, the first and last landmark are not weighted any differently from the other landmarks. Weighting them by a half, as in the case of the C_{map} and C_{nll} mean scores, would skew the true standard deviation of the scores. Thus, the equations for $\mathrm{E}\left[C^2_{map}\right]$ and $\mathrm{E}\left[C^2_{nll}\right]$ are shown in Equation 2.19 and Equation 2.20 respectively, where N_L is the number of landmarks in the word.

$$\mathrm{E}\left[C^2_{map}\right] = \frac{1}{N_L}\sum_{k=1}^{N_L} C^2_{map} \tag{2.19}$$

$$\mathrm{E}\left[C^2_{nll}\right] = \frac{1}{N_L}\sum_{k=1}^{N_L} C^2_{nll} \tag{2.20}$$

The $\mathrm{E}\left[C_{map}\right]$ and $\mathrm{E}\left[C^2_{nll}\right]$ are calculated in a similar fashion. Since the C^A_{map} and C^A_{nll} are calculated with the weighted first and last boundary, their values can not be used as $\mathrm{E}\left[C_{map}\right]$ and $\mathrm{E}\left[C^2_{nll}\right]$ respectively. Instead, the values are calculated via regular arithmetic means as shown in Equations 2.21 and Equation 2.22.

$$\mathrm{E}\left[C_{map}\right] = \frac{1}{N_L}\sum_{k=1}^{N_L} C_{map} \tag{2.21}$$

$$\mathrm{E}\left[C_{nll}\right] = \frac{1}{N_L}\sum_{k=1}^{N_L} C_{nll} \tag{2.22}$$

At the end of each word the values are combined to calculate values for σ_{map} and σ_{nll}. The calculations follow the exact form of Equation 2.7 and Equation 2.8 and are shown here in Equation 2.23 and Equation 2.24.

$$\sigma_{map} = \mathrm{E}\left[C_{map}^2\right] - \mathrm{E}\left[C_{map}\right]^2 \tag{2.23}$$

$$\sigma_{nll} = \mathrm{E}\left[C_{nll}^2\right] - \mathrm{E}\left[C_{nll}\right]^2 \tag{2.24}$$

The last two word level features, N_{nbest} and N_{phones}, are easily derived from the recognition output. Minimum value for N_{nbest} is one, otherwise no hypothesis would be made, and for N_{phone} it is 2, as each word must have at least a start and end boundary.

2.2.3 Experiments

Test Conditions

The word level confidence scores were evaluated on the utterances in a test set described in Appendix A. For each utterance the best recognition hypothesis in the *N-best* list was used for evaluation. Word level confidence scores were calculated for each word in the hypothesis, and the correctness of each word in the hypothesis was evaluated against a forced transcription of the utterance. The correctness of each word is binary, if the hypothesized word matches the forced transcription in the word and location then it is considered correct. The performance of the confidence measures is evaluated based on the correctness information and confidence measures associated with each word.

Evaluation Metrics

The performance of the individual word level features varies a great deal. Some of the features work well by themselves, while others yield little or no value alone. Because of their nature, it is difficult to have a single figure of merit which adequately describes the performance of each of the features. The goal of the word level confidence measure is to classify words as correctly or incorrectly recognized. This becomes a classic detection problem, thus a Receiver Operating Characteristic (ROC) graph, plotting the relationship between detection and false alarms, seems like an appropriate manner for comparing the performance of the various features. Words with confidence scores exceeding a set threshold are classified as correct while those below it are classified as incorrectly recognized. The ROC curve plots the detection/false alarm relationship as the decision threshold is varied. In this context, detection refers to the probability that a word which is in fact correctly classified has a confidence score which indicates that the word is correct:

$$\text{P(detection)} = \text{P(word classified as correct}|\text{word is correct)}$$

Similarly, a false alarm refers to the probability that a word is classified as correct when in fact the word is incorrectly recognized:

$$\text{P(false alarm)} = \text{P(word classified as correct}|\text{word is incorrect)}$$

A more detailed description of ROC curves is given in Appendix B.

While a single figure of merit is inadequate for fully describing the performance of a feature, it can be helpful for observing relative performance of features at particular points of interest. Since detection and false-alarm rate are functions of each other, picking one determines the other. Depending on the goal of the confidence scoring, one of the two maybe more relevant. In the context of this thesis, a high detection rate is desirable, thus values for detection are pegged and a figure or merit is calculated based on the pegged values. In this case, the pegged value describes the minimum bound for the detection. Therefore the figure of merit describes the performance of a feature for all the detection values greater than and equal to the pegged value. The figure of merit, illustrated in Figure 2.2, is the area under the ROC curve and above the pegged threshold for detection, normalized by the area above the pegged threshold. The greater the area under the curve the better the feature works for detection rates greater than and equal to the threshold.

Results

Table 2.2 displays the FOM performance of the various features for various thresholds. The most significant column in the table is column with threshold 0.8, since it best describes the detection range in which we are interested in applying the confidence scores. The features are ordered based on performance on this column. Included in this table is the FOM feature *chance* which corresponds to randomly classifying words as correct and incorrect. The *chance* FOM represents a theoretical lower bound on performance, any FOM lower than *chance* can be made greater than *chance* by simply reversing the decision rule for that feature.

Feature	**Threshold**				
	0.5	0.6	0.7	**0.8**	0.9
C^G_{nll}	0.6276	0.5715	0.5032	**0.4114**	0.2826
C^A_{nll}	0.5689	0.5184	0.4562	**0.3782**	0.2687
$C_{min-map}$	0.5981	0.5381	0.4622	**0.3617**	0.2164
C^G_{map}	0.5953	0.5305	0.4542	**0.3546**	0.2155
$C_{min-nll}$	0.5270	0.4647	0.3910	**0.3018**	0.1751
C^A_{map}	0.5201	0.4466	0.3590	**0.2591**	0.1441
σ_{map}	0.4527	0.3878	0.3093	**0.2136**	0.1132
$C_{min-map-\text{internal}}$	0.4027	0.3417	0.2732	**0.1965**	0.0977
p^A	0.3334	0.2852	0.2321	**0.1752**	0.1105
N_{phones}	0.2107	0.1633	0.1160	**0.0776**	0.0421
σ_{nll}	0.1735	0.1301	0.0904	**0.0555**	0.0253
N_{nbest}	–	–	–	–	0.0681
chance	0.25	0.20	0.15	0.1	0.05

Table 2.2: The phone level performance of all individual features based on the figure of merit.

The performance differences between the geometric and arithmetic mean methods for combining phone level features are clearly seen in the Table 2.2. The geometric means C^G_{nll} and C^G_{map} outperform their arithmetic mean counter parts C^A_{nll} and C^A_{map} respectively. This illustrates the geometric means ability to punish the entire word score for a poor individual phone level score. If a word is correctly recognized, then the individual phone level boundary models should match the acoustic observations reasonably well. Therefore, for a correctly recognized word the phone level scores are also reasonably good. Conversely, if a word is incorrectly recognized, it is likely that at least one phone level score is low. The geometric mean is able to pull the word level score down based on this single low score. Where as an arithmetic mean may be only slightly impacted by a single low score, especially if the number of landmarks in the word is large.

Analysis based solely on the figure of merit described above indicates that the best feature in the set is C^G_{nll}, with C^A_{nll} and $C_{min-map}$ somewhat behind. The C^G_{nll} score is clearly the best one and the C^A_{nll} appears to be very close to $C_{min-map}$, however performance of the latter two features is largely a function of the threshold of interest. For some thresholds the two features are very similar in performance and for others there is a significant difference. This threshold dependent variation illustrates the problem of attempting to evaluate feature performance based on a single figure of merit.

To better understand the relationship between the performance of various features an ROC graph can be utilized. Figure 2.3 displays the ROC curves for the three best performing features. As the ROC shows the performance of the features depends on which operating point is chosen. From the ROC curve it is clear that for a low threshold, such that high detection probability is achieved, the C_{nll}^{G} is in fact the best feature. The figure does a great job in illustrating the threshold dependent performance of the C_{nll}^{A} and $C_{min-map}$ features. If the performance criterion for the features calls for a high *detection* rate, say over 90%, there is little difference between the performance of the two features. However, if a low *false alarm* rate is important then $C_{min-map}$ clearly out performs C_{nll}^{A}.

While the above features performed quite well on their own some of the other features did not. To get a better feel for just how poorly some of the features did on their own Figure 2.4 shows the ROC curves for N_{phones}, σ_{nll}, and $p(\vec{x})$. A straight line between the origin and the top right hand corner or the graph would represent a completely random assignment or correctness. The curves for these features are quite close to that. Some of them appear to be negatively correlated at some points, this can be attributed to statistical noise.

The best performing single features, such as C_{nll}^{G} and $C_{min-map}$, perform reasonably well. They can achieve approximately 85% correct acceptance rate with about 50% false alarm rate. While this is reasonable, improvements can be made by combining multiple features together as described in Chapter 3. The performance of the features is also dependent on what type of words are being analyzed, the impact of the word types is discussed in Chapter 4.

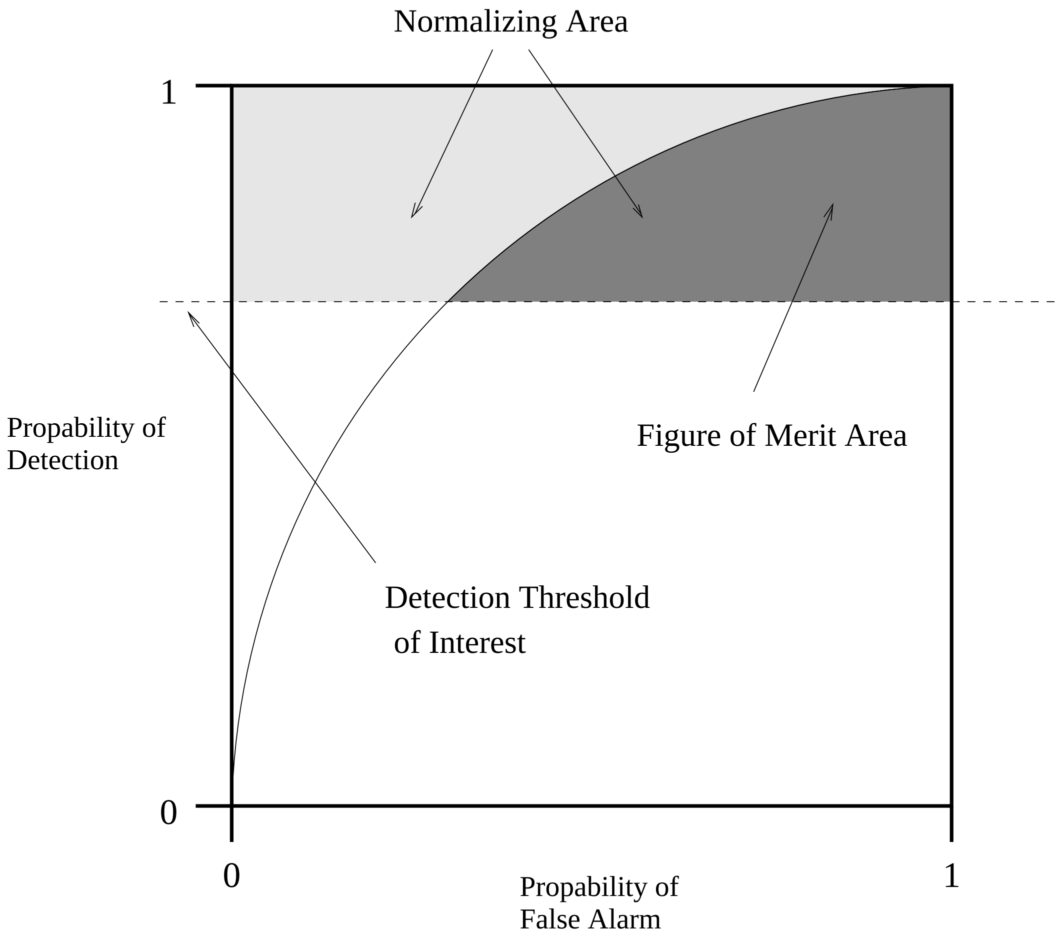

Figure 2.2: Illustration of the figure of merit used in this thesis as an indicator of confidence score performance.

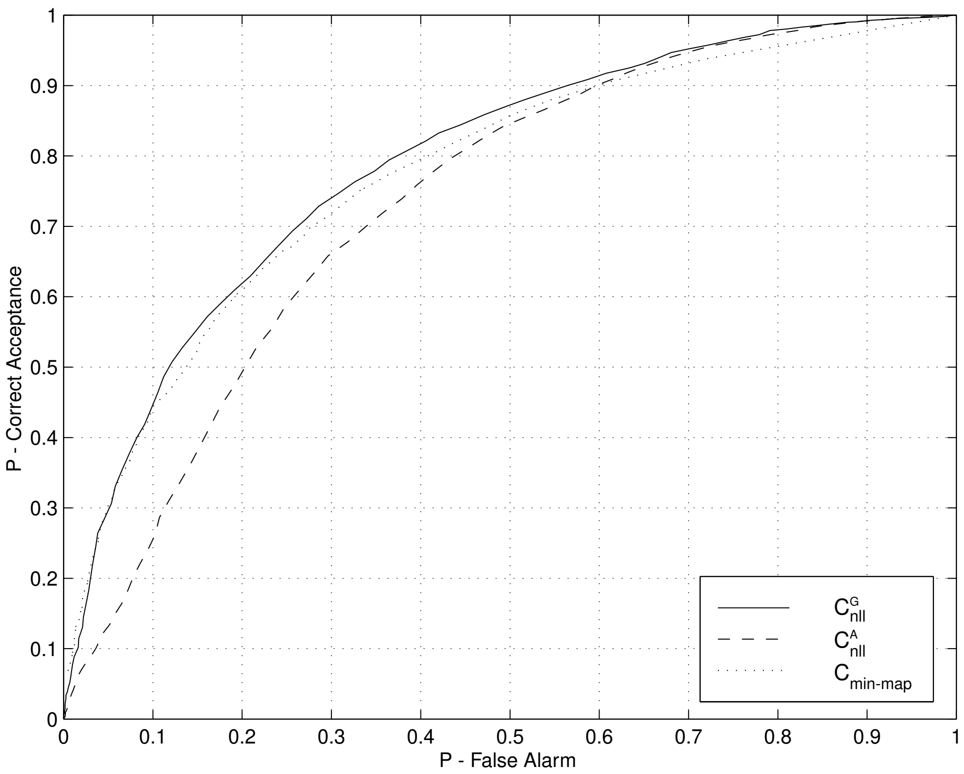

Figure 2.3: The ROC curves for the three best performing individual word level confidence features.

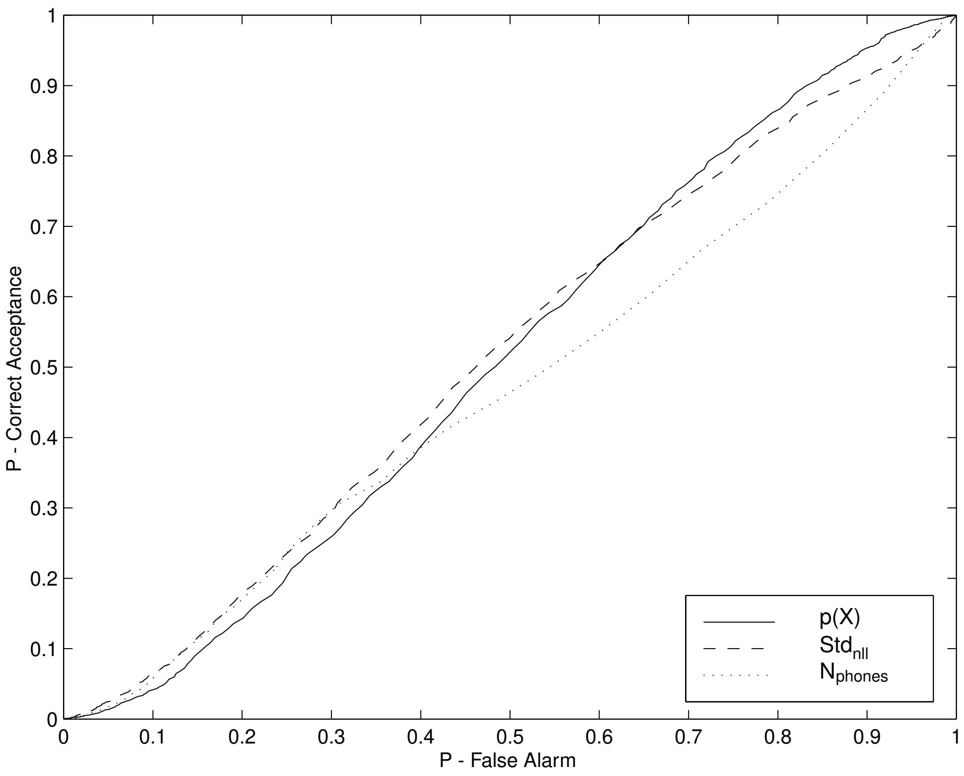

Figure 2.4: ROC curves for poorly performing individual word level confidence features.

Chapter 3

Combining Features

Chapter 2 describes various features which can be used as indicators of confidence at both word and phone level. The performance of these features varies greatly. There are several ways in which these features can be utilized. For example, any one single feature could be used as a confidence metric, but this wouldn't use all the information available. Alternatively, all the features could be used simultaneously which would incorporate all the information. However, analyzing the 12 different confidence metrics, each correlated in a different manner, can be very difficult. Yet another approach, which is the one adopted by this thesis, is to derive a new single measure from complete list of 12 features. The idea is that the 12 features can be combined in some fashion where they complement each other to create a new feature which is more robust. Ideally, the various features should be combined in such a fashion that the weaknesses of each feature are covered up by the strengths of another. This thesis explores two different ways for combining the features, probabilistic modeling for hypothesis testing and Fischer Linear Discriminant analysis. The following sections describe each of these approaches and Section 3.3 outlines the results from these approaches.

3.1 Probabilistic Modeling

3.1.1 Theory

A probabilistic hypothesis testing approach was experimented with for the purpose of combining features into a single metric. Words in the training half of the development set are labeled, based on a comparison between the hypothesis and forced transcription, as either correct or incorrect. A detailed description of the development set can be found in Appendix A. The $\vec{F}$ vectors associated with the correct words are used to train a *correct* mixture Gaussian model $M_{correct}$. Similarly, the incorrect words' $\vec{F}$

vectors are used to train an *incorrect* model $M_{incorrect}$. Once the models are trained, the problem becomes a simple hypothesis testing problem shown in Equation 3.1. New word level scores $\vec{F}$ are scored against the *correct* and *incorrect* models and the word is classified as correct or incorrect based on the threshold K. Word is classified as correct if:

$$\frac{\mathrm{p}(\vec{F}|M_{correct})}{\mathrm{p}(\vec{F}|M_{incorrect})} > K \tag{3.1}$$

By varying the K, a ROC performance evaluation can be performed for this technique. This thesis performed initial experiments to evaluate the performance of the hypothesis testing approach for two type of Gaussian mixture models, the diagonal and full covariance mixture Gaussian models.

3.1.2 Diagonal Gaussian Mixture Model

Diagonal mixture Gaussian models with 50 mixture Gaussian components were trained for both the correct $M_{correct}$ model and incorrect $M_{incorrect}$ model. Since diagonal mixture Gaussians assume zero covariance between dimensions, the number of trainable parameters is significantly reduced. Only the diagonal terms of the covariance matrix must be calculated for each mixture. The number of individual word level confidence features in a feature vector $\vec{F}$ is 12, thus the number of covariance parameters to be trained is $12 * 50 = 600$. The relatively small number of covariance parameters makes it possible to train robust diagonal models with a relatively small amount of training data. The downside of using strictly diagonal models, and a limited number of mixtures, is that it maybe difficult to model some complex relationships in between dimensions. Generally correlations between dimensions can be accounted for by additional mixtures, but with 50 mixtures it may be difficult to capture all the inter-dimensional covariances. Principal components analysis (PCA) can be beneficial in conjunction with diagonal mixture models, however it was not performed as the expected gains in performance would not be sufficiently large to be significant.

3.1.3 Full Covariance Gaussian Mixture Model

Full covariance models differ from the diagonal ones in that covariances between individual mixtures are now allowed. This allows the models to account for more complex behavior with the same number of mixtures. The downside to allowing non-zero covariances is that the number of parameters, which need to be trained, increases significantly. With a 12 dimensional $\vec{F}$ and 50 mixtures, the number of covariance parameters increases to $\frac{12^2+12}{2} * 50 = 3900$. Considering the limited size of training sets (approximately 8000 correct tokens, and 1000 incorrect tokens), it is likely that the full covariance $M_{correct}$ and $M_{incorrect}$ models are not sufficiently trained. The $M_{incorrect}$ model is likely to be especially poor since only about 10% of the training set corresponds to misrecognized words.

3.2 Fisher Linear Discriminant Analysis

The goal of the Fisher Linear Discriminant Analysis (FLDA) is to reduce the dimensionality of a space to one dimension while achieving maximum separation between classes [5]. Figure 3.1 illustrates an example in two dimensions on how FLDA technique can be used to reduce dimensionality without losing the ability to differentiate between classes.

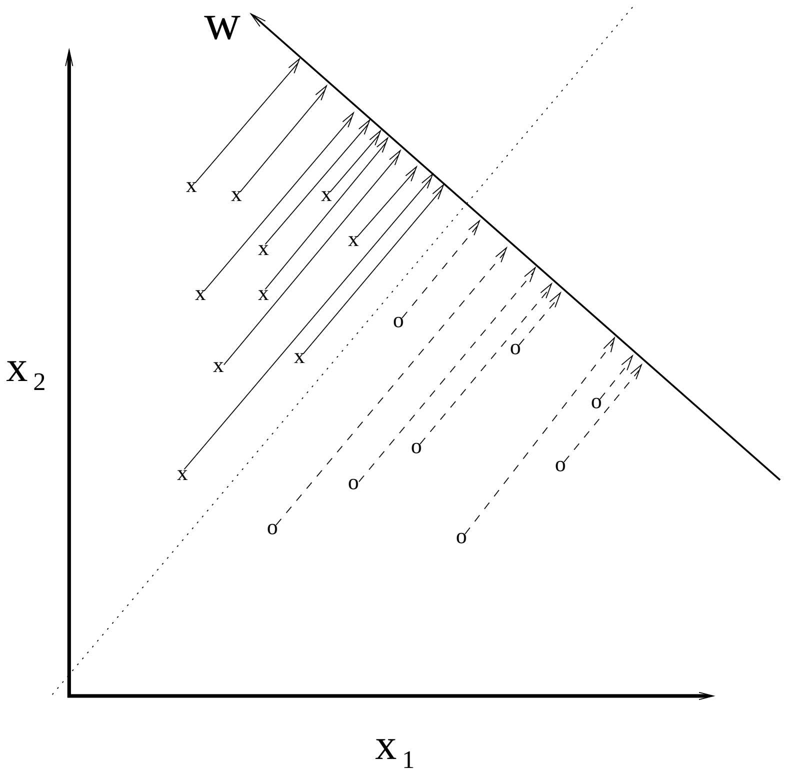

Figure 3.1: An illustration of FLDA dimensionality reduction via projection to a single dimension.

To achieve the above criterion, FLDA must calculate a projection vector $\vec{w}$ which projects the data from d dimensions to 1. Suppose that the original set or data consists of n d-dimensional samples $\vec{x}_1, \ldots, \vec{x}_n$ (in our case d equals 12). Further more, suppose that that the samples $\vec{x}_1, \ldots, \vec{x}_n$ consist of samples from two different classes χ_1 and χ_2(which in our case are the correctly recognized and incorrectly recognized words). Thus, by a linear combination of the components of $\vec{x}$ the new set of n 1-dimensional samples, y, can be obtained.

$$y = \vec{\mathrm{w}}^{\mathrm{t}}\vec{\mathrm{x}} \tag{3.2}$$

The set $\vec{y}$ consisting of $\vec{y}_1, \ldots, \vec{y}_n$, is also divided into two classes Υ_1 and Υ_2. The direction of $\vec{w}$ is important while the magnitude of $\vec{w}$ is of no real significance since it only scales the y. For an appropriately chosen $\vec{w}$ and χ_1 and χ_2 which are initially separated in the d-dimensional space, the resulting Υ_1 and Υ_2 can also be well separated as shown in Figure 3.1.

The difference of the sample means is used as the measure of separation. The sample mean for the projected points $\tilde{m}_i$ is given by Equation 3.4.

$$\begin{aligned} \tilde{m}_i &= \frac{1}{n_i} \sum_{\vec{y} \in \Upsilon_i} y & (3.3) \\ &= \frac{1}{n_i} \sum_{\vec{\mathrm{x}} \in \chi_{\mathrm{i}}} \vec{\mathrm{w}}^{\mathrm{t}}\vec{\mathrm{x}} = \vec{\mathrm{w}}^{\mathrm{t}}\vec{\mathrm{m}}_{\mathrm{i}} & (3.4) \end{aligned}$$

Thus $|\tilde{m}_1 - \tilde{m}_2| = |\vec{\mathrm{w}}^{\mathrm{t}}(\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2)|$. Since the difference in means can be made arbitrarily large by scaling $\vec{w}$, the difference must be large relative to a measure of standard deviation for each of the classes. Instead of a simple variance, a *scatter* is defined for the projected samples as $\tilde{s}_i^2$.

$$\tilde{s}_i^2 = \sum_{y \in \Upsilon_i} (y - \tilde{m}_i)^2 \tag{3.5}$$

An estimate of the variance of the pooled data is therefore $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$, where $(\tilde{s}_1^2 + \tilde{s}_2^2)$ is the *within-class scatter* of the projected samples. The *Fisher linear discriminant* is defined at the $\vec{\mathrm{w}}^{\mathrm{t}}\vec{\mathrm{x}}$ for which $\mathbf{J}(\vec{\mathrm{w}})$, as in Equation 3.6, is maximized.

$$\mathbf{J}(\vec{\mathrm{w}}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{3.6}$$

Scatter matrices $\mathrm{S_i}$ and $\mathrm{S_W}$ are defined as $\mathrm{S_i} = \sum_{\mathbf{x} \in \chi_{\mathrm{i}}} (\vec{\mathrm{x}} - \vec{\mathrm{m}}_{\mathrm{i}})(\vec{\mathrm{x}} - \vec{\mathrm{m}}_{\mathrm{i}})^{\mathrm{t}}$ and $\mathrm{S_W} = \mathrm{S_1} + \mathrm{S_2}$, in order to define J as an explicit function of $\vec{w}$. With these definitions $\tilde{s}_i^2$ can be defined as follows.

$$
\begin{aligned}
\tilde{s}_i^2 &= \sum_{\vec{\mathrm{x}} \in \chi_\mathrm{i}} (\vec{\mathrm{w}}^\mathrm{t}\vec{\mathrm{x}} - \vec{\mathrm{w}}^\mathrm{t}\vec{\mathrm{m}}_\mathrm{i})^2 \\
&= \sum_{\vec{\mathrm{x}} \in \chi_\mathrm{i}} \vec{\mathrm{w}}^\mathrm{t}((\vec{\mathrm{x}} - \vec{\mathrm{m}}_\mathrm{i})(\vec{\mathrm{x}} - \vec{\mathrm{m}}_\mathrm{i})^\mathrm{t}\vec{\mathrm{w}} \\
&= \vec{\mathrm{w}}^\mathrm{t}\mathrm{S}_\mathrm{i}\vec{\mathrm{w}}
\end{aligned}
\tag{3.7}
$$

Therefore $\tilde{s}_1^2 + \tilde{s}_2^2 = \vec{\mathrm{w}}^\mathrm{t}\mathrm{S}_\mathrm{W}\vec{\mathrm{w}}$ and similarly,

$$
\begin{aligned}
(\tilde{m}_1 - \tilde{m}_2)^2 &= (\vec{\mathrm{w}}^\mathrm{t}\vec{\mathrm{m}}_1 - \vec{\mathrm{w}}^\mathrm{t}\vec{\mathrm{m}}_2)^2 \\
&= \vec{\mathrm{w}}^\mathrm{t}(\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2)(\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2)^\mathrm{t}\vec{\mathrm{w}} \\
&= \vec{\mathrm{w}}^\mathrm{t}\mathrm{S}_\mathrm{B}\vec{\mathrm{w}}
\end{aligned}
\tag{3.8}
$$

where $\mathrm{S}_\mathrm{B} = (\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2)(\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2)^\mathrm{t}$. As S_W is defined, it is the *within-class scatter matrix* which is proportional to the sample covariance of the pooled d-dimensional data, and S_B is the *between-class scatter matrix.* Equation 3.6 can be written in terms of S_B and S_W as shown in Equation 3.9 below.

$$
J(\vec{\mathrm{w}}) = \frac{\vec{\mathrm{w}}^\mathrm{t}\mathrm{S}_\mathrm{B}\vec{\mathrm{w}}}{\vec{\mathrm{w}}^\mathrm{t}\mathrm{S}_\mathrm{W}\vec{\mathrm{w}}} \tag{3.9}
$$

It can be shown that the $\vec{w}$ which maximizes J must satisfy $\mathrm{S}_\mathrm{B}\vec{\mathrm{w}} = \lambda\mathrm{S}_\mathrm{W}\vec{\mathrm{w}}$ which is a generalized eigenvalue problem. If S_W is nonsingular a conventional eigenvalue problem can be obtained by setting $\mathrm{S}_\mathrm{W}^{-1}\mathrm{S}_\mathrm{B}\vec{\mathrm{w}} = \lambda\vec{\mathrm{w}}$, from which the solution, as shown in Equation 3.10, can be derived.

$$
\vec{\mathrm{w}} = \mathrm{S}_\mathrm{W}^{-1}(\vec{\mathrm{m}}_1 - \vec{\mathrm{m}}_2) \tag{3.10}
$$

In this thesis, the projection vector $\vec{w}$ was calculated based on the development data set as defined in Appendix A. Once the $\vec{w}$ was calculated the feature vector $\vec{F}$ was multiplied to get the final confidence metric F_{score} as shown in Equation 3.11.

$$
F_{score} = \vec{w}^t\vec{\mathrm{x}} \tag{3.11}
$$

3.3 Results

Based on the initial experiments, the FLDA approach proved to be the more effective method of the two. Both of the mixture model approaches, Diagonal and Full Covariance, performed about equally well. The evaluation of the above feature combination methods was performed on the development set described in Appendix A.

The Development set was divided into two subsets, a training subset and a testing subset. The Diagonal and Full Covariance models, along with the fisher projection vector, were trained on the training subset and then tested on the testing subset.

Figure 3.2 displays the relative performance of the above methods on the testing subset of the development set. The fisher discriminant analysis provided significant improvements in performance, while the diagonal and full covariance model provided little or no gains over a single one of the best performing features. While the fisher discriminant analysis performed significantly better than the mixture model approaches, it is not clear which of the mixture models performed better. Which approach performs better depends on the desired operating point. For a operating point which required very low false alarm rate, a rate below 15%, the Full Covariance approach performs slightly better. Conversely, for a operating point which requires a high correct acceptance, say over 50%, the Diagonal models perform better. It is expected that with increased training data the performance of the Full Covariance models would exceed the performance of the Diagonal models, whether or not the performance would ever reach the level of the FLDA is unclear.

Because the fisher discriminant analysis performed so much better than the other methods, it was adopted as the method for combining the individual features. Figure 3.3 illustrates the difference in performance on the final testing set between C^G_{nll}, the best performing single feature, and F_{score}.

The following Table 3.1 shows the figure of merit values for the F_{score} and C^G_{nll} shown in Figure 3.3.

Feature	**Threshold**				
	0.5	0.6	0.7	**0.8**	0.9
F_{score}	0.6746	0.6209	0.5500	**0.4502**	0.2889
C^G_{nll}	0.6276	0.5715	0.5032	**0.4114**	0.2826
chance	0.25	0.20	0.15	0.1	0.05

Table 3.1: The figure of merit performance of the best individual word level feature, C^G_{nll}, and the FLDA derived word level score, F_{score}.

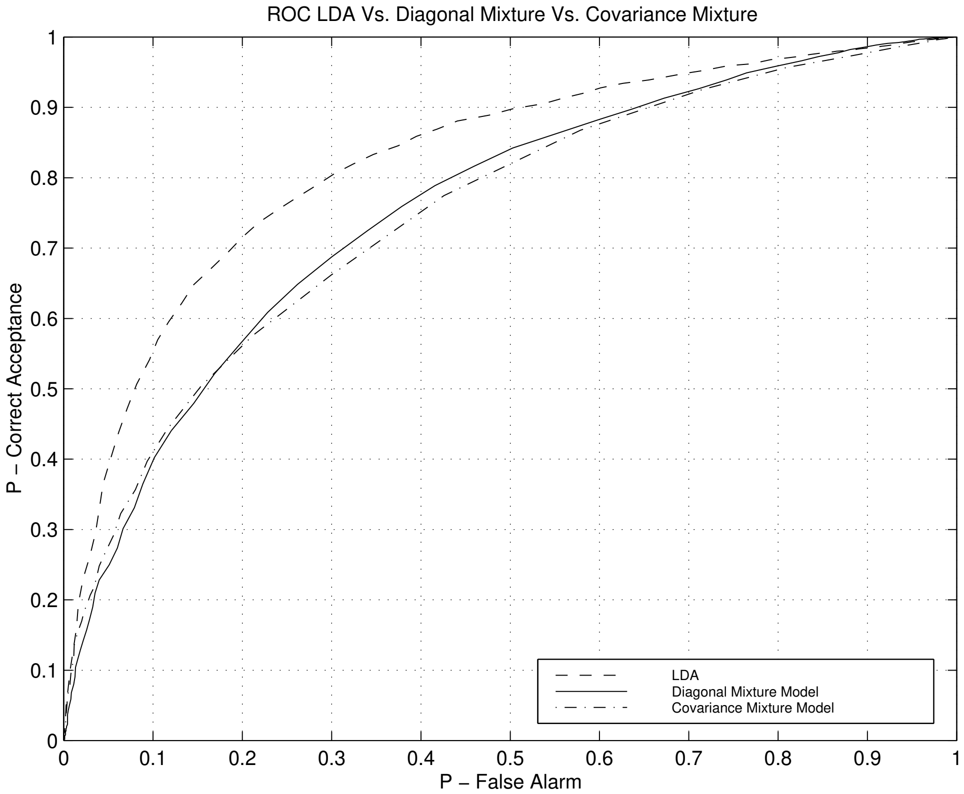

Figure 3.2: The ROC curves indicating relative word level confidence performance for various feature combination methods.

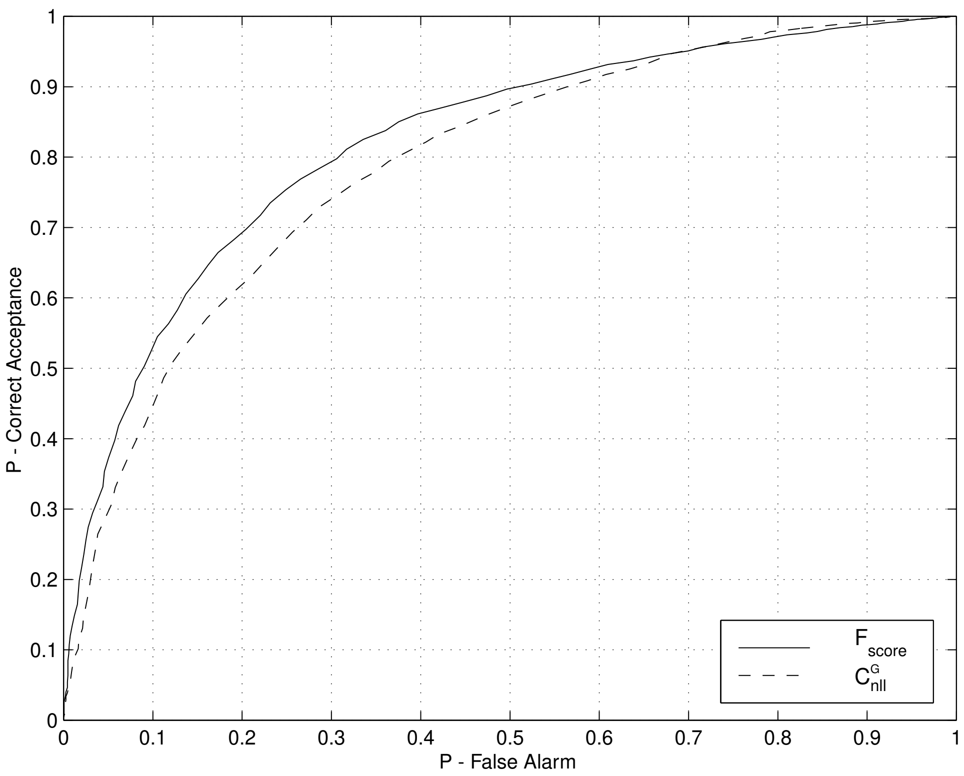

Figure 3.3: Relative word level ROC performance of the FLDA combined word level confidence score, F_{score}, and the best single word level confidence feature, C_{nll}^{G}.

Chapter 4

Analysis Using Word Classes

The confidence metrics described in the previous chapters are applicable to all words hypothesized by the recognizer. However, in terms of content, some words are more significant than others. Significance of a word can be thought of as the amount of critical information it carries. While all words in an utterance carry information, only some of the words carry information necessary to convey the overall meaning of an utterance. Because language is highly domain specific, much of the meaning in an utterance can be implicit. For example, in a limited domain system such as JUPITER, a request for weather information in Boston could be: "*Could you please tell me the weather in Boston today?*", or alternatively it could be: "*Boston today.*" One is a complete sentence, the other is not, yet they both convey the same information. In the latter case, the fact that the utterance is a question for weather is implicit from the context. In this case two words were able to convey the same information as ten, from which it follows that some words clearly have more content information than others. Words which are crucial to the meaning of an utterance can be considered very *significant*, or high in content value. Similarly which can be done without are less *significant*, or low in content value.

Because some words are more significant than others, we are more interested in the correctness of some words than others. It doesn't matter much if a word like "please" is misrecognized as "fleece" since in the context of weather information there is no content in those words. However, if a word like "Boston" is misrecognized as "Austin" this could cause significant problems. The confidence scores can be helpful in discovering recognition errors for all types of words, but clearly there is greater value in discovering errors on words with high content value. The purpose of this chapter is to evaluate how well the confidence metrics work for words with various degrees of information content. Each word is categorized into a content class based on the amount of critical information the word contains, and next the performance of the confidence scores is evaluated for words in each content class.

4.1 Description of Classes

The content classes are defined by the amount of critical information that a word carries. Four content classes are defined in this thesis. The content classes are labeled 1 through 4, with class 1 representing words with the lowest content and class 4 the highest. The precise categorization of the words is shown in Appendix C.

4.1.1 Content Class 4

The highest content class, class 4, consists of words which are crucial to a correct understanding of a query. For the JUPITER domain, *location* is the most important piece of information, thus this content class consists of all the locations in JUPITER's vocabulary. Locations in JUPITER are primarily city and country names, like:' '*Boston*" and "*Japan*", and to a smaller extent continents and other geographic locations. Words in this class are often sufficient in themselves for correctly responding to a user's query.

4.1.2 Content Class 3

Content type 3, consists of words which are almost as important as the location, but may not be quite sufficient in themselves to understand a query. These words primarily describe time, and to a smaller extent specific weather conditions of interest.

4.1.3 Content Class 2

The second lowest class is labeled content type 2, it contains words which contain little crucial information to a weather domain system. These words by themselves contain little content information. They generally complement complex queries rather than express content themselves. Some of the words in this class include: "*what*", "*thanks*", and "*forecast.*" These words in themselves do not add meaning, and they are often implicit. However, they can be helpful in detecting information about the users communication status with JUPITER, and are important for grammatical correctness. Grammatical correctness has little value in JUPITER, however some of these words can still be useful on their own. For example, a user saying "*thanks*" or "*thank you*" can indicate that the user has received the information they were looking for and is now ready to move to a new query or end the call.

Word Class	Number of Tokens
Class 1	7781
Class 2	500
Class 3	942
Class 4	1856

Table 4.1: Number of tokens in each word class defined in this thesis.

4.1.4 Content Class 1

The lowest class, content type 1, contains words like "*a*", "*the*", "*you*", etc. These words are typically function words which contain no content but are needed in order to create syntactically correct sentences. This category also includes filled pause words, such as: "*uh*" and "*um.*" All the words which do not fall in the word content classes 2,3 and 4, as described above, are classified as content type 1.

From the perspective of understanding, the word classes classes 3 and 4 are most important, while the words in classes 1 and 2 have little or no value in terms of deciphering a query. Thus, it is desirable to have good confidence score performance on classes 3 and 4, while the performance on classes 1 and 2 matters less.

4.2 Effects on Performance

As mentioned, performance on words significant in terms of understanding is most important. As it turns out, the performance of the words in classes 3 and 4, which represent high content words, is in fact better than the performance of the words in classes 1 and 2. Figure 4.1 illustrates the relative performance of the Fisher combined confidence metric F_{score} on the four word classes.

The content classes 3 and 4 perform clearly better than the content classes 1 and 2. However, it is difficult to ascertain which of the two classes, 3 or 4, performs better because the ROC curves are jagged. The ROC curves for the classes 2 and 3 are especially jagged because the number of tokens in those classes is much fewer than in the classes 1 and 4. It is likely that additional data would smooth out the curves for classes 2 and 3. The smoothed curves 2 and 3 would probably fall near 1 and 4 respectively. Table 4.1 shows the number of tokens in each class.

Because of the limited data in classes 2 and 3, two new classes were defined. The

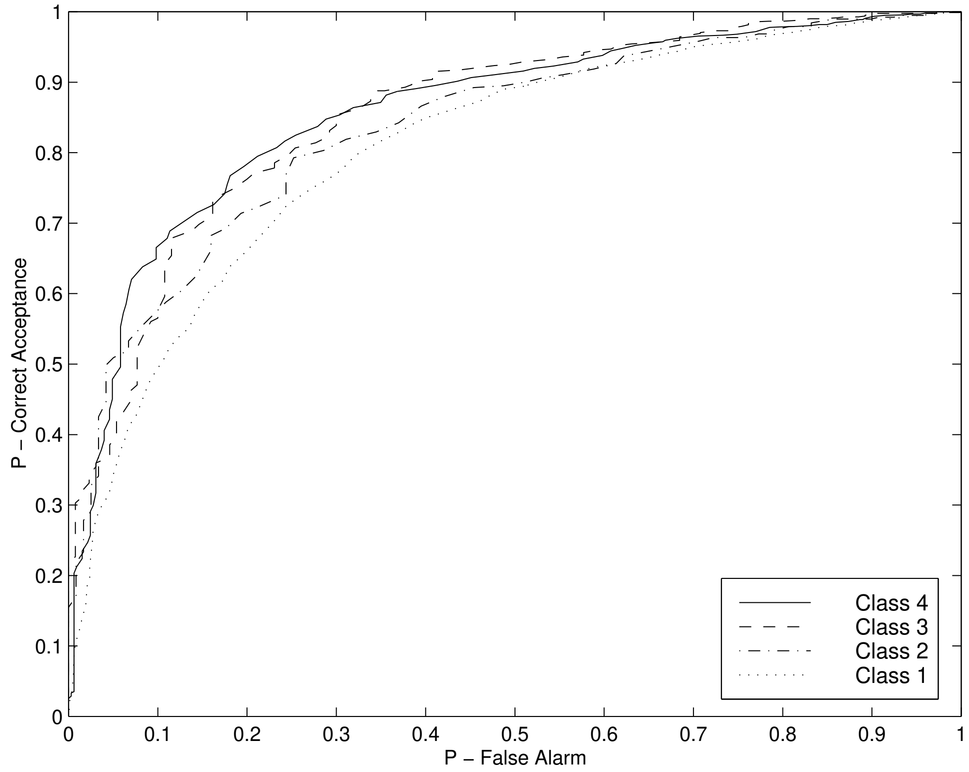

Figure 4.1: Relative ROC performance of the word level confidence score F_{score} for the various word classes defined in this thesis.

new classes, *class high* and *class low*, are combinations of the original classes. Class *high* is the result of merging class 3 into class 4 and similarly class *low* is the result of merging class 2 into class 1. This merging of classes not only smoothes out the data, since each class has more tokens, but it also simplifies the analysis. The number of tokens in these two classes is shown in Table 4.2. Class high now corresponds to all the words with high content value and class low corresponds to words with low content value. The Figure 4.2 shows the relative performance of these two classes using the feature F_{score}.

Since misrecognizing words which have little or no meaning matters less, it is fine that the performance of the confidence scores is worse for the content class *low*. The better performance of the confidence metrics on the content class *high* words can be partially attributed to the difference in the lengths, and acoustic confusability of the words in each class. The words in the content class *high* tend to be longer and

Word Class	Number of Tokens
Class Low	8281
Class High	2798

Table 4.2: Number of tokens in in content class *high* and *low*. Content class *low* corresponds to the original content classes 1 and 2, and content class *high* corresponds to the original content classes 3 and 4.

acoustically more distinct than their counter parts in content class *low*. Because of the manner in which the confidence metrics for words are calculated in this thesis, one landmark at a time, the longer the word the more accurate of a measure of confidence is achieved. In a short word the confidence scores can be harmed by phone level outliers, whereas in a longer word anomalies can be averaged out yielding more robust confidence estimates.

Figure 4.3 illustrates the performance of the word class *high* in respect to the average for all the words. The performance of the word content class *high*, consistently exceeds the performance of all the words.

The results shown in Figure 4.3 are encouraging. However, the difference in performance based on word content class was less dramatic for the single best performing feature C^G_{nll}. This may be attributed to the fact that F_{score} is derived from multiple features, some of which may work well only on content words, which are generally longer. Features like the $\tilde{\sigma}_{map}$ and σ_{nll} are become more robust as words get longer, which could account for increased performance on high content words. Since the C^G_{nll} is less dependent on the lengths of the words, similar difference in performance between high and low content words can not be seen. The *high* content class still performed better than the *low* content class at most points, although the performance of the classes is roughly equal at some operating points. Figure 4.4 illustrates the relative performance of the C^G_{nll} score for content type *high* and *low*.

The difference in performance between F_{score} and C^G_{nll} for all words was not very significant as shown in Figure 3.3, however, with the content type *high* the difference is somewhat more pronounced. Figure 4.5 illustrates the difference in performance between F_{score} and C^G_{nll} for words with content type *high*. The figure illustrates the real world advantage of using FLDA to combine multiple features. Table 4.3 shows the figure of merit performance values for F_{score} and C^G_{nll} on various content classes.

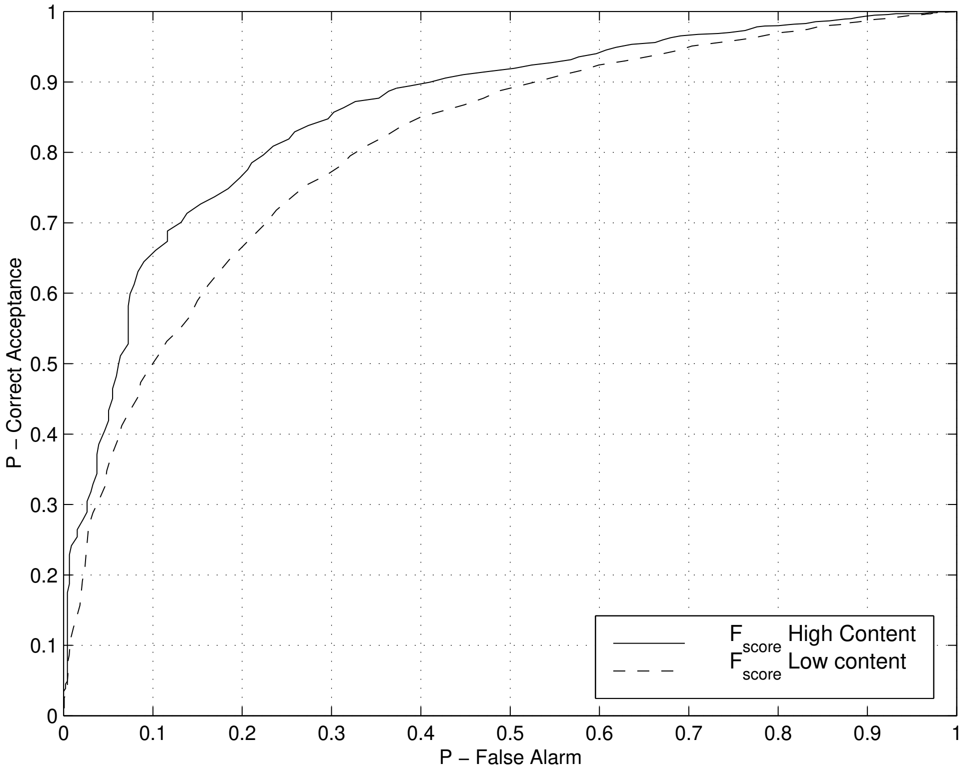

Figure 4.2: Relative ROC performance for the word level score F_{score} on word classes high and low.

Feature	**Content Type**	**Threshold**				
		0.5	0.6	0.7	**0.8**	0.9
F_{score}	High	0.7403	0.6931	0.6233	**0.5249**	0.3500
F_{score}	Low	0.6543	0.6000	0.5298	**0.4311**	0.2752
F_{score}	All	0.6746	0.6209	0.5500	**0.4502**	0.2889
C^G_{nll}	High	0.6446	0.5887	0.5182	**0.4297**	0.3156
C^G_{nll}	Low	0.6265	0.5700	0.5019	**0.4102**	0.2749
C^G_{nll}	All	0.6276	0.5715	0.5032	**0.4114**	0.2826
chance		0.25	0.20	0.15	0.1	0.05

Table 4.3: Figure of merit performance values for F_{score} and C^G_{nll} on content classes *high*, *low*, and all words.

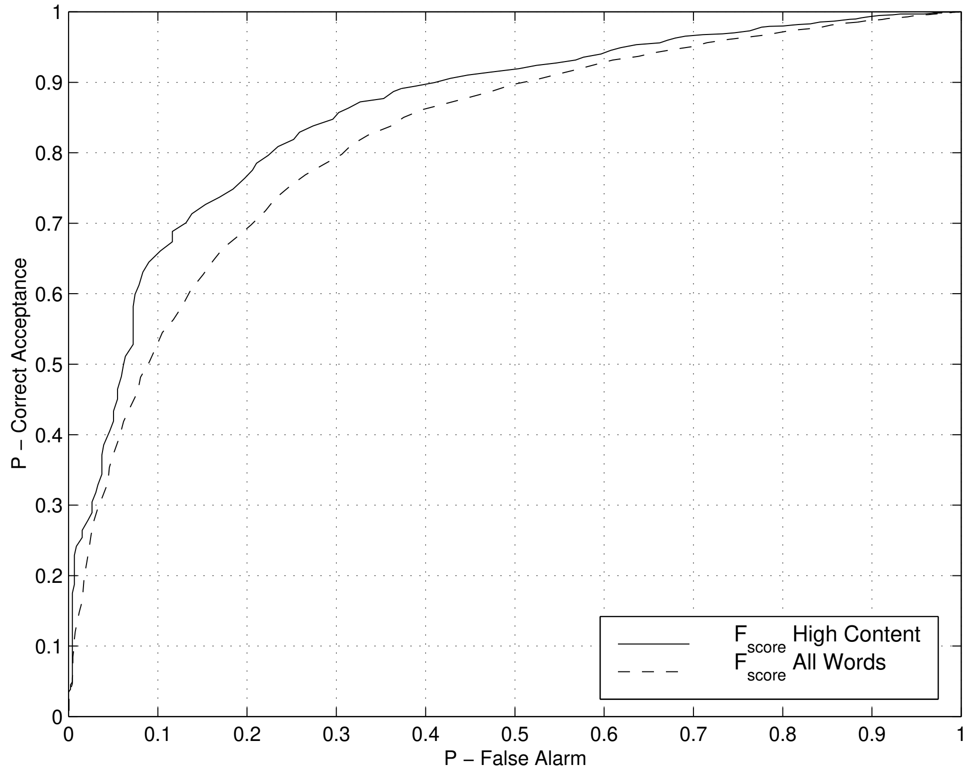

Figure 4.3: Relative ROC performance of F_{score} on content type *high* and on all words.

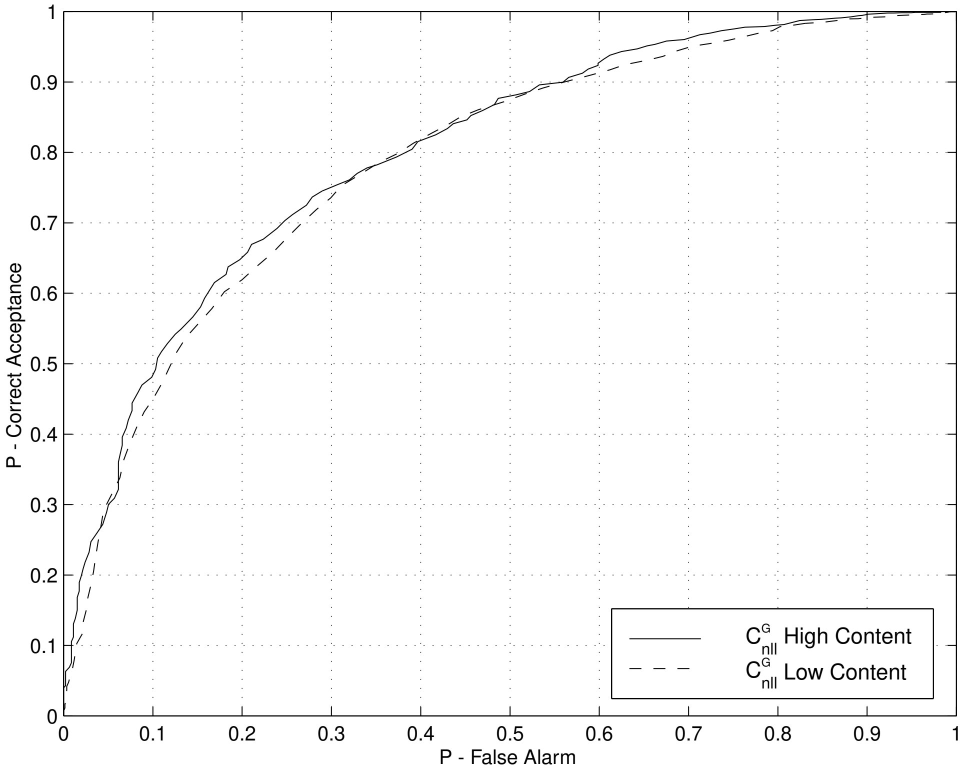

Figure 4.4: Relative ROC performance of C_{nll}^{G} for content types high and low.

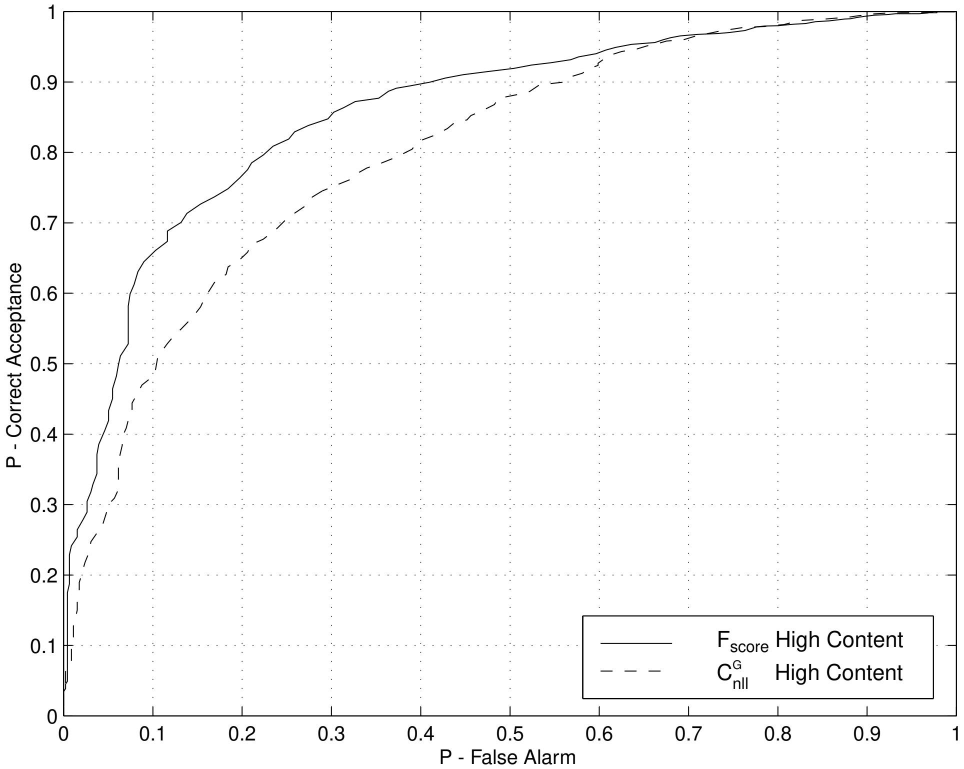

Figure 4.5: Relative ROC performance of F_{score} and C_{nll}^{G} for content type high.

Chapter 5

Catch-all Model

Efficiency is a key issue in any real time system. Incorporating confidence scores into such a system requires efficiency in the part of the confidence score calculation. In this thesis, a *catch-all* model has been used in the calculation of confidence features as described in Chapter 2. Because of its size this *catch-all* model is computationally inefficient. This chapter describes the approach taken in this thesis for improving the efficiency of calculations involving the *catch-all* model.

5.1 Catch-all Model Description

The biggest hindrance to performance is related to the *catch-all* model's size in terms of the number of Gaussian components in the mixture model. The *catch-all* model is created by pooling the mixture Gaussian components from the entire set of JUPITER boundary models. The mathematical description, already mentioned in Chapter 2, is shown in Equation 5.1.

$$\mathrm{p}(\vec{x}) = \sum_{j=1}^{N_c} \mathrm{p}(\vec{x}|c_j)\mathrm{P}(c_j) \tag{5.1}$$

In Equation 5.1 N_c is the number of JUPITER boundary model classes, $\mathrm{p}(\vec{x}|c_j)$ is the likelihood of the observation $\vec{x}$ given class c_j, and $\mathrm{P}(c_j)$ is the prior probability of class c_j. The likelihood $\mathrm{p}(\vec{x}|c_j)$ can be further broken down to additive components which represent the contributions of the individual Gaussians which are used to represent the model for class c_j, this is shown in Equation 5.2. In the equation $N_{gauss}(j)$ is the number of Gaussians modeling class j, $w_{k,j}$ is the weight, and $g_{k,j}(\vec{x})$ is the Gaussian score for the observation $\vec{x}$ for the kth Gaussian in the jth model.

$$\mathrm{p}(\vec{x}|c_j) = \sum_{k=1}^{N_{gauss}(j)} w_{k,j} g_{k,j}(\vec{x}) \tag{5.2}$$

From the above description, the number of Gaussians describing the *catch-all* model $N_{catch-all-gauss}$ is the sum of the number of Gaussians for each of the classes as shown in Equation 5.3. Where K is the number of classes and $N_{gauss}(j)$ is the number of Gaussians modeling the jth class.

$$N_{catch-all-gauss} = \sum_{j=1}^{K} N_{gauss}(j) \tag{5.3}$$

The *catch-all* model in this thesis contains 11433 Gaussian components, which is much larger than the individual JUPITER boundary models which are generally made of less than 50 Gaussians. The number of Gaussian components makes the model very inefficient and impractical for real-time systems. To get around this problem, a method for approximating the model with a smaller number of Gaussians was proposed and is described in Section 5.2.

5.2 Catch-all Model Reduction

The process of reducing the *catch-all* model size involves an iterative bottom-up clustering process of finding the most similar pair of Gaussians and then combining them. On each iteration two Gaussians most similar to each other are found and then combined into a new Gaussian. The similarity measure used in this thesis is a weighted *Bhattacharyya* distance, the general form of which, $B_{distance}$, is shown in Equation 5.4.

$$B_{distance} = -\log \int \sqrt{P_1(x) P_2(x)} dx \tag{5.4}$$

The *Bhattacharyya* distance behaves as a measure of overlap between two Gaussians. The value for the distance ranges between 0 and ∞, corresponding to a full overlap and no overlap between the Gaussians respectively. In practice, since the Gaussians are in the same space, there is always at least a small amount of overlap between the Gaussians and the distance metric never goes to ∞. The specific implementation of the *Bhattacharyya* distance metric for Gaussians yields Equation 5.5, where μ_1 and μ_2 are the means of the Gaussians and Σ_1 and Σ_2 are the covariance.

$$B_{distance} = \frac{1}{8}(\vec{\mu}_1 - \vec{\mu}_2)^{\mathrm{T}} \left(\frac{\Sigma_1 + \Sigma 2}{2}\right)^{-1} (\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2} \ln \frac{|(\Sigma_1 + \Sigma_2)/2|}{|\Sigma_1|^{\frac{1}{2}} |\Sigma_2|^{\frac{1}{2}}} \tag{5.5}$$

The *Bhattacharyya* distance is scaled so that similarly weighted Gaussians are more likely to be combined. This prevents a single high variance Gaussian from continuously absorbing neighboring Gaussians, and growing in weight, while outliers remain unabsorbed. The goal is to compress the acoustic space evenly so that the entire space is covered with reasonable resolution. The weighting scalar B_{scale} is a function of the weights of the Gaussians, w_1 and w_2 respectively, as shown in Equation 5.6.

$$B_{scale} = \sqrt{\frac{w_1^2 + w_2^2}{2w_1w_2}} \tag{5.6}$$

The B_{scale} exhibits behavior which satisfies the above goal, as $w_1 \rightarrow w_2$ then $B_{scale} \rightarrow 1$. Conversely, as $w_1 \gg w_2$ or $w_1 \ll w_2$ then $B_{scale} \rightarrow \infty$. This behavior causes the weighted distance metric B_{SD}, as shown in Equation 5.7, to exaggerate distances between Gaussians with big differences in weight and thereby accomplishes the goal that was set out.

$$B_{SD} = B_{scale}B_{distance} \tag{5.7}$$

After calculating the B_{SD} between every pair of Gaussians, the pair with the lowest B_{SD} is combined to form a new Gaussian. The parameters for the new Gaussian are derived from the parameters of the Gaussians from which it is born. The weight of the new Gaussian w_{new} is equal to the sum of the weights of its Gaussian parents.

$$w_{new} = w_1 + w_2 \tag{5.8}$$

The mean of each dimension of the new diagonal Gaussian is a weighted sum of the means of the parent Gaussians, normalized by the sum of the weights of the parent Gaussians as shown in Equation 5.9.

$$\mu_{new} = \frac{w_1\mu_1 + w_2\mu_2}{w_1 + w_2} \tag{5.9}$$

And the new variance σ_{new} for each dimension, is a weighted sum of the mean adjusted variances of the parents as shown in Equation 5.10, where σ_1 and σ_2 corresponds to the variances of the parents and μ_{new} is as described in Equation 5.9.

$$\sigma_{new} = w_1\left(\sigma_1(\mu_1 - \mu_{new})^2\right) + w_2\left(\sigma_2 + (\mu_2 - \mu_{new})^2\right) \tag{5.10}$$

After the new Gaussian is defined, it is added to the Gaussians describing the *catch-all* model and the parents from which it was created are removed, effectively reducing the number of Gaussians by one. The iteration is then repeated as many times as required to achieve the desired level of reduction in size.

5.2.1 Implementation Details

The estimation of the *catch-all* model is not without a penalty. A small amount of theoretical performance is given up and the C_{map} and C_{nll} scores are slightly altered. Namely, the C_{map} score is no longer constrained to be between 0 and 1. The new range for the C_{map} score can be between 0 and some value much greater than 1. However the distribution of the scores in this research heavily favor scores between 0 and 10, very few scores exceed a value of 10, but when they do they can exceed it by an extremely large margin. This kind of distribution of scores can be difficult to work with. To simplify this problem, without losing the ability to differentiate between high scores, a non-linear transformation was applied to the scores. A new score $\tilde{C}_{map}$ was calculated by mapping score between 0 and 2 linearly to values between 0 and 2, and values greater than two get mapped as logs of the amount greater than 1 plus 2, as seen in Equation 5.11.

$$\tilde{C}_{map} = \begin{cases} C_{map} & \text{if } 0 \leq C_{map} \leq 2 \\ 2 + \log\left(C_{map} - 1\right) & \text{otherwise} \end{cases} \tag{5.11}$$

Because the C_{nll} score is in the log domain, even large variances in the scores get scaled to a reasonable range. Therefore no additional adjustments were required for calculating the C_{nll}. The F_{score} scores calculated with reduced *catch-all* models use $\tilde{C}_{map}$ in place of all the C_{map}.

5.2.2 Performance

The method described above was used to reduce the size of the *catch-all* model by 75, 95, 99 and 99.5%. Each decrease in model size, gives up a small amount of theoretical performance. In theory, there is no way the performance of a reduced model can exceed the performance of the full model. As such, any apparent performance increases in the reduced models are due to random statistical discrepancies and should correct themselves given a large enough set of data. Figure 5.1 illustrates the relative performance of the reduced models on the F_{score} for all content types.

Only slight performance degradation is expressed even when the *catch-all* model is reduced in size by 99.5%. Figure 5.2 displays a sub-section of Figure 5.1 to better illustrate the relative performance of the reduced models.

The relative performance of the 99.5% reduced model with the non-reduced *catch-all* model is shown in Figure 5.3. Table 5.1 shows the figure of merit measures for the

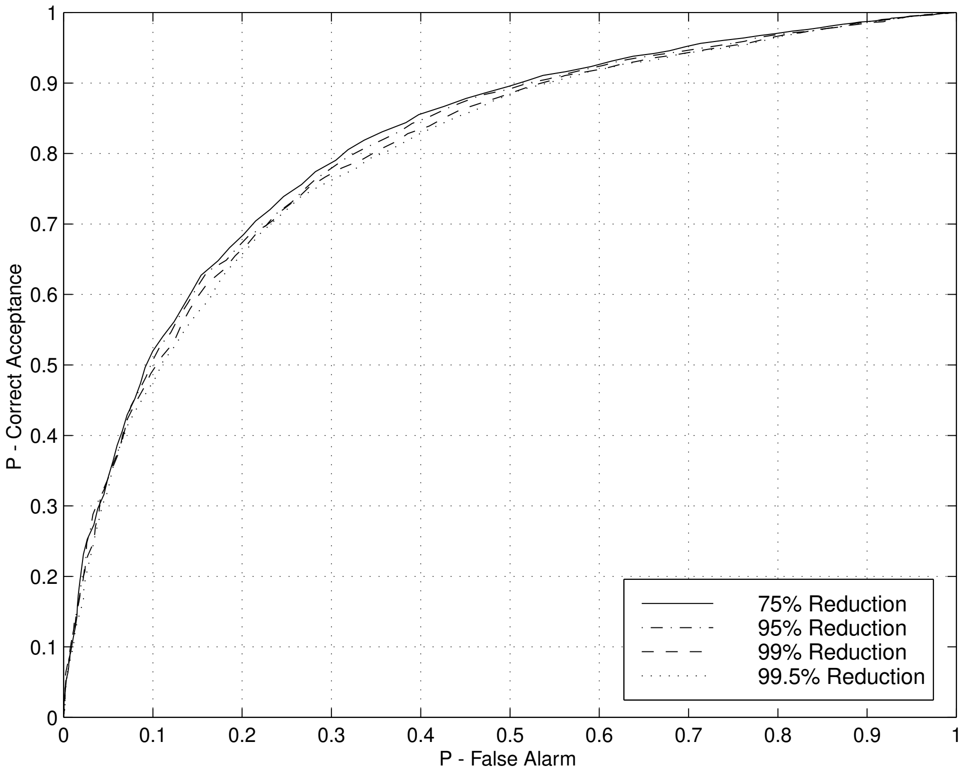

Figure 5.1: Relative ROC performance of reduced *catch-all* models on F_{score} for all words

curves shown in Figure 5.1, Figure 5.2 and Figure 5.3.

This relatively small loss in performance carries over to the content words as described in Chapter 4. The relative performance of the reduced models for F_{score} on high content words is shown in Figure 5.4. Table 5.2 shows the figure of merit measures for the curves shown in Figure 5.4.

For the high content words, at a 90% detection rate, the performance of the 99.5% reduced model appear equal to that of the non-reduced model. This is an encouraging result, although it is difficult to ascertain whether or not the performance of the are in fact equal at that point, or if it is simply a nuance of the testing data. The number of tokens in the high content category is too small to make this distinction for sure.

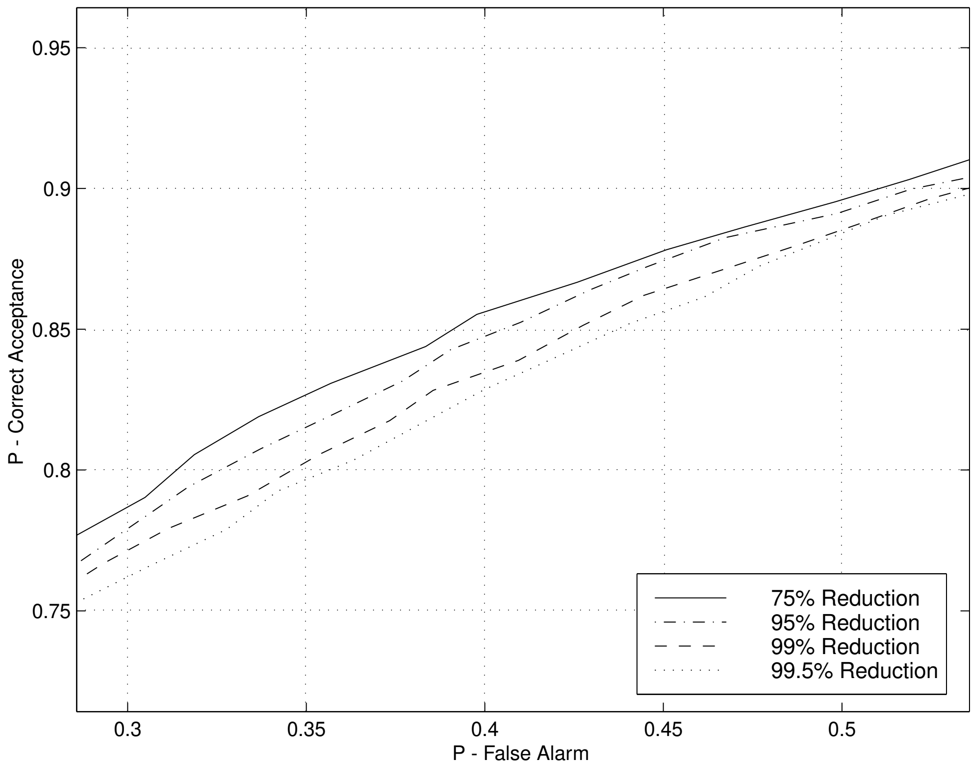

Figure 5.2: A close-up of the ROC curves, on F_{score} for all words, showing the effects of reducing the size of the *catch-all* model via estimation.

Feature	**Threshold**				
	0.5	0.6	0.7	**0.8**	0.9
No Reduction for All Content	0.6746	0.6209	0.5500	**0.4502**	0.2889
75% Reduction for All Content	0.6678	0.6139	0.5433	**0.4451**	0.2879
95% Reduction for All Content	0.6578	0.6024	0.5304	**0.4316**	0.2743
99% Reduction for All Content	0.6469	0.5910	0.5177	**0.4161**	0.2627
99.5% Reduction for All Content	0.6398	0.5839	0.5106	**0.4092**	0.2587

Table 5.1: Effects of *catch-all* model reduction in performance based on the figure of merit.

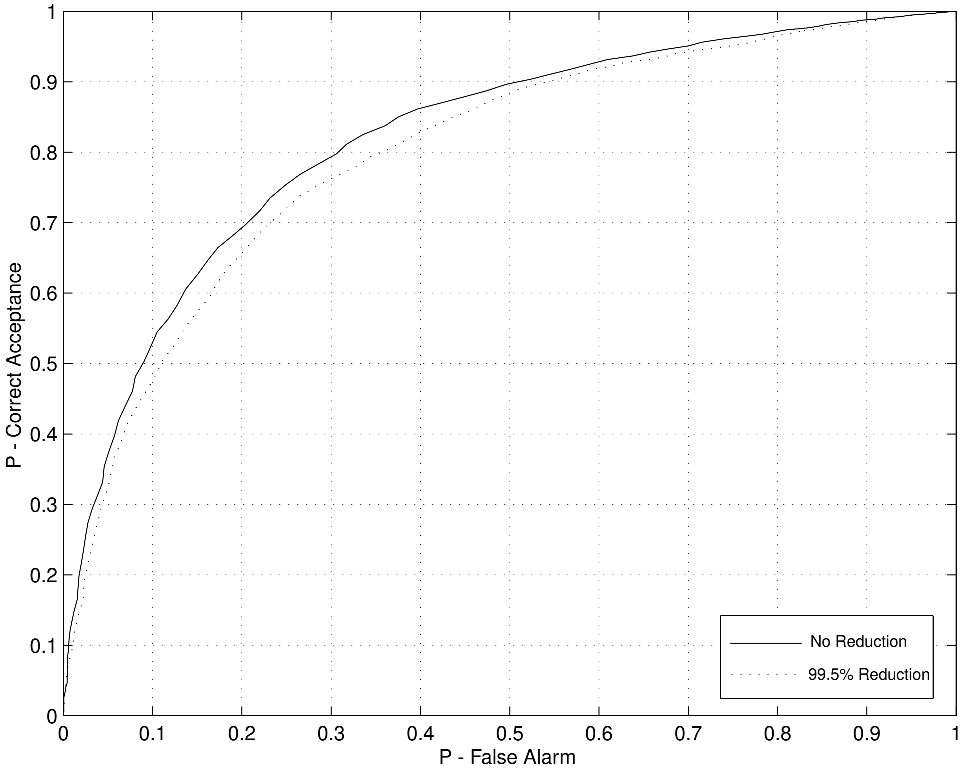

Figure 5.3: An ROC illustration of the relative F_{score} performances for the *full* and 99.5% reduced *catch-all* models on all words.

Feature	**Threshold**				
	0.5	0.6	0.7	**0.8**	0.9
No Reduction for High Content	0.7403	0.6931	0.6233	**0.5249**	0.3500
75% Reduction for High Content	0.7344	0.6859	0.6174	**0.5200**	0.3514
95% Reduction for High Content	0.7266	0.6768	0.6050	**0.5055**	0.3330
99% Reduction for High Content	0.7187	0.6680	0.5962	**0.4938**	0.3219
99.5% Reduction for High Content	0.7104	0.6593	0.5882	**0.4876**	0.3194

Table 5.2: Effects of *catch-all* model reduction in the figure of merit performance on high content words.

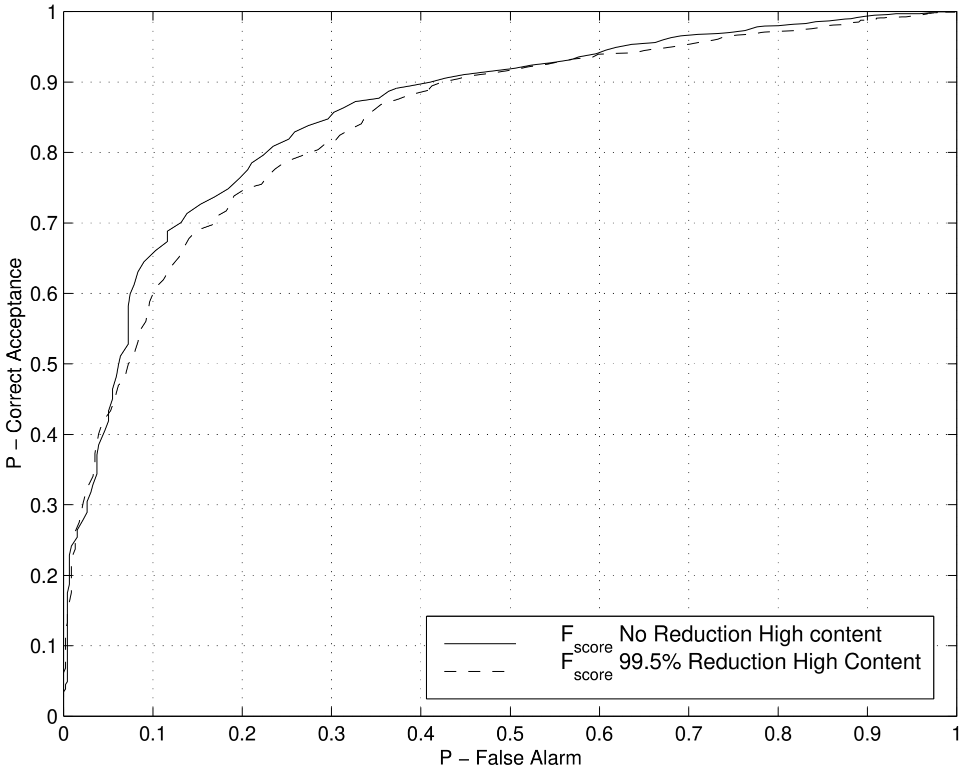

Figure 5.4: Relative ROC performance of the *full* and 99.5% reduced *catch-all* models on high content words.

Chapter 6

Utilizing Confidence Scores

There are many practical applications for word level confidence scores. Work has been done in applying the scores for re-scoring *N-best* lists [16, 15], improved back-off strategies for language modeling, recognition performance prediction [23], and many others. The confidence scores developed in this thesis are valid for use with these applications. However, due to limitations in time this thesis attempts to apply confidence scores for only one application, improving parsing performance. The process of incorporating the confidence scores into the natural language processor TINA is discussed in Section 6.1.

Due to time constraints an in depth analysis of the confidence scores for the purpose of user feedback was not possible. None the less, much thought was given to the possibilities surrounding user feedback. While an actual implementation was not possible, Section 6.2 describes the concepts and ideas surrounding this topic.

6.1 Improved Parsing

The natural language processing unit TINA receives a *N-best* list of sentences from the SUMMIT recognizer. The *N-best* list consists of *N* recognition hypotheses, where each hypothesis is a string of words with no additional information attached. This list of utterances is re-organized into a word graph which is used for the speech understanding task. In the process of building the word graph, scores indicative of confidence in each word are attached to arcs in the word graph. Before the availability of actual acoustic confidence scores, an *ad hoc* method for indicating some confidence like measure was calculated for each word. The goal is to replace this *ad hoc* scoring with actual confidence scores.

6.1.1 Ad Hoc Scoring

Prior to the development of word level confidence scores an *ad hoc* method for expressing word arc confidence was used. The *ad hoc* method uses the location of an utterance in the *N-best* list, and the number of times a word occurs, as the basis for word arc scores.

The *ad hoc* scores are calculated as the *N-best* list is collapsed into a word graph. The utterances in the *N-best* list are weighted based on their location in the list. The weight is the highest for the best (1st) hypothesis in the *N-best* list, and then decreases linearly with each subsequent utterance. For example, an *N-best* list with N utterances and a weight of w_0 on the 1st utterance, the weight for the nth utterance is then $w_0 - n + 1$, as shown in Equation 6.1. In TINA, the weight of the best hypothesis on the *N-best* list w_0 is hand picked, and has been determined via experimentation to be 16.

$$w_n = w_0 - n + 1 \tag{6.1}$$

The *ad hoc* confidence measure associated with each arc in the word graph is the sum of the weights w_n for each occurrence of a word in a specific locations. Figure 6.1 shows an example *N-best* list, for N equal to 3, and the word graph to which it is collapsed. The letters *A, B, C, etc.* represent hypothesized words in the utterances, and the numbers to the left of the *N-best* utterances indicate the utterance weights w_n. The word graph under the *N-best* list is labeled in the form LABEL:SCORE, where LABEL is the hypothesized word and SCORE is the *ad hoc* score associated with that arc.

The *ad hoc* confidence score, mathematically referred to as $S_{ad-hoc}(i)$, for word arc i is shown in Equation 6.2,

$$S_{ad-hoc}(i) = \sum_{n=1}^{N_{nbest}} w_n C_n \tag{6.2}$$

where N_{nbest} is the number of *N-best* utterances, w_n is the weight for utterance n, and C_n is the number of times the word occurred in the specific location in utterance n. Since multiple instances of a word in an utterance are considered different words, the C_n is constrained to be 0 or 1.

This scoring method is motivated by the correlation between the number of times a word occurs in the hypotheses and recognition accuracy. The more times a particular word appears in the *N-best* hypotheses, the more likely the word is in fact correct.

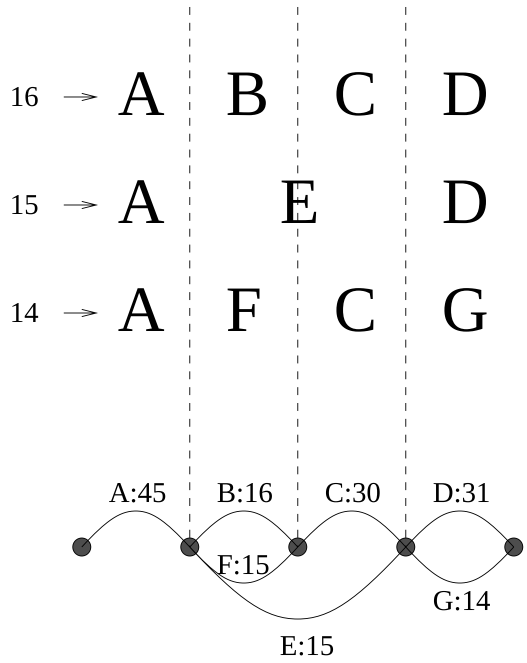

Figure 6.1: Illustration of *N-best* list and the corresponding word graph

By decrementing the weight for each lower consecutive hypothesis in the *N-best* list, the ordinal ranking of the hypotheses in the *N-best* list is accounted for. This *ad hoc* score could be incorporated into confidence calculation, however due to limitations in time this was not done in this thesis, and it is unclear how great of an effect it would have on the performance.

6.1.2 New Scoring

The goal is to replace the *ad hoc* scoring method with some method that utilizes the confidence scores described in this thesis. Several methods for replacing the *ad hoc* scores were explored. The simplest method was to replace the utterance level weight w_n with the actual confidence scores associated with each word. The resulting new scores, shown in Equation 6.3, vary slightly from the *ad hoc* scores in Equation 6.2.

$$S_{new}(i) = \sum_{n=1}^{N_{nbest}} C_{word}(i, n) C_n \tag{6.3}$$

In Equation 6.3 w_n is replaced with $C_{word}(i, n)$ which is the word level confidence score for the word i in the utterance n, and the remaining terms remain as in Equation 6.2.

A slight variation on this method is combination of the *ad hoc* method with the new confidence scores. The confidence scores are now scaled by the utterance level weight, so that the confidence score have more weight the earlier they appear in the *N-best* list. The resulting new score $S_{new-combo}(i)$ utilizes the w_n scores as described in Equation 6.1 and a word level confidence measure $C_{word}(i,n)$. The form of the new combination method is shown in Equation 6.4, where the terms remain the same as in previous equations.

$$S_{new-combo}(i) = \sum_{n=1}^{N_{nbest}} w_n C_{word}(i,n) C_n \tag{6.4}$$

The behavior of $S_{new-combo}$ is expected to be similar to a performance of a S_{new} which uses C_{word} scores that incorporate the *ad hoc* scores into the confidence calculation.

Because the decreasing weight method described in Equation 6.1 makes sense, a slight variation of the method was also explored. Instead of a linear decrease in the weight for each subsequent *N-best* hypothesis, a non linear weight was explored. The new weight w_{n-new} is a exponential function with varying rates of decay as shown in Equation 6.5, where n is the nth hypothesis and α is a constant between 0 and 1.

$$w_{n-new} = \alpha^{n-1} \tag{6.5}$$

By varying α various rates of decay can be used, the appropriate value can be found via experimentation. Substituting the new weight w_{n-new} for the ad-hoc weight function w_n yields a new scoring method shown in Equation 6.7.

$$\begin{aligned} S_{new-weight}(i) &= \sum_{n=1}^{N_{nbest}} w_{n-new} C_{word}(i,n) C_n && (6.6) \\ &= \sum_{n=1}^{N_{nbest}} \alpha^{n-1} C_{word}(i,n) C_n && (6.7) \end{aligned}$$

6.1.3 Results

The *ad hoc* method for scoring the word arcs in a word graph was determined through a process of experimentation. The performance of the *ad hoc* method has been reasonably good and has not been significantly exceeded via other methods. Incorporating the confidence scores into word graph scoring provided a performance level roughly equal to that of the *ad hoc* method.

The following Table 6.1 outlines performance of the various confidence based scoring methods described above. The results are reported for C^{A}_{nll} as the confidence measure C_{word}, and they are reported relative to the performance of the *ad hoc* method for word graph scoring. The columns *Better Und.* and *Worse Und.* describe the number of times a metric performed better and worse respectively in terms of understanding. Understanding performance is evaluated on a *key-value* representation of the recognition result [4, 13]. The columns *Better Rec.* and *Worse Rec.* describe the number of times a metric performed better and worse in terms of recognition.

Scoring Method	**Better Und.**	**Worse Und.**	**Better Rec.**	**Worse Rec.**
$S_{new-weight}$;$\alpha = 2/3$	5	3	25	20
S_{new}	5	7	31	26
S_{ad-hoc}	0	0	0	0
$S_{new-combo}$	28	39	62	106

Table 6.1: Parsing performance, in terms of the number of differences in the outcomes in recognition and understanding between new methods and the original *ad hoc* method, for various word graph weighing techniques.

Based on this crude quantitative evaluation $S_{new-weight}$, with an α value of $2/3$, is the best performing method for scoring the word graph. However, while this scoring results in improvements in both the understanding and recognition sides, the number of times the new scores outperforms the old *ad hoc* method appears insignificant.

As mentioned above, there were instances where the new scoring method provided a much better result which was encouraging, unfortunately this did not occur very often. Of the 5 times that the best performing new scoring method $S_{new-weight}$ performed better, in terms of understanding, only twice the performance increase was due to high content words. Increases in performance due to low content words do not add much value, as they add little information regarding the users query. Table 6.2 breaks down the differences in performance in terms of word content values for the $S_{new-weight}$ and $S_{ad-hoc}(i)$.

A similar breakdown of recognition performance, into word classes, is shown in Table 6.3. Similarly, increased performance on low content words means little, while the performance differences on high content words are more significant. Unfortunately

	High Content	Low Content
Und. Better	2	3
Und. Worse	1	2

Table 6.2: Understanding performance in terms of word content

there appear to be more cases where high content words are misrecognized than when they are correct.

	High Content	Low Content
Rec. Better	6	19
Rec. Worse	9	11

Table 6.3: Recognition performance in terms of word content

As a whole the differences in performance between the new methods and the old *ad hoc* method are insignificant. This might be expected as due to time constraints, only the surface of the word graph scoring problem was scratched. The initial result are encouraging and future work will likely provide increased performance via an appropriate use of the confidence score.

6.2 User Feedback

While modifications to user feedback were not implemented in this thesis, much thought was given to possible benefits and problems that may arise from incorporating the confidence scores into the feedback process. The general idea is that confidence scores may be useful in detecting problems with recognition and the system maybe able give useful feedback to the user which may help correct possible recognition problems.

User feedback modification involves critical two parts, figuring out when to prompt user for feedback and what kind of feedback to give. The first part can be addressed by analyzing the result of the parsing process. The parsed result contains information about the nature of the words. Using the parsed result as a guide to which words are important, the confidence scores can be looked up for key words. The important words are words which are critical for deriving correct responses to users queries, these generally the words in the *content type high* class as described in Section 4.2. If a key word has a low confidence score, then the user can be asked to confirm or repeat that word. A problem with this approach it is only able to address problems with misrecognitions within a word class. If a city name is misrecognized as a pronoun, or some other low content type word, this type of analysis is unable to locate or alleviate the problem because only the confidences of words in content class high are considered.

The manner in which the user is asked to confirm or repeat information is the second critical part of the user feedback. The specific wording used to prompt the user for more information has great effect the type of response the user will give. There are many forms of feedback that are valid, this thesis discusses two of them. They are both forms of confirmation and vary in their approach and complexity.

The first method is to repeat the recognized word along with a question about the validity of that word. For example, a system which has a low confidence in the word *Boston* may say to the user: "*I'm sorry, did you mean Boston,*" to which the user may reply yes or no. A benefit of this method is that the words *yes* and *no* are acoustically very distinct and thus the confirmation can be made with high accuracy. A problem with this method is that the person may not answer with a simple *yes* or *no* answer. Especially if the recognizer has made a mistake and the person wants to correct it. For example, a user may respond to the above feedback by saying: "*No, I said Austin,*" or "*No, I did not mean Boston. I meant Austin.*" If the person responds in a verbose manner then the confirmation may become more difficult to make. A verbose answer requires a new recognition and parsing process which is complex and may not itself be correct. To alleviate the problem with verbose responses, the user can be directed to answer in a specific fashion. Instead of giving feedback which may lead to a potentially open ended response, the feedback can be worded in a fashion which limits the likely responses. For example, the system may instead say: "*I'm sorry, did you mean Boston. Yes or no?*" This may slow down the dialog as additional steps are now necessary to resolve a *no* answer, however the confirmation performance is likely to increase.

The second approach is to ask a more open ended question like: "*I'm sorry, could you repeat the location you were interested in.*" A query of this sort may yield a wide range of responses, however they should be consistent in that they all contain the location of interest which can then be compared to the original recognition result. The benefit of this type of feedback is that the answers are more likely to be of a similar format. Because people are not asked to answer a question, which has multiple valid answers, the responses can be more predictable. The downside to this type of confirmation is that it is not as robust as a simple *yes* or *no* classification. This method is especially sensitive to unknown words. Words that are not in the vocabulary of the system will cause serious problems with this type of feedback. The system will be unable to hypothesize the correct location regardless of how many times the user repeats themselves. With a directed dialog method described above, it is possible to stop asking the yes/no type confirmations after a certain number of times and simply declare that the location is unknown. With the second approach, since no confirmation is asked for, it is difficult to ascertain whether or not the new hypothesis is still wrong.

Further work is necessary to figure out the optimal way to interpret the users responses to the feedback. For example, a case where the user is asked to repeat the location of interest, if the repeated location differs from the original recognition location, what is the appropriate behavior for the system. Should the system accept the new location as the truth, should it pick the one with the higher confidence score, should it prompt for further confirmation, these are questions which are not easily answered without actual experiments. It is clear that the user feedback problem is not simple and requires much work, but the benefits seem clear and future work is likely to address this problem.

Chapter 7

Conclusion and Future Work

This thesis was motivated by the interesting possibilities that robust confidence scores bring to spoken language systems. Because confidence scores were not readily available, methods for creating confidence scores were first explored. In, and between, the creation of the confidence scores and the final applications of confidence scores lay many steps. On each step in the process, decisions were made and interesting results were observed. This chapter discusses the some of the interesting findings from each step in the process and discusses possible future directions relevant to that matter. As each step builds on the previous ones, improvements on any of the steps will improve the final performance of the confidence scores and their applications. The significant steps in the process are roughly divided along the chapters in this thesis, some of the steps involve processes while others are purely analytical.

This thesis creates word level confidence metrics from phone level confidence scores. Although this is not the only way to derive word level scores, it proved fairly robust. Because this thesis derives word level scores from the phone level ones, the performance of the word level scores is in part limited by the performance of the phone level scores. Similarly the applications which use the confidence scores are limited in their performance by how well the word level scores perform. This performance interdependency in between various levels in the system is why improvements at all levels are crucial to boosting the performance of confidence scores in real world applications.

It is possible to circumvent some of the interdependency by eliminating steps. For example, it is possible to forego phone level scoring by starting the confidence scoring analysis at the word level instead. This is done by utilizing only word level features. Word level features include language model scores, durations and other more complex measurements like A-stabil [18, 21]. Future work will surely address the differences in performance of word level confidence scores based solely on word level features and ones based on phone level features. While it is uncertain which approach is better, it is likely that the best result would be achieved by utilizing both, phone level and word level features.

7.1 Phone Level Confidence Scores

The phone level scoring appeared to work well. However, to make improvements at this level two issues need to be addressed. First, a method for evaluating phone level confidence scores must be created. Second, more phone level features must be proposed and evaluated.

The phone level confidence scores were not evaluated on their own, their performance was inferred from the performance of the word level scores instead. Ultimately, developing a robust method for evaluating the performance of the phone level confidence metrics is a good idea. A possible approach for evaluating phone level confidence scores is to use the result of a forced transcription path as the correct phonetic transcription of an utterance, and then compare the hypothesized phones against this transcription. When the two match the phone is considered correct, and similarly in places of difference the phone is considered incorrect. By evaluating the correctness of phone hypothesis via such a method, a performance evaluation could be performed. Having such a method would make further development and research into phone level confidence features more fruitful.

In addition to a method for evaluating the phone level performance, new features should be proposed and evaluated. Variations in the *catch-all* model would lead to an interesting experiment. For example, instead of using a generic *catch-all* model, various near-miss [2] or boundary specific anti-models could be used [20, 25]. Some of the anti-model research has been motivated by computational issues. Instead of having to use a generic model which is large, smaller computationally efficient normalizing models are used. Each boundary specific anti-model is made of all the models most like the boundary itself. It is unclear which method for improving computation is better, estimating a *catch-all* model, or using small anti-models in the first place. A comparison of the two method should prove interesting. Through further experimentation and with the help of phone level performance evaluation methods, the performance of phone level confidence scoring is likely to increase.

7.2 Word Level Confidence Scores

The word level scores performed well enough that they appear to be useful. A detection rate of about 90% can be achieved for content words while keeping the false alarm rate less than 40% as shown in Figure 7.1. To improve the performance of the word level scores, two straight forward things can be done. First, new features can be proposed which complement the current set of features, and second, new methods for combining the features should be evaluated.

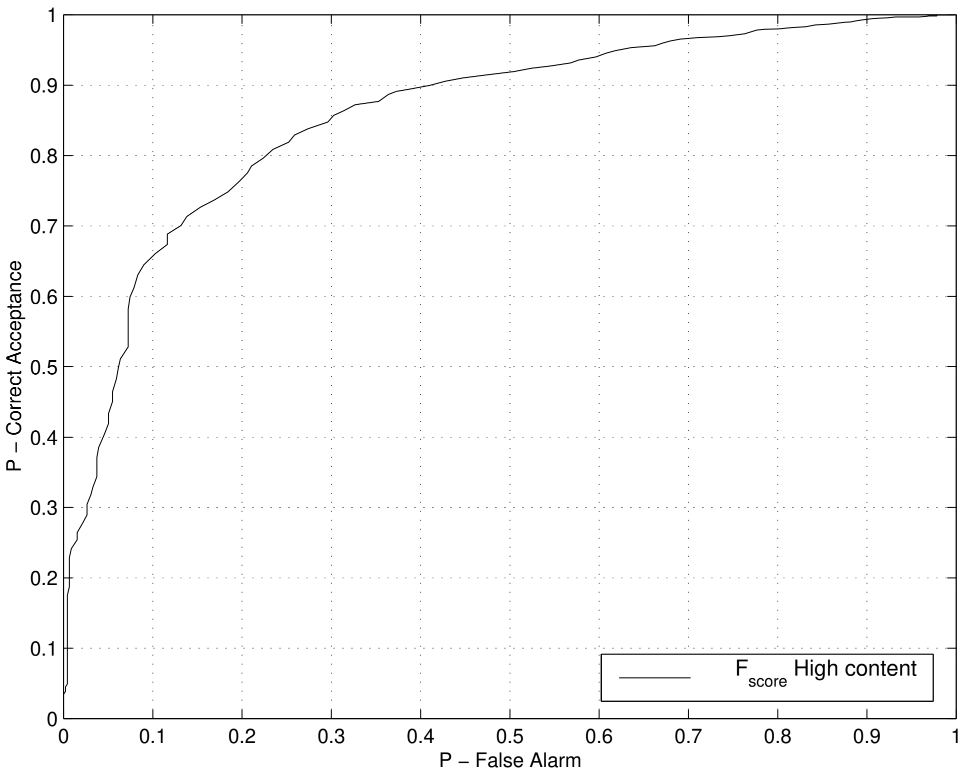

Figure 7.1: Performance of F_{score} on high content words

This thesis limits the word level features to acoustic features derived from the phone level scores and a few word level features like the number of landmarks within the word. By incorporating non-acoustic features like the language model scores, performance of the word level scores is likely to increase. To increase the performance of the features as a collective set, new features should have low correlations with the current features as well as each other. Highly correlated features, even if they themselves work well, complement each other poorly and thus add little to the over all performance of the set of features. This problem is already seen with the set of features explored in this thesis, because the features were all derived from the same underlying acoustic information their cross correlations were also high.

In addition to exploring new features, new methods for combining/evaluating a set of features should be evaluated. This thesis explored two methods for combining/evaluating for this purpose, of which Fisher Linear Discriminant Analysis proved best. There are other methods for combining/evaluating a set of features like neural-networks. Some methods are more sensitive to the amount of training data available than others. Which opens the possibility that with more data the relative performance of various methods for combining/evaluating features may change.

7.3 Performance as a Function of Content Type

A strong correlation between performance of the word level confidence measures and word content type was found. Words with high content value performed systematically better than words with little or no content value. This result was very encouraging in terms of possible future applications of the scores. The higher performance can be attributed to the higher acoustic differentiability between content words. Content words tend to be longer and more distinct than their non-content counter parts. In the future a more in-depth analysis should be made to determine what exactly are the factors that contribute to this difference in performance. It is likely that a better understanding of why scores work better on some words than others will lead to confidence metrics which work better on all types of words.

Creating content specific confidence metrics would lead to an interesting comparison in the future. Based on the content type of a word, one set of features is used for low content words and a different set for high content words.

In itself, the fact that the confidence metrics perform better on high content words can be used to justify using the confidence scores in conjunction with word spotting and other key word related tasks.

7.4 Catch-all Model

The initial size of the *catch-all* model made it inefficient, thus methods for reducing the *catch-all* model size were explored. The reduction of the *catch-all* model was successful. The size of the *catch-all* model was decreased by 99.5% with only a 7% reduction in performance (when evaluated at 80% detection rate). The reduced model provides a good balance between computational efficiency and confidence metric performance. Further reduction in size is not necessary since even todays machines are able to handle the computation with no problems. However, as the computational power of computers increases, this reduction becomes less meaningful and in the future this reduction might be foregone. Although, it is possible that the acoustic models will grow in complexity and size as the computational power increases, and the *catch-all* reduction will remain significant for some time to come.

7.5 Applications of Confidence Scores

While the applications of the confidence scores were the original motivation behind this thesis, only a small amount of work was completed on this front. It is likely that the confidence scoring methods developed in thesis will be used in future confidence score applications. The value of confidence scores in the parsing process is unclear. Small gains were seen in the experiments performed, however a much more in-depth analysis is required to make any conclusive statements regarding that application.

There are many unexplored applications for the confidence scores. For example, user feedback briefly discussed in Section 6.2, appears very interesting and is sure to be investigated further. Applications is the area of confidence scores which is the least understood and studied. It is also the area which is going to see the most new research in the future years. The lessons learned in this thesis will help in understanding the strengths and weaknesses of the metrics and should prove valuable in developing future applications.

7.6 Summary

This thesis set out to explore applications of confidence scores and to create the necessary infrastructure for accessing such scores. This thesis was more successful in setting up the methods for the creating confidence scores and, due to time constraints, less so in developing and analyzing various applications of confidence scores. The confidence scores are now accessible and can be utilized by any element of the GALAXY architecture. However, time constraints limited that amount of work done in the applications side. It is likely that future work will pick off where this thesis left off, thus covering a more in-depth analysis of the applications of confidence scores.

Appendix A

The Corpora and JUPITER Recognizer

The work in this thesis is conducted in the JUPITER [10] weather information domain. JUPITER is a telephone based spoken language system which provides users with weather related information for locations around the world. The system has access to weather information for approximately 500 locations around the world. The amount of available real user data for JUPITER is the biggest motivation for working in this domain. With over 180,000 utterances of recorded data, the amount of JUPITER data far exceed that of all the other domains being developed at SLS. The system has been accessible to the public since 1997 via a toll free phone number. The JUPITER recognizer used in this thesis was trained on 20064 utterances and achieves a word error rate of 19.2% on the *Testing* data described below [9]. The performance of the system varies greatly depending on the utterance content, namely if an utterance is *in-domain* or not. An utterance is considered *in-domain* if it contains *no* out of vocabulary words, partial words, crosstalk, or other disrupting effects. The word error rate drops to around on 11% for *in-domain* utterances and goes up to approximately 65% for out of domain ones.

The development and testing of the confidence scores was done on two sets of data. The development set was initially further divided into two, a training and testing, subsections. In the development phase, the training of FLDA projection vectors and mixture Gaussian models, for combining multiple features, was performed on the testing set. Tests during the development phase were conducted on the testing subsection of the development set. The final result reported in this thesis are on the test set while the entire development set, both the initial training and testing halves, was used for training. Table A.1 shows the respective sizes of above data sets.

Set Name		Utterances	Words
JUPITER Training		20064	122267
Confidence Development	Train	1719	9121
	Test	1718	8957
	Total	3437	18078
Confidence Testing		2405	11339
NL Testing		2391	12318

Table A.1: Sizes of data sets used in this thesis.

For the experiments involving parsing and TINA, as described in Chapter 6, a different set of JUPITER data was used. The last line of Table A.1 describes the data used in those experiments. The experiments with TINA required no training of additional models therefore a separate testing and training set were not required.

Appendix B

ROC Computation

The Receiver Operating Characteristic (ROC) is a method for analyzing the performance of a classification task which makes use of an adjustable threshold. In the context of confidence scoring, words are classified either as correctly recognized or incorrectly recognized. The classification decision is based on the confidence scores associated with each word. If a confidence metric exceeds a threshold then the word is classified as correctly recognized, or if it doesn't then it is classified as incorrectly recognized. Figure B.1 illustrates two hypothetical distributions of confidence metrics and the threshold which is used for classification.

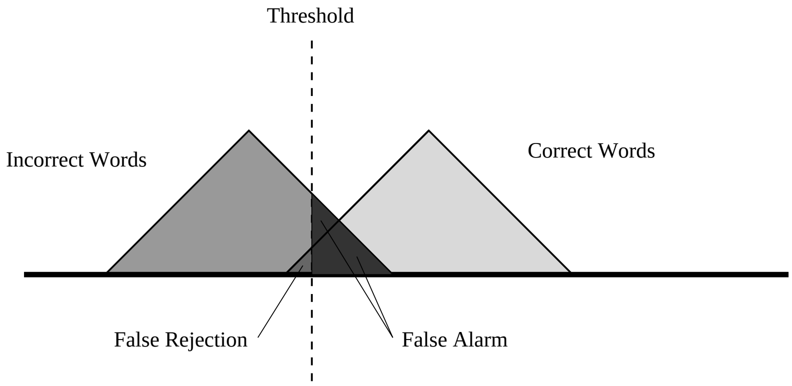

Figure B.1: Hypothetical distribution of confidence metrics

The various outcomes of the classification are shown in Figure B.2. An ROC curve plots the probability of *Detection* on the y-axis and the probability of *False Alarm* on the x-axis as a function of the same threshold. The area under the *incorrect words*' probability density function (PDF), and to the right of the threshold, is equal to the false alarm probability. Similarly the area under the *correct words*' PDF, and to the right of the threshold, is equal to the detection probability.

$$P(\text{detection}) = P(\text{word classified as correct}|\text{word is correct})$$
$$P(\text{false alarm}) = P(\text{word classified as correct}|\text{word is incorrect})$$

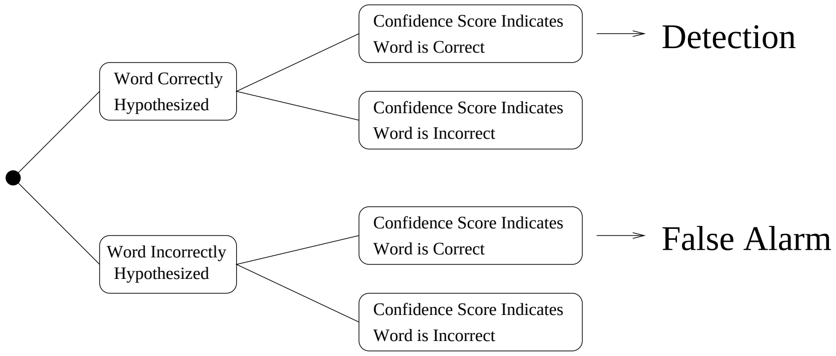

Figure B.2: Possible outcomes of classification

The *Detection/False Alarm* relationship can be mapped out as a function of the decision threshold by varying the threshold and plotting the corresponding probabilities. Figure B.3 shows a typical curve which results from such a process, the top right hand corner of the curve corresponds to a threshold at $-\infty$ and at the bottom left hand corner the thresholds is at ∞.

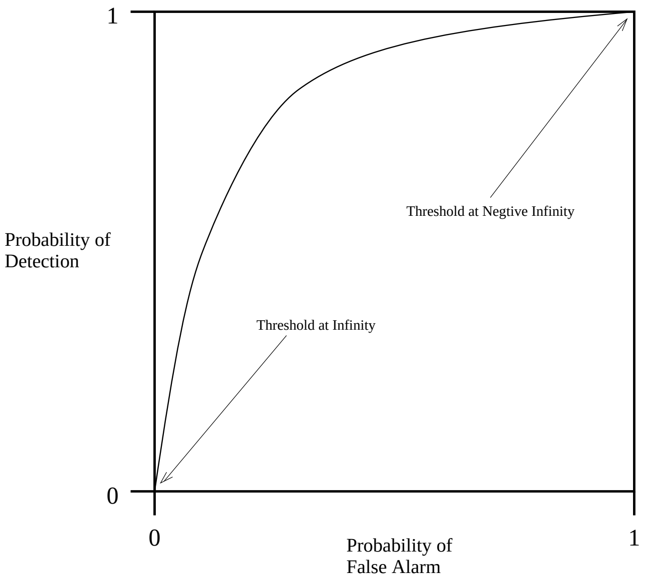

Figure B.3: A typical ROC curve

Appendix C

Word Classes

Here is a listing of the words which fall into the content types 2, 3, and 4. Everything which is not in these content classes is in content class 1.

Content Type 4

aberdeen	abidjan	abilene	acapulco
addis_ababa	afghanistan	africa	ainsworth
akron	alabama	alaska	albany
alberta	albuquerque	algeria	algiers
allentown	alliance	altus	amarillo
america	american_samoa	amman	amsterdam
anaheim	anchorage	anderson	ankara
ann_arbor	anniston	antarctica	antigua
argentina	arizona	arkansas	arlington
aruba	asia	aspen	astoria
asuncion	athens	atlanta	atlantic_city
augusta	austin	australia	austria
b_c	baghdad	bahamas	bahrain
baltimore	bangalore	bangkok	bangladesh
bangor	bar_harbor	barbados	barcelona
basel	baton_rouge	beaufort	beijing
beirut	belarus	belfast	belgium
belgrade	belize	bellingham	bemidji
berlin	bermuda	bern	billings
birmingham	bismarck	block_island	bogota
boise	bolivia	bombay	bonn
bordeaux	bosnia	bosnia_herzegovina	boston

Content Type 4 Cont...

boulder	bowling_green	bradford	brasilia
brazil	breckenridge	bremerton	bridgeport
bristol	britain	british_columbia	british_isles
brookings	brunei	brunswick	brussels
bucharest	budapest	buenos_aires	buffalo
bulgaria	burbank	burlington	burma
cairo	calcutta	calgary	california
cambodia	cambridge	canada	canary_islands
cancun	canton	cape	cape_cod
cape_town	caracas	caribou	casablanca
casper	cedar_city	central_america	chadron
champaign	charleston	charlotte	charlottesville
chattanooga	chicago	childress	chile
china	christchurch	cincinnati	clearwater
cleveland	coeur_d+alene	cologne	colombia
colombo	colorado	colorado_springs	columbia
columbus	concord	concordia	connecticut
copenhagen	costa_rica	cozumel	crestview
crossville	cuba	curacao	cut_bank
cyprus	czech_republic	d_c	dakar
dalhart	dallas	dallas_fort_worth	damascus
danville	dayton	daytona	daytona_beach
death_valley	del_rio	delaware	delhi
denmark	denver	des_moines	detroit
detroit_lakes	devils_lake	dickinson	djibouti
dominican_republic	dothan	dover	dubai
dublin	dubois	dugway_proving	duluth
durango	durham	dusseldorf	dyersburg
eagle	ecuador	edinburgh	edmonton
egypt	el_dorado	el_paso	el_salvador
elko	ely	england	enid
erie	essen	estherville	estonia
ethiopia	europe	everett	fairbanks
fargo	farmington	fayetteville	fiji
finland	flagstaff	florence	florida
fort_collins	fort_de_france	fort_knox	fort_lauderdale
fort_myers	fort_smith	fort_worth	france
frankfurt	franklin	french_guiana	fresno
gage	gainesville	geneva	georgia
germany	gibraltar	glacier_park	glasgow

Content Type 4 Cont...

glens_falls	goodland	grand_canyon	grand_forks
grand_rapids	great_britain	greece	green_bay
greenland	greensboro	greenville	greenwood
grenoble	groton	guadalajara	guadaloupe
guam	guangzhou	guatemala	gulfport
guyana	hagerstown	haiti	hanoi
harare	harrisburg	harrison	hartford
havana	havre	hawaii	heidelberg
helsinki	hibbing	hill_city	hilo
hilton_head	hobart	honduras	hong_kong
honolulu	hopkinsville	hot_springs	houghton
houlton	houston	hungary	huntsville
huron	hyannis	iceland	idaho
idaho_falls	illinois	india	indiana
indianapolis	indonesia	international_falls	iowa
iran	iraq	ireland	islamabad
israel	istanbul	italy	ivory_coast
jackson	jackson_hole	jacksonville	jakarta
jamaica	jamestown	japan	jerusalem
johannesburg	jonesboro	jordan	juneau
kabul	kahului	kalispell	kansas
kansas_city	katmandu	kentucky	kenya
key_west	kingston	kinshasa	knoxville
korea	kuala_lumpur	kunming	kuwait
l_a	la_paz	laconia	lagos
lake_tahoe	las_vegas	latvia	leadville
lebanon	lewiston	lexington	lhasa
libya	lihue	lima	limon
lisbon	lithuania	little_rock	london
long_beach	long_island	longview	los_angeles
louisiana	louisville	lovelock	lubbock
luxembourg	lynchburg	lyon	madras
madrid	maine	malad_city	malaysia
mali	malta	managua	manchester
manhattan	manila	manitoba	marseille
martha+s_vineyard	martinique	maryland	massachusetts
massena	mcalester	mccomb	melbourne
memphis	meridian	mexico	mexico_city
miami	miami_beach	michigan	middle_east
midwest	milan	milwaukee	mineral_wells

Content Type 4 Cont...

minneapolis	minnesota	minot	minsk
mississippi	missoula	missouri	mitchell
mobile	mogadishu	monaco	montana
monte_carlo	montego_bay	monterrey	montevideo
montgomery	montpelier	montreal	morocco
moscow	mount_mckinley	mount_washington	munich
muscle_shoals	myrtle_beach	nairobi	nantucket
naples	nashville	nassau	natchez
nebraska	nepal	netherlands	nevada
new_brunswick	new_caledonia	new_delhi	new_england
new_hampshire	new_jersey	new_mexico	new_orleans
new_york	new_york_city	new_york_state	new_zealand
newark	newfoundland	newport_news	niagara_falls
nicaragua	nice	nigeria	nome
norfolk	north_america	north_carolina	north_dakota
north_myrtle_beach	north_pole	northern_ireland	northwest_territories
norway	nova_scotia	oakland	ogden
ohio	oklahoma	oklahoma_city	olympia
omaha	ontario	oregon	orlando
osaka	oslo	ottawa	p_e_i
paducah	pakistan	panama	panama_city
paraguay	paris	park_city	pendleton
pennsylvania	pensacola	perth	peru
philadelphia	philippines	phnom_penh	phoenix
pierre	pine_belt_region	pine_bluff	pittsburgh
plattsburgh	pocatello	poland	ponca_city
port_angeles	portland	portsmouth	portugal
prague	prince_edward_island	providence	provo
puerto_rico	pyongyang	quebec	quillayute
quito	raleigh	rapid_city	reno
reykjavik	rhinelander	rhode_island	richmond
rio_de_janeiro	riyadh	roanoke	rochester
rocky_mountains	romania	rome	russia
sacramento	saint_kitts	saint_louis	saint_lucia
saint_martin	saint_paul	saint_petersburg	saint_thomas
salisbury	salt_lake_city	samoa	san_antonio
san_diego	san_francisco	san_jose	san_juan
santa_fe	santiago	sao_paulo	sarajevo
saskatchewan	saudi_arabia	savannah	scotland
scotts_bluff	scranton	seattle	senegal

Content Type 4 Cont...

seoul	serbia	seville	shanghai
shreveport	siberia	singapore	sioux_falls
slovakia	snowmass	sofia	somalia
south_africa	south_america	south_carolina	south_dakota
south_pole	spain	spokane	springfield
sri_lanka	stockholm	strasbourg	stuttgart
sumter	sun_valley	suriname	sweden
switzerland	sydney	syracuse	syria
tacoma	tahiti	tahoe	taipei
taiwan	tallahassee	tallinn	tampa
tanzania	taos	tasmania	tehran
tel_aviv	tennessee	texarkana	texas
thailand	thief_river_falls	tibet	timbuktu
titusville	tokyo	tonopah	topeka
toronto	toulouse	trenton	trinidad
tucson	tulsa	tunisia	tupelo
turkey	tuscaloosa	twin_falls	tyler
u_k	u_s	u_s_a	uganda
ukraine	united_arab_emirates	united_kingdom	united_states
united_states_of_america	uruguay	utah	utica
uzbekistan	vail	valentine	vancouver
vegas	venezuela	venice	vermont
vernal	victoria	vienna	vietnam
virgin_islands	virginia	virginia_beach	vladivostok
wales	warsaw	washington	washington_d_c
washington_state	watertown	wendover	west_palm_beach
west_virginia	wheeling	whidbey_island	wichita
wichita_falls	williamsburg	williamsport	williston
wilmington	winnemucca	winnipeg	winston_salem
wisconsin	worcester	world	wyoming
yellowstone	yukon	zaire	zambia
zimbabwe	zurich		

Content Type 3

afternoon	afternoon+s	antarctic	april
arctic	atlantic	august	b_w_i
caribbean	cloudy	cold	current
d_f_w	december	dulles	evening
evening+s	february	fog	foggy
freezing	friday	friday+s	gatwick

Content Type 3 Cont...

hail	heat_wave	heathrow	hot
humid	humidity	hurricane	hurricanes
indian	j_f_k	january	july
june	la_guardia	logan	march
may	mediterranean	midway	monday
monday+s	morning	morning+s	national
november	o+hare	ocean	october
orly	overcast	pacific	present
pressure	rain	raining	rainy
rise	rising	saturday	saturday+s
sea	september	set	setting
shining	shower	showers	sleet
smog	snow	snowfall	snowing
snowstorm	stapleton	storm	storms
stormy	sunday	sunday+s	sunny
sunrise	sunset	temperature	thunder
thunderstorms	thursday	thursday+s	today
today+s	tomorrow	tomorrow+s	tornado
tornados	tuesday	tuesday+s	warm
wednesday	wednesday+s	wind	windy
yesterday	yesterday+s		

Content Type 2 Cont...

all	anticipated	any	can
could	did	do_you	does
each	east	eastern	eight
eighteen	eighteenth	eighth	eighty
eleven	eleventh	every	expected
fifteen	fifteenth	fifth	fifty
first	five	forty	four
fourteen	fourteenth	fourth	likely
must	nine	nineteen	nineteenth
ninety	ninth	north	northeast
northern	northwest	one	predicted
second	seven	seventeen	seventeenth
seventh	seventy	should	six
sixteen	sixteenth	sixth	sixty
south	southeast	southern	southwest
ten	tenth	thank_you	thanks
third	thirteen	thirteenth	thirtieth

Content Type 2

thirty	three	twelfth	twelve
twentieth	twenty	two	west
western			

Content Type 1

All words not listed in Content Types 2,3 and 4.

Examples:	a	about	besides
	for	in	is
	I_would	on	like
	that	the	what_is
	weather		

Bibliography

[1] Z. Bergen and W. Ward. A senone based confidence measure for speech recognition. In *Proc. European Conference on Speech Communication and Technology*, pages 819–822, Rhodes, Greece, 1997.

[2] J. Chang. *Near-Miss Modeling: A Segment-Based Approach to Speech Recognition.* PhD thesis, Massachusetts Institute of Technology, 1998.

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

[4] G. Chung and S. Seneff. Improvements in speech understanding accuracy through the integration of hierarchical linguistic, prosodic, and phonological constraints in the JUPITER domain. In *Proc. International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[5] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1973.

[6] L. Gillick, Y. Ito, and J. Young. A probabilistic approach to confidence estimation and evaluation. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 879–882, Munich, Germany, 1997.

[7] J. Glass. A probabilistic framework for feature-based speech recognition. In *Proc. International Conference on Spoken Language Processing*, Philadelphia, PA, 1996.

[8] J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue. Multi-lingual spoken language understanding in the MIT VOYAGER system. *Speech Communication*, 17:1–18, 1995.

[9] J. Glass and T. Hazen. Telephone-based conversational speech recognition in the JUPITER domain. In *Proc. International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[10] J. Glass, T. Hazen, and L. Hetherington. Real-time telephone-based speech recognition in the JUPITER domain. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[11] J. Glass, J. Polifroni, S. Seneff, and V. Zue. Multilingual speech-understanding for human-computer interaction. In *CRIM-FORWISS Workshop*, Montreal, Canada, 1996.

[12] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J Polifroni, S. Seneff, and V. Zue. Galaxy: A human-language interface to on-line travel information. In *Proc. International Conference on Spoken Language Processing*, pages 707–710, Yokohama, Japan, 1994.

[13] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In *Proc. International Conference on Spoken Language Processing*, Philadelphia, PA, 1996.

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

[15] E. Lleida and R. Rose. Efficient decoding and training procedures for utterance verification in continuous speech recognition. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, 1996.

[16] C. Neti, S. Roukos, and E. Eide. Word-based confidence measures as a guide for stack search in speech recognition. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 883–886, Munich, Germany, 1997.

[17] C. Pao, P. Schimdt, and J. Glass. Confidence scoring for speech understanding systems. In *Proc. International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[18] H. Qiu. Confidence measures for speech recognition systems. Master's thesis, Carnegie Mellon University, 1996.

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

[20] R. Rose and D. Paul. A hidden markov model based keyword recognition system. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[21] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech recognition. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 875–878, Munich, Germany, 1997.

[22] S. Seneff. TINA: A natural language system for spoken language applications. *Computational Linguistics*, vol.18(no.1):pp.61–86, March 1992.

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

[24] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 887–890, Munich, Germany, 1997.

[25] J. Wilpon, L. Rabiner, C. Lee, and E. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.38(no.11):pp.1870–8, November 1990.

[26] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Recent progress on the SUMMIT system. In *Proc. DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, 1990.