

January 01, 2009

Study of algorithms to combine multiple automatic speech recognition (ASR) system outputs

Harish Kashyap Krishnamurthy
Northeastern University

Recommended Citation

Krishnamurthy, Harish Kashyap, "Study of algorithms to combine multiple automatic speech recognition (ASR) system outputs" (2009). *Electrical and Computer Engineering Master's Theses*. Paper 24. <http://hdl.handle.net/2047/d10019273>

This work is available open access, hosted by Northeastern University.

**STUDY OF ALGORITHMS TO COMBINE MULTIPLE AUTOMATIC SPEECH
RECOGNITION (ASR) SYSTEM OUTPUTS**

A Thesis Presented

by

Harish Kashyap Krishnamurthy

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Master of Science

in

Electrical and Computer Engineering

in the field of

Communication & Digital Signal Processing

Northeastern University
Boston, Massachusetts

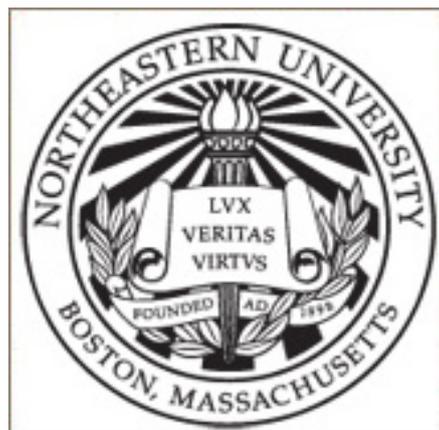
April 2009

HARISH KASHYAP KRISHNAMURTHY

STUDY OF ALGORITHMS TO COMBINE
MULTIPLE AUTOMATIC SPEECH
RECOGNITION (ASR) SYSTEM OUTPUTS

STUDY OF ALGORITHMS TO COMBINE
MULTIPLE AUTOMATIC SPEECH
RECOGNITION (ASR) SYSTEM
OUTPUTS

HARISH KASHYAP KRISHNAMURTHY



Master of Science

Communication and Digital Signal Processing
Electrical and Computer Engineering
Northeastern University

April 2009

Harish Kashyap Krishnamurthy: *Study of Algorithms to Combine Multiple Automatic Speech Recognition (ASR) System outputs*,
Master of Science, © April 2009

{prayers to Sri Hari Vayu Gurugalu}

Dedicated to the loving memory of my late grandparents,
Srinivasamurthy and Yamuna Bai.

ABSTRACT

Automatic Speech Recognition systems (ASRs) recognize word sequences by employing algorithms such as Hidden Markov Models. Given the same speech to recognize, the different ASRs may output very similar results but with errors such as insertion, substitution or deletion of incorrect words. Since different ASRs may be based on different algorithms, it is likely that error segments across ASRs are uncorrelated. Therefore it may be possible to improve the speech recognition accuracy by exploiting multiple hypotheses testing using a combination of ASRs. System Combination is a technique that combines the outputs of two or more ASRs to estimate the most likely hypothesis among conflicting word pairs or differing hypotheses for the same part of utterance. In this thesis, a conventional voting scheme called Recognized Output Voting Error Reduction (ROVER) is studied. A weighted voting scheme based on Bayesian theory known as Bayesian Combination (BAYCOM) is implemented. BAYCOM is derived from first principles of Bayesian theory. ROVER and BAYCOM use probabilities at the system level, such as performance of the ASR, to identify the most likely hypothesis. These algorithms arrive at the most likely word sequences by considering only a few parameters at the system level. The motivation is to develop newer System Combination algorithms that model the most likely word sequence hypothesis based on parameters that are not only related to the corresponding ASR but the word sequences themselves. Parameters, such as probabilities with respect to hypothesis and ASRs are termed word level probabilities and system level probabilities, respectively, in the thesis. Confusion Matrix Combination is a decision model based on parameters at word level. Confusion matrix consisting of probabilities with respect to word sequences are estimated during training. The system combination algorithms are initially trained with known speech transcripts followed by validation on a different set of transcripts. The word sequences are obtained by processing speech from Arabic news broadcasts. It is found that Confusion Matrix Combination performs better than system level BAYCOM and ROVER over the training sets. ROVER still proves to be a simple and powerful system combination technique and provides best improvements over the validation set.

*First I shall do some experiments before I proceed farther,
because my intention is to cite experience first and
then with reasoning show why such experience is bound to
operate in such a way. And this is the true rule by which those who
speculate about the effects of nature must proceed*

— Leonardo Da Vinci [4]

ACKNOWLEDGMENTS

Foremost, I would like to thank my supervisor Prof. John Makhoul¹, without whom this wonderful research opportunity with BBN would have been impossible. John Makhoul's stature is such that, not just myself, but many people in BBN and speech community around the world have always looked upto as the ideal researcher.

Hearty thanks to Spyros Matsoukas², whom I worked with closely throughout my Masters. Spyros was not only a lighthouse to my research but also helped towards their implementation. I must say that I learnt true, efficient and professional programming from Spyros. He was always pleasant, helpful and always patiently acquiescing to my flaws.

Many thanks to Prof. Jennifer Dy³, for teaching the pattern recognition course. Prof. Jennifer was encouraging and interactions with her proved very useful.

Many thanks to Prof. Hanoch Lev-Ari⁴, who, I must say was easily approachable, popular amongst students and was a beacon for all guidance.

I can never forget Joan Pratt, CDSP research lab mates and friends at Northeastern. I thank Prof. Elias Manolakos for having referred me to various professors for research opportunities.

Lastly and most importantly, I wish to thank my family, Sheela, Krishnamurthy, Deepika and Ajit Nimbalkar for their emotional support. I thank all my friends especially Raghu, Rajeev and Ramanujam who have been like my extended family. Special thanks to my undergraduate advisor and friend, Dr. Bansilal from whom I have drawn inspiration for research.

¹ Chief Scientist, BBN Technologies

² BBN Technologies

³ Associate Professor, Northeastern University

⁴ Dean of ECE, Northeastern University

CONTENTS

1	Introduction to Speech Recognition and System Combination	1
1.1	Architecture of ASR	1
1.1.1	Identifying Word Sequences	2
1.1.2	Acoustic Modeling	3
1.1.3	Language Modeling	4
1.1.4	Evaluation of the Speech Recognition System	4
1.2	Confidence Estimation	5
1.2.1	Posterior Probability decoding and confidence scores	5
1.2.2	Large Vocabulary Speech Recognition Algorithms	6
1.2.3	N-Best Scoring	7
1.3	System Combination	7
1.3.1	Introduction to System Combination	7
1.4	The framework of a typical system combination algorithm	7
1.4.1	System Combination: A literature survey	8
1.5	Thesis Outline	10
2	Experimental Setup	13
2.1	Introduction	13
2.2	Design of Experiments	13
2.2.1	System Combination Experiment Layout	13
2.2.2	Benchmark STT Systems	13
3	ROVER - Recognizer Output Voting Error Reduction	15
3.1	Introduction	15
3.2	Dynamic Programming Alignment	16
3.3	ROVER Scoring Mechanism	17
3.3.1	Frequency of Occurrence	18
3.3.2	Frequency of Occurrence and Average Word Confidence	18
3.3.3	Maximum Confidence Score	19
3.4	Performance of ROVER	19
3.4.1	The Benchmark STT Systems	19
3.5	Features of ROVER	20
4	Bayesian Combination - BAYCOM	21
4.1	Introduction	21
4.2	Bayesian Decision Theoretic Model	21

4.2.1	BAYCOM Training	23
4.2.2	BAYCOM Validation	23
4.2.3	Smoothing Methods	24
4.3	BAYCOM Results	24
4.3.1	The Benchmark STT Systems	24
4.4	Tuning the Bin Resolution	24
4.5	Tuning Null Confidence	25
4.6	Features of BAYCOM	26
5	Confusion Matrix Combination	29
5.1	Introduction	29
5.2	Computing the Confusion Matrix	29
5.2.1	Confusion Matrix Formulation	30
5.2.2	Validation of Confusion Matrix combination	30
5.2.3	Validation Issues in Confusion Matrix Combination	32
5.3	Confusion Matrix Combination Results	32
5.4	Features of CMC	34
6	Results	35
6.1	Analysis of Results	35
6.1.1	System Combination Experiment Combining 2 MPFE and 1 MMI System	35
6.1.2	BAYCOM Experiment Combining 2 MPFE and 1 MMI System	35
6.2	Smoothing Methods for System Combination Algorithms	39
6.3	Backing off	39
6.4	Mean of Probability of Confidence Score Bins	39
7	Conclusions	41
	BIBLIOGRAPHY	42

LIST OF FIGURES

Figure 1	ASR	3
Figure 2	A typical Hidden Markov Model	4
Figure 3	syscomb	8
Figure 4	rover	15
Figure 5	wtn	16
Figure 6	wtn2	17
Figure 7	wtn-3	17
Figure 8	WTN	18
Figure 9	Building the Confusion Matrices	31

LIST OF TABLES

Table 1	Training Hours for each ASR to be combined	14
Table 2	Training on at6	14
Table 3	Validation on ad6	14
Table 4	Training on at6	19
Table 5	Validation on ad6	19
Table 6	Training on at6	24
Table 7	Varying Nullconf	26
Table 8	Varying bin resolution between 0 and 0.3	26
Table 9	Training on at6	26
Table 10	Training on at6	32
Table 11	Validation on ad6	33
Table 12	Varying bin resolution between 0 and 5	33
Table 13	Varying Nullconf between 0 and 1	33
Table 14	Training on at6	34
Table 15	Validation on ad6	34
Table 16	Rover on MPFE and MMI	35
Table 17	Optimum values of a and c	36
Table 18	BAYCOM on MPFE and MMI	36
Table 19	Varying Nullconf between 0 and 1	37
Table 20	Varying bin resolution between 0 and 1	37
Table 21	Training on at6	38
Table 22	Validation on ad6	38

Table 23	Training on at6	39
Table 24	Training on at6	40
Table 25	Validation on ad6	40

ACRONYMS

ASR Automatic Speech Recognition

WER Word Error Rate

HMM Hidden Markov Model

ROVER Recognizer Output Voting Error Reduction

BAYCOM Bayesian Combination

CMC Confusion Matrix Combination

MMI Maximum Mutual Information

ML Maximum Likelihood

INTRODUCTION TO SPEECH RECOGNITION AND SYSTEM COMBINATION

Speech signals consist of a sequence of sounds produced by the speaker. Sounds and the transitions between them serve as a symbolic representation of information, whose arrangement is governed by the rules of language [19]. Speech recognition, at the simplest level, is characterized by the words or phrases you can say to a given application and how that application interprets them. The abundance of spoken language communication in our daily interaction accounts for the importance of speech applications in human-machine interaction. In this regard, automatic speech recognition (ASR) has gained a lot of attention in the research community since 1960s. A separate activity initiated in the 1960s, dealt with the processing of speech signals for data compression or recognition purposes in which a computer recognizes the words spoken by someone [16]. Automatic speech recognition is processing a stored speech waveform and expressing in text format, the sequence of words that were spoken. The challenges to build a robust speech recognition system include the form of the language spoken, the surrounding environment, the communicating medium and/or the application of the recognition system [12]. Speech Recognition research started with attempts to decode isolated words from a small vocabulary and as time progressed focus shifted towards working on large vocabulary and continuous speech tasks [17]. Statistical modeling techniques trained from hundreds of hours of speech have provided most speech recognition advancements. In the past few decades dramatic improvements have made high performance algorithms and systems that implement them available [21].

1.1 ARCHITECTURE OF ASR

A typical Automatic Speech Recognition System (ASR) embeds information about the speech signal by extracting acoustic features from it. These are called acoustic observations.

Most computer systems for speech recognition include the following components [18]:

- Speech Capturing device

- Digital Signal Processing Module
- Preprocessed Signal Storage
- Hidden Markov Models
- A pattern matching algorithm

ASR: Speech Capturing device, which usually consists of a microphone and associated analog to digital converter that converts the speech waveform into a digital signal. A Digital Signal Processing (DSP) module performs endpoint detection to separate speech from noise and converts the raw waveform into a frequency domain representation, and performs further windowing, scaling and filtering [18]. Goal is to enhance and retain only the necessary components of spectral representation that are useful for recognition purposes. The preprocessed speech is buffered before running the algorithm. Modern speech recognition systems use HMMs to recognize the word sequences. The problem of recognition is to search for the word sequence that most likely represents the acoustic observation sequence using the knowledge from the acoustic and language models. A block diagram of an ASR is shown in [Figure 1](#)

The pattern matching algorithms that form the core of speech recognition has evolved over time. Dynamic time warping compares the preprocessed speech waveform directly against a reference template. Initially experiments were designed mostly by applying dynamic time warping, hidden markov models and Artificial Neural Networks.

1.1.1 Identifying Word Sequences

Given the acoustic evidence (observation sequence) O , the problem of speech recognition is to find the most likely word sequence W^* among competing set of word sequences W ,

$$W^* = \arg \max_W p(W|O) \quad (1.1)$$

The probability of word sequence given the observation sequence O can be written using the Bayes theorem as,

$$p(W|O) = \arg \max_w \frac{p(W) \times p(O|W)}{p(O)} \quad (1.2)$$

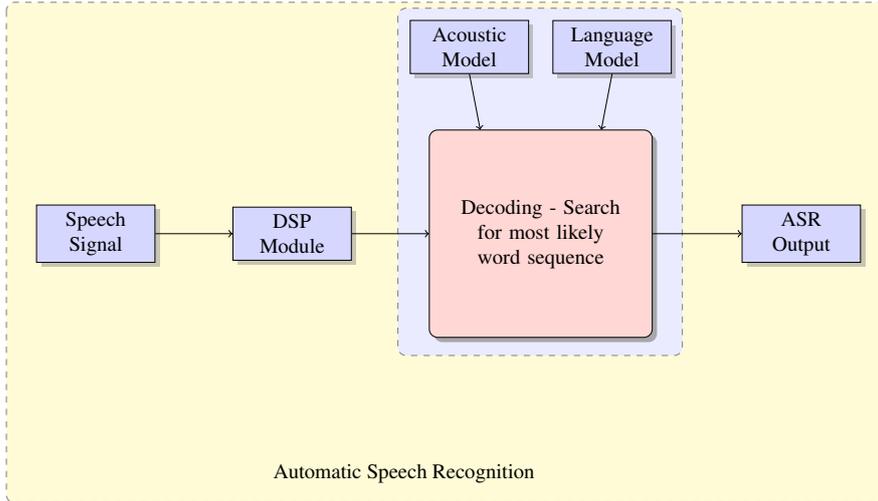


Figure 1: Automatic Speech Recognition

Since $p(O)$ is constant w.r.t given word sequence W ,

$$W^* = \arg \max_w p(W) \times p(O|W) \quad (1.3)$$

Computing $p(O|W)$ is referred to as "acoustic modeling" and computing $p(W)$ is called "language modeling", and searching for the most likely sequence that maximizes the likelihood of the observation sequence is referred to as "decoding".

1.1.2 Acoustic Modeling

The acoustic model generates the probability $p(O|W)$. For Large Vocabulary Continuous Speech Recognition (LVCSR), it is hard to estimate a statistical model for every word in the large vocabulary. The models are represented by triphones (phonemes with a particular left and right neighbor or context). The triphones are represented using a 5 state Hidden Markov Model (HMM) as shown in [Figure 2](#). The output distributions for the HMMs are represented using mixtures of Gaussians.

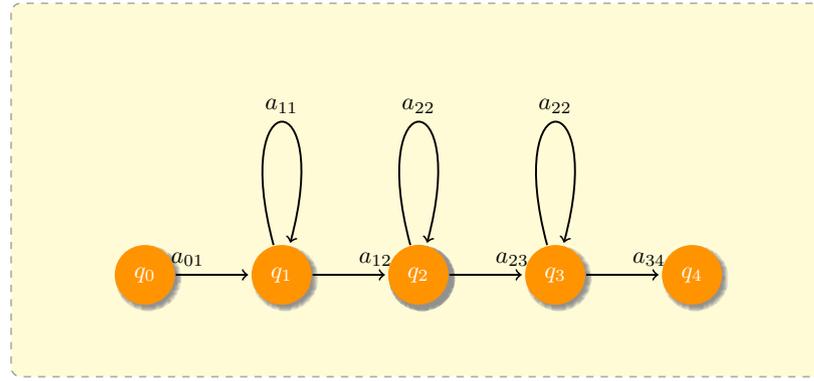


Figure 2: A typical Hidden Markov Model

1.1.3 Language Modeling

The language model models the probability of a sequence of words. The probability of a word W_i is based on the n -gram probabilities of the previous $n - 1$ words.

$$p(W_i|W_1, W_2, \dots, W_{i-1}) \approx p(W_i|W_{i-n+1}, W_{i-n+2}, \dots, W_{i-1}) \quad (1.4)$$

Eq. 1.4 represents the forward n -gram probability.

1.1.4 Evaluation of the Speech Recognition System

To evaluate the performance of any speech recognizer, the speech community employs Word Error Rate (WER). The hypothesized transcript is aligned to the reference transcript on words through the method of dynamic programming. Three sets of errors are computed:

- S: Substitution Error, a word is substituted by ASR to a different word.
- I: Insertion Error, a word present in the hypothesis, but absent in the reference.
- D: Deletion Error, a word present in the reference, but missing from the hypothesis.
- R: Number of words in the reference.

$$\text{WER} = \frac{(S + I + D) \times 100}{R} \quad (1.5)$$

1.2 CONFIDENCE ESTIMATION

Automatic Speech Recognition has achieved substantial success mainly due to two prevalent techniques, hidden markov models of speech signals and dynamic programming search for large scale vocabularies [14]. However, ASR as applied to real world data still encounters difficulties. System performance can degrade due to either less available training data, noise or speaker variations and so on. To improve performance of ASRs in real world data has been an interesting and challenging research topic. Most speech recognizers will have errors during recognition of validation data. ASR outputs also have a variety of errors. Hence, it is extremely important to be able to make important and reliable judgements based on the error-prone results [14]. The ASR systems hence, automatically assess the reliability or probability of correctness with which the decisions are made. These probabilities output, called confidence measures (CM) are computed for every recognized word. CM indicate as to how likely the word was correctly recognized by the ASR. Confidence Estimation refers to annotating values in the range 0 to 1 that indicates the confidence of the ASR with respect to the word sequence output. An approach based on interpretation of the confidence as the probability that the corresponding recognized word is correct is suggested in [10]. It makes use of generalized linear models that combine various predictor scores to arrive at confidence estimates. A probabilistic framework to define and evaluate confidence measures for word recognition was suggested in [23]. Some other literature that explain different methods of confidence estimation can be found in [25], [24], [5].

1.2.1 *Posterior Probability decoding and confidence scores*

In the thesis, estimation of word posterior probabilities based on word lattices for a large vocabulary speech recognition system proposed in [8] is used. The problem of the robust estimation of confidence scores from word posteriors is examined in the paper and a method based on decision trees is suggested. Estimating the confidence of a hypothesized word directly as its posterior probability, given all acoustic observations of the utterance is proposed in this work. These probabilities are computed on word graphs using a forward-backward algorithm. Estimation of posterior probabilities on n-best lists instead of word graphs and compare both algorithms in detail. The posterior probabilities

computed on word graphs was claimed to outperform all other confidence measures.

The word lattices produced by the Viterbi decoder were used to generate confusion networks. These confusion networks provide a compact representation of the most likely word hypotheses and their associated word posterior probabilities [7]. These confusion networks were used in a number of post-processing steps. [7] claims that the 1-best sentence hypotheses extracted directly from the networks are significantly more accurate than the baseline decoding results. The posterior probability estimates are used as the basis for the estimation of word-level confidence scores. A system combination technique that uses these confidence scores and the confusion networks is proposed in this work. The confusion networks generated are used for decoding. The confusion network consists of each hypothesis word tagged along with posterior probability. The word with the maximum posterior probability will most likely output the best hypothesis with lowest word error rate for the set. A confidence score is certainty measure of a recognizer in its decision. These confidence scores are useful indicators that can be further processed. Bayesian Combination(BAYCOM) and Recognizer Output Voting Error Reduction (ROVER) are examples of Word Error Rate (WER) improvement algorithms that use confidence scores output from different systems [9, 20]. They are useful in the sense of decision making such as selecting the word with highest confidence score or rejecting a word with confidence scores below a threshold. The word posterior probabilities of the words in confusion network can be used directly as confidence scores in cases where WER is low and in cases of higher WER, Normalized Cross Entropy (NCE) measures are preferred.

1.2.2 *Large Vocabulary Speech Recognition Algorithms*

Early attempts towards speech recognition was by applying expert knowledge techniques. These algorithms were not adequate for capturing the complexities of continuous speech [17]. Later research focussed on applying artificial intelligence techniques followed by statistical modeling to improve speech recognition. Statistical techniques along with artificial intelligence algorithms helps improve performance. The algorithms studied in the thesis comprise of large scale vocabulary and is a classical demonstration of applying statistical algorithms to different artificial intelligence based ASRs.

1.2.3 *N-Best Scoring*

Scoring of N best sentence hypothesis was introduced by BBN as a strategy for integration of speech and natural language [6]. Among a list of N candidate sentences, a natural language system can process all the competing hypothesis until it chooses the one that satisfies the syntactic and semantic constraints.

1.3 SYSTEM COMBINATION

1.3.1 *Introduction to System Combination*

Combining different systems was proposed in 1991, [1], by combining a BU system based on stochastic segment models (SSM) and a BBN system based on Hidden Markov Models . It was a general formalism for integrating two or more speech recognition technologies developed at different research sites using different recognition strategies. In this formalism, one system used the N-best search strategy to generate a list of candidate sentences that were rescored by other systems and combined to optimize performance. In contrast to the HMM, the SSM scores a phoneme as a whole entity, allowing a more detailed acoustic representation. If the errors made by the two systems differ, then combining the two sets of scores can yield an improvement in overall performance. The basic approach involved

1. Computing the N-best sentence hypotheses with one system
2. Rescoring this list of hypotheses with a second system
3. Combining the scores and re-ranking the N-Best hypothesis to improve overall performance

1.4 THE FRAMEWORK OF A TYPICAL SYSTEM COMBINATION ALGORITHM

The general layout of system combination algorithms used in the thesis can be explained with the help of [Figure 3](#). The experiments largely consist of:

- Training phase
- Validation phase

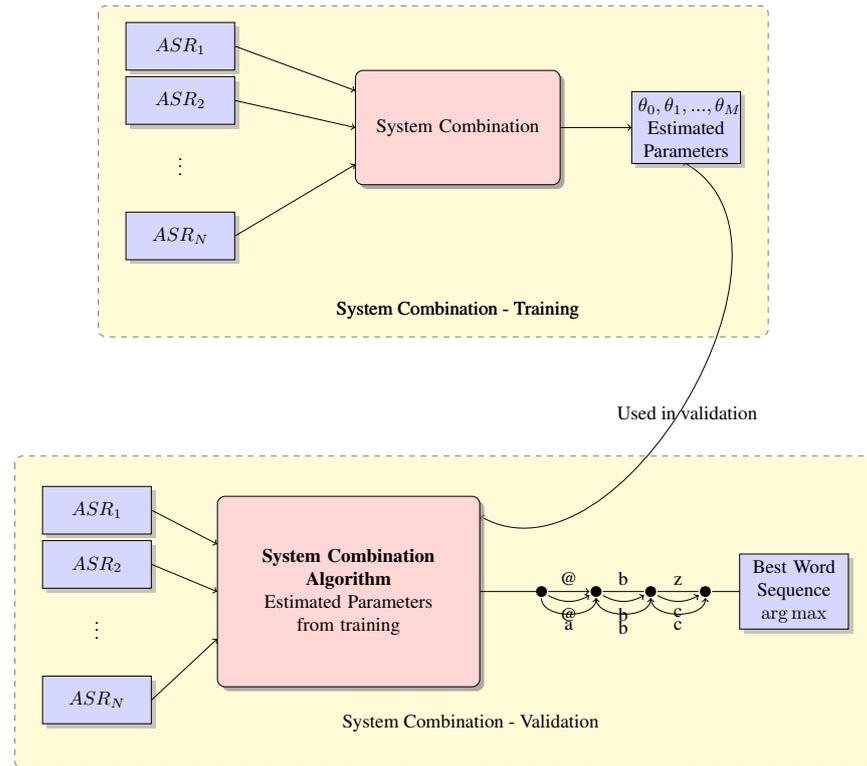


Figure 3: System Combination Algorithm

Training phase consists of estimating parameters that are used during validation. These parameters are usually word probabilities, probability distributions or can simply be optimized variables that output the best word sequences. M parameters of the vector θ are estimated during training phase. These parameters are used in the validation phase. The words output by each ASR along with their word confidences are substituted by values computed by the system combination algorithm. A word transition network aligns the competing words output from the combined ASRs by the method explained in [Chapter 3](#). The words having highest annotated confidence scores among the competing words in the word transition network are chosen as the best words. The evolutions and development of the system combination algorithms are explained in the next section.

1.4.1 System Combination: A literature survey

A system combination method was developed at National Institute of Standards and Technology (NIST) to produce a composite

Automatic Speech Recognition (ASR) system output when the outputs of multiple ASR systems were available, and for which, in many cases, the composite ASR output had a comparatively lower error rate. It was referred to as A NIST Recognizer Output Voting Error Reduction (ROVER) system. It is implemented by employing a "voting" scheme to reconcile differences in ASR system outputs. As additional knowledge sources are added to an ASR system, (e.g., acoustic and language models), error rates get reduced further. The outputs of multiple of ASR systems are combined into a single, minimal cost word transition network (WTN) via iterative applications of dynamic programming alignments. The resulting network is searched by a "voting" process that selects an output sequence with the lowest score [9]. Another variation of ROVER was suggested in [13].

Also combining different systems has been proved to be useful for improving gain in acoustic models [11]. It was proved that better results are obtained when the adaptation procedure for acoustic models exploits a supervision generated by a system different than the one under adaptation. Cross-system adaptation was investigated by using supervisions generated by several systems built varying the phoneme set and the acoustic front-end. An adaptation procedure that makes use of multiple supervisions of the audio data for adapting the acoustic models within the MLLR framework was proposed in [11].

An integrated approach where the search of a primary system is driven by the outputs of a secondary one is proposed in [15]. This method drives the primary system search by using the one-best hypotheses and the word posteriors gathered from the secondary system. A study of the interactions between "driven decoding" and cross-adaptations is also presented.

A computationally efficient method for using multiple speech recognizers in a multi-pass framework to improve the rejection performance of an automatic speech recognition system is proposed in [22]. A set of criteria is proposed that determine at run time when rescoring using a second pass is expected to improve the rejection performance. The second pass result is used along with a set of features derived from the first pass and a combined confidence score is computed. The combined system claims significant improvements over a two-pass system at little more computational cost than comparable one-pass and two-pass systems.[22]

A method for predicting acoustic feature variability by analyzing the consensus and relative entropy of phoneme posterior probability distributions obtained with different acoustic mod-

els having the same type of observations is proposed in [2]. Variability prediction is used for diagnosis of automatic speech recognition (ASR) systems. When errors are likely to occur, different feature sets are combined for improving recognition results.

Bayesian Combination, BAYCOM, a Bayesian decision-theoretic approach to model system combination proposed in [20] is applied to recognition of sentence level hypothesis. BAYCOM is an approach based on bayesian theory that requires computation of parameters at system level such as Word Error Rate (WER). The paper argues that mostly the previous approaches were ad-hoc and not based on any known pattern recognition technique. [20] claims that BAYCOM gives significant improvements over previous combination methods.

1.5 THESIS OUTLINE

The thesis has been organized as follows. The system combination algorithms are applied to a set of benchmark ASR systems and their performance are evaluated. The ASR outputs of word sequences that are to be combined may differ in, time at which they are output, as well as the length of the word sequences. Hence combining the various ASR outputs are non-trivial. [Chapter 2](#) explains how the different ASR outputs are combined as well as the type of the ASRs, which is necessary for the application of the system combination algorithms. Amongst the existing system combination algorithms, ROVER, the most prevalent and popular system combination method, is explained in [Chapter 3](#). ROVER is used as a benchmark for comparing different system combination algorithms. It is however, based on training a linear model for few parameters. BAYCOM at the word level is deduced from the first principles of BAYCOM at the sentence level in [Chapter 4](#). Training BAYCOM at the word level requires computation of parameters related to the system such as the word error rate of the individual ASRs combined. While BAYCOM does provide improvements in the Word Error Rate over all the individual systems combined, motivation is to explore algorithms where parameters related to the word level are used rather than those at the system level. Hence, analysis of ROVER and BAYCOM motivates us to explore techniques where parameters used are not only related to the ASR systems that output the word sequences, but the specific word sequences themselves. A novel system combination method, Confusion Matrix Combination (CMC) that uses confusion matrices to store word level

parameters is proposed in [Chapter 5](#). Lastly, we compare and analyze the performance of these algorithms over arabic news broadcast in [Chapter 6](#). [Chapter 7](#) gives the outcome of the study of the system combination algorithms as well as directions for future work.

EXPERIMENTAL SETUP

2.1 INTRODUCTION

This chapter provides details about the basic setup of experiments cited in the thesis. This is useful to analyze performance of each algorithm against the same input data. This section is devoted to not only provide details on the design of experiments but also the methodology involved in analyzing the results.

2.2 DESIGN OF EXPERIMENTS

2.2.1 *System Combination Experiment Layout*

Initially, ASR systems that are to be combined are selected and confidence estimation experiments are run to annotate word confidences for each of the words output by the ASRs. [Table 1](#) shows an example of 3 models selected and the corresponding number of training hours. The experiments conducted essentially involve execution of the speech recognition, confidence estimation or system combination algorithms in a parallel computing environment. Since the number of training hours are usually large, the algorithms are usually parallelized and run on a cluster. The experiment numbers, provided at each experiment in the thesis, serve as job-ids for the job submission queue and are referred to the experiments cited in the thesis. 2 of the models, Maximum Mutual Information (MMI) vowelized system(18741) and Maximum Likelihood (ML) vowelized system(18745) are trained by 150 Hours of broadcast news in arabic language. The third model, is also an MMI vowelized system(18746), however trained differently, with unsupervised training by 900 hours of broadcast news in arabic language. Hence, there are 3 ASR system outputs trained differently, that are combined.

2.2.2 *Benchmark STT Systems*

Training sets, at6, as shown in [Table 2](#) are used to train the system combination algorithms. The training and validation sets are benchmarks to compare and analyze each system combina-

EXPT. NO	MODEL TYPE	TRAINING IN HOURS
18741	MMI baseline vowelized system.	150
18745	ML Vowelized System.	150
18746	MMI vowelized system with unsupervised training	900

Table 1: Training Hours for each ASR to be combined

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	28.3
21993tw	MPFE BBN	26.2
Limsi	MMI	27.4

Table 2: Training on at6

tion algorithm that are explained from the [Chapter 3](#) onwards. With this setup as the benchmark we shall see the performance of the popular system combination algorithm ROVER in the next chapter.

Validation sets for testing the training system combination algorithms is done on ad6 sets which are 6 hours long. The 3 systems combined are 2 MPFE from BBN and 1 MMI system from Limsi-1 as shown in [Table 3](#).

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	24.6
21993dw	MPFE BBN	24.6
Limsi	MMI	28.8

Table 3: Validation on ad6

ROVER - RECOGNIZER OUTPUT VOTING ERROR REDUCTION

3.1 INTRODUCTION

ROVER is a system developed at National Institute of Standards and Technology (NIST) to combine multiple Automatic Speech Recognition (ASR) outputs. Outputs of ASR systems are combined into a composite, minimal cost word transition network (WTN). The network thus obtained is searched by a voting process that selects an output sequence with the lowest score. The voting or rescoring process reconciles differences in ASR system outputs. This system is referred to as the NIST Recognizer Output Voting Error Reduction (ROVER) system. As additional knowledge sources are added to an ASR system, (e.g., acoustic and language models), error rates are typically reduced. The ROVER system is implemented in two modules as shown in Figure 4. First, the system outputs from two or more ASR systems are combined into a single word transition network. The network is created using a modification of the dynamic programming alignment protocol traditionally used by NIST to evaluate ASR technology. Once the network is generated, the second module evaluates each branching point using a voting scheme, which selects the best scoring word having the highest number of votes for the new transcription [9].

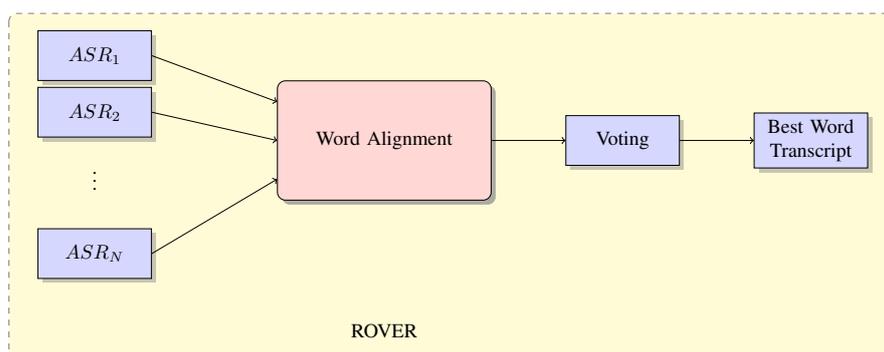


Figure 4: ROVER system architecture

3.2 DYNAMIC PROGRAMMING ALIGNMENT

The first stage in the ROVER system is to align the output of two or more hypothesis transcripts from ASR systems in order to generate a single, composite WTN. The second stage in the ROVER system scores the composite WTN, using any of several voting procedures. To optimally align more than two WTNs using DP would require a hyper-dimensional search, where each dimension is an input sequence. Since such an algorithm would be difficult to implement, an approximate solution can be found using two-dimensional DP alignment process. SCLITE is a dynamic programming engine that determines minimal cost alignment between two networks. From each ASR, a WTN is formed by SCLITE. It finds WTN that involves minimal cost alignment and no-cost transition word arcs. Each of the systems is a linear sequence of words. First a base WTN, usually with best performance (lowest WER) is selected and other WTNs are combined in an order of increasing WER. DP alignment protocol is used to align the first two WTNs and later on, additional WTNs are added on iteratively. Figure 5 shows outputs of 3 ASRs to be combined by dynamic programming.

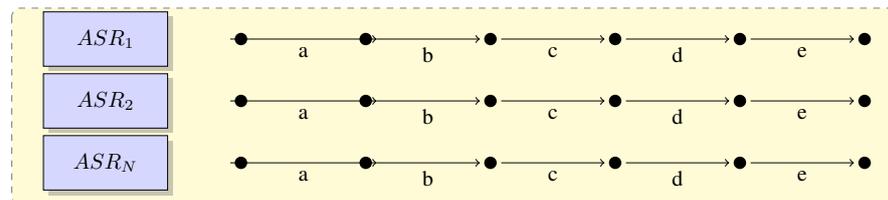


Figure 5: WTNs before alignment

The first WTN, WTN_{Base} is designated as the base WTN from which the composite WTN is developed. The second WTN is aligned to the base WTN using the DP alignment protocol and the base WTN is augmented with word transition arcs from the second WTN. The alignment yields a sequence of correspondence sets between WTN_{Base} and WTN-2. Figure 6 shows the 5 correspondence sets generated by the alignment between WTN_{Base} and WTN-2.

The composite WTN can be considered as a linear combination of word-links with each word link having contesting words output from different ASRs combined. Using the correspondence sets identified by the alignment process, a new, combined WTN, WTN_{Base} , illustrated in Figure 7, is made by

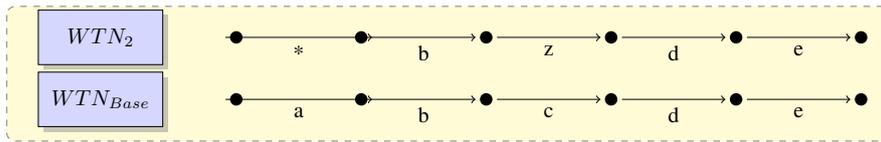


Figure 6: WTN-2 is aligned with WTN_{Base} by the DP Alignment

copying word transition arcs from WTN_2 into WTN_{Base} . When copying arcs into WTN_{Base} , the four correspondence set categories are used to determine how each arc copy is made [9]. For a correspondence set marked as:

1. Correct : a copy of the word transition arc from WTN_2 is added to the corresponding word in WTN_{Base} .
2. Substitution: a copy of the word transition arc from WTN_2 is added to WTN_{Base} .
3. Deletion: a no-cost, NULL word transition arc is added to WTN_{Base} .
4. Insertion: a sub-WTN is created ,and inserted between the adjacent nodes in WTN_{Base} to record the fact that the WTN_2 network supplied a word at this location. The sub-WTN is built by making a two-node WTN, that has a copy of the word transition arc from WTN_2 , and P NULL transition arcs where P is the number of WTNs already merged into WTN_{Base} .

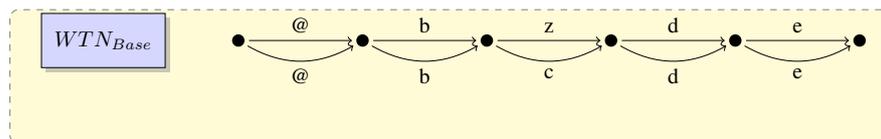


Figure 7: The final composite WTN.

Now that a new base WTN has been made, the process is repeated again to merge WTN_3 into WTN_{Base} . [Figure 8](#) shows the final base WTN which is passed to the scoring module to select the best scoring word sequence.

3.3 ROVER SCORING MECHANISM

The ASRs combined necessarily have to supply a word confidence ranging between 0 and 1 for each word output from the

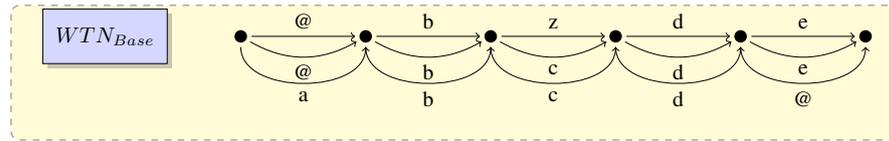


Figure 8: The final composite WTN.

ASRs. These word confidences can be considered as the amount of confidence of each ASR pertaining to each word output. For this purpose, Confidence estimation is performed on each training set before combining them. The voting scheme is controlled by parameters α and null confidence N_c that weigh

- Frequency of occurrence and
- Average Confidence score.

These two parameters, tuned for a particular training set, are later used for validations. Alignment of words in a Word Transition Network using SCLITE.

The scoring mechanism of ROVER can be performed in 3 ways by prioritizing:

- Frequency of Occurrence
- Frequency of Occurrence and average word confidence
- Frequency of Occurrence and Maximum confidence

$$S(w_i) = \alpha \times F(w_i) + (1 - \alpha) \times C(w_i) \quad (3.1)$$

where $F(w_i)$ is the frequency of occurrence and $C(w_i)$ is the word confidence.

3.3.1 Frequency of Occurrence

Setting the value of α to 1.0 in Equation 3.1 nullifies confidence scores in voting. The major disadvantage of this method of scoring is that the composite WTN can contain deletions or missing words.

3.3.2 Frequency of Occurrence and Average Word Confidence

Missing words are substituted by a null confidence score. Optimum null confidence score, $\text{Conf}(@)$ is determined during training.

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2

Table 4: Training on at6

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	26.3
21993dw	MPFE BBN	26.0
Limsi	MMI	27.0
21993dr	ROVER	22.6

Table 5: Validation on ad6

3.3.3 Maximum Confidence Score

This voting scheme selects the word sequence that has maximum confidence score by setting the value of α to 0.

3.4 PERFORMANCE OF ROVER

ROVER is run on the benchmarking STT systems as shown in [Table 4](#).

3.4.1 The Benchmark STT Systems

Training ROVER on at6 systems that are used as benchmark to compare and analyze different system combination algorithms as explained in [Chapter 2](#) is shown in [Table 4](#). ROVER gives a WER of 24.2 lesser than all the individual WERs of systems combined.

Validation sets for testing the training system combination algorithms is done on ad6 sets which are 6 hours long. The performance of ROVER on validation sets as shown in [Table 5](#) outputs a WER of 22.6 which is lesser than all the individual WERs of systems combined.

3.5 FEATURES OF ROVER

ROVER is based on training a linear equation with two variables that weigh frequency of occurrence of words and word confidences followed by voting. The motivation is to look for system combination algorithms that consider not only frequency of occurrence of words and word confidences but other apriori parameters that can bias speech recognition such as WERs of ASRs combined. Bayesian Combination (BAYCOM) is an algorithm that considers WERs of systems combined and is also based on the classical pattern recognition technique derived from Bayes theorem. Next chapter, BAYCOM at the word level is explored.

4.1 INTRODUCTION

Bayesian Combination algorithm proposed by Ananth Sankar uses Bayesian decision-theoretic approach to decide between conflicting sentences in the outputs of the ASRs combined [20]. BAYCOM proposed is for sentence recognition. BAYCOM is derived from the same principles but applied to word recognition. Bayesian combination differs from ROVER in that it is based on a standard theory in pattern recognition. BAYCOM uses multiple scores from each system to decide between hypothesis. In this thesis, BAYCOM is applied at word level to determine most likely word sequences amongst conflicting word pairs.

4.2 BAYESIAN DECISION THEORETIC MODEL

The following section describes combination at the sentence level. It is different from the ROVER described in chapter 4. Consider M ASRs which process utterance x . Let the recognition hypothesis output by model i be $h_i(x)$. Given sentence hypothesis s_1, s_2, \dots, s_M , the event h corresponding to: Hypothesis h is correct can be written as:

$$h^* = \arg \max_h P(h|h_1, \dots, h_M, s_1, \dots, s_M) \quad (4.1)$$

Since BAYCOM is applied to word recognition, the hypothesis s_1, s_2, \dots, s_M can be substituted as word hypothesis. According to Bayes Theorem, posterior probability,

$$P(h|h_1, \dots, h_M, s_1, \dots, s_M) = P(h) \times \frac{P(h_1, \dots, h_M, s_1, \dots, s_M|h)}{P(h_1, \dots, h_M, s_1, \dots, s_M)} \quad (4.2)$$

since the denominator is independent of h assuming that model hypothesis are independent events, from the above two equations,

$$h^* = \arg \max_h P(h) \prod_{i=1}^M P(s_i|h_i, h) \times P(h_i|h) \quad (4.3)$$

The second term can be distinguished into 2 disjoint subsets as Correct events and Error Events. Therefore, the probability can be written as:

where $P(S_i|C)$ and $P(S_i|E)$ are the conditional score distributions given that the hypothesis h_i is correct and incorrect respectively.

$$\prod_{i=1}^M P(S_i|h_i, h)P(h_i|h) = \prod_{i \in I_C} P_i(C)P(S_i|C) \prod_{i \in I_E} P_i(E)P(S_i|E) \quad (4.4)$$

Multiplying and Dividing by $\prod_{i=1}^M P_i(E)P(S_i|E)$,

$$\prod_{i=1}^M P(S_i|h_i, h)P(h_i|h) = \prod_{i \in I_C} \frac{P_i(C)P(S_i|C)}{P_i(E)P(S_i|E)} \prod_{i \in I_E} P_i(E)P(S_i|E) \quad (4.5)$$

$$h^* = P(h) \prod_{i:h_i=h} \frac{P_i(C)P(S_i|C)}{P_i(E)P(S_i|E)} \quad (4.6)$$

Taking the logarithm,

$$h^* = \arg \max_h \left\{ \log P(h) + \sum_{i:h_i=h} \log \frac{P_i(C)}{P_i(E)} + \sum_{i:h_i=h} \log \frac{P(S_i|C)}{P(S_i|E)} \right\} \quad (4.7)$$

1. $P(h)$ = Probability of the hypothesis from the language model
2. $P_i(C)$ = Probability that model is Correct
3. $P_i(E) = 1 - P_i(C)$ Probability that model is Incorrect
4. $P_i(S_i|C)$ Probability distribution of the hypothesis scores given that the hypothesis is correct
5. $P_i(S_i|E)$ Probability distribution of the hypothesis scores given that the hypothesis is incorrect.

4.2.1 BAYCOM Training

BAYCOM training involves calculating the probability terms in Equation 4.7 for each ASR. These probabilities are used during validation. $P_i(C)$ is the probability of words recognized correctly. This is calculated by comparing the speech output from each ASR to the reference file and recording the number of correct words recognized. $P_i(C) = \frac{N_i(C)}{N_{si}}$, where $N_i(C)$ is the number of correct words and N_{si} is the number of words output by ASR_i. $P_i(E) = 1 - P_i(C)$. $P(S_i|C)$ and $P(S_i|E)$ are calculated by deciding on the bin resolution for the probability scores. The bin resolution for each training session is kept constant. $BIN_RESOL = 1.0/N_B$, where N_B is the number of bins that divide the probability distribution ranging from 0 to 1.0. These parameters are stored for each ASR employed in system combination and used during validation along with the language model probability $P(h)$.

4.2.2 BAYCOM Validation

ASR outputs from the validation set are combined into a single composite WTN. Stored values of probabilities during training are used to calculate a new confidence score according to the BAYCOM equation. The conflicting words in a link are assigned a new BAYCOM confidence score as in Equation 4.7. Maximum confidence score of a word is then chosen as the right word. This occurs when there are missing word outputs from ASRs. A null confidence score is substituted to missing words during training. Also, during training, the null confidence score is varied in a range and tuned for a minimum WER.

Bin Resolution of BAYCOM is tuned for minimum Word Error Rate (WER) during training. Validation sets may have probability scores output from an ASR which do not have corresponding probability distribution of scores in training data. Hence, this results in 0 probability for either $P(S_i|C)$ and $P(S_i|E)$ for a particular word output. To account for missing probabilities, substitution is necessary as the comparison between word sequences is not fair unless all data are available. Hence, smoothing is a method that helps to account for missing probability values.

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.3

Table 6: Training on at6

4.2.3 Smoothing Methods

There are various methods to substitute missing probability values. Some of the methods are to substitute the following for missing probability scores:

- Mean of the confidence scores
- Mean of the neighboring confidence scores whenever available
- Backing off smoothing methods to previous word sequence probability.

4.3 BAYCOM RESULTS

BAYCOM was run on the same benchmarking STT systems to compare performance with ROVER.

4.3.1 The Benchmark STT Systems

ROVER gives a WER of 24.2 lesser than all the individual WERs of systems combined. WER of systems trained by BAYCOM was 23.3 for a bin resolution of 0.01. Next, the optimum bin resolution and nullconf are determined by tuning. Table 6 shows the WERs of ROVER and BAYCOM trained on at6 systems.

4.4 TUNING THE BIN RESOLUTION

In some system combination algorithms, it is necessary to estimate the probability of confidences. The confidences are themselves values between 0 and 1 and their probabilities implies frequency of occurrence of the confidence values. Estimation

of these probabilities is done by computing the histogram. Histogram of confidence values gives frequency table of the latter and hence is used as a good estimate of the sought parameter. Binning of probability values in the range of 0 to 1 is necessary to compute the histogram. Binning the confidences can be large or small depending on the sparsity of the obtained data and the distribution. A smaller value of bin resolution or finer bin resolution is a better estimate of the probability of confidences. Finer bin resolution can lead to 0 bin values when the confidence values in a particular bin are not present. This is not acceptable as log values of probabilities are used and hence $\log 0$ would lead to undefined results hence can lead to errors in recognition.

Alternatively, choosing a larger bin resolution value does not guarantee complete data sparsity but only increases the likelihood of availability of speech data. However, this approximates the parameter sought and reduces accuracy. Therefore, choosing an optimum bin resolution is a trade off between histogram distribution of confidence values and desired accuracy. The employed method is to train baycom for a range of bin resolutions and choose that bin resolution which gives lowest WER. The trained value of bin resolution is considered as the best estimate.

4.5 TUNING NULL CONFIDENCE

If there are missing confidence values then a confidence value of 0 can lead to errors in recognition as log values of probabilities are used and log null is undefined. Hence, it necessitates a substitution of an estimate. This value again is determined for the data set by training baycom for a particular set of words in a range of null confidences. The best null confidence for the training set is determined by choosing the value which corresponds to the best WER.

DETERMINING OPTIMUM NULLCONF: Optimal nullconf is determined as shown in [Table 7](#) shows WER corresponding to varying nullconfs. A bin resolution at 0.1 was fixed and nullconfs were varied between -10 to 3 and nullconf seemed to be insensitive to the output WER.

DETERMINING OPTIMUM BIN RESOLUTION: Next fixing any of the nullconf values, optimal bin resolution is determined by varying bin resolutions in a range. Bin resolutions were varied between 0.01 and 0.3 in steps as shown in [Table 8](#). Nullconf was fixed at 3.0.

EXPT. NO	NULLCONF VALUE	WER
25519	-10.0 to 3.0	23.2

Table 7: Varying Nullconf

BIN RESOLUTION - EXPT 23338	WER
0.01	23.3
0.02	23.3
0.05	23.3
0.1	23.2
0.2	23.2
0.24	23.2
0.26	23.2
0.28	23.2
0.3	23.2

Table 8: Varying bin resolution between 0.0 and 0.3

Hence, the WERs of ROVER and BAYCOM trained for optimum nullconf and bin resolution are shown in [Table 9](#)

4.6 FEATURES OF BAYCOM

BAYCOM at the word level successfully reduces the WER as compared to individual WERs of the combined ASRs. BAYCOM considers Word Error Rate of systems combined as prior probabilities. However, if it was possible to consider ASR performance on each hypothesis words recognized as against individual WERs as prior probabilities then we can expect lesser

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.2

Table 9: Training on at6 - optimum bin and nullconf

approximation to BAYCOM equations. This requires computation of larger set of probability parameters which are granular in approach compared to BAYCOM. A matrix that stores the reference-hypothesis word pairs and their parameters and serves as a look up table is a solution. Next chapter, a novel algorithm called Confusion Matrix Combination based on modification of BAYCOM is proposed.

CONFUSION MATRIX COMBINATION

5.1 INTRODUCTION

System Level Baycom requires computation of probability parameters with respect to each ASR during training ???. The validation algorithm then uses these probabilities that match the probability of word sequences to decide between them. When probabilities relating to word sequences are substituted with probability parameters relating to those at system level, the estimates are approximated. Considering probability parameters corresponding to word sequence pairs are better estimates rather than considering parameters corresponding to system level. Confusion Matrix combination is proposed, which is granular in approach and requires computation of probabilities corresponding to each of the word sequences of each ASRs. This necessitates a larger mechanism of storing information. Hence, a confusion matrix corresponding to each ASR is formulated. The confusion matrix records information of hypothesis-reference word pairs during training phase. No bias between correct and error words are used as in BAYCOM. It is observed that ASRs have a characteristic possibility of confusing certain reference words to particular hypothesis words. Hence, this information is useful in the deductions of Confusion Matrix Combination(CMC).

5.2 COMPUTING THE CONFUSION MATRIX

Consider M ASRs which process utterance x. Let the recognition hypothesis output by model i be $W_i(x)$. For event W corresponding to "Hypothesis W is correct", the best word W^* ,

$$W^* = \operatorname{argmax}P(W|W_1, \dots, W_M, S_1, \dots, S_M) \quad (5.1)$$

where W_1, W_2, \dots, W_M are words from M combined ASRs and S_1, S_2, \dots, S_M are confidence scores corresponding to these words. By Maximum Likelihood theorem, Posterior probability of the

best hypothesis word amongst the hypothesis words from K combined systems is given by,

$$W^* = \arg \max_h \prod_{i=1}^K P(W_i^m | W) P(S_i^m | W_i^m, W) P(W) \quad (5.2)$$

where W_i^m and S_i^m are the Hypothesis and Confidence score of i^{th} ASR. $P(W)$ is the probability of the hypothesis derived from the language model. Taking log,

$$\begin{aligned} \log\{W^*\} = \max_{i \in K} \{ & \sum_{i=1}^K \log\{P(W_i^m | W)\} + \\ & \sum_{i=1}^K \log\{P(S_i^m | W_i^m, W)\} + \log\{P(W)\} \} \end{aligned} \quad (5.3)$$

5.2.1 Confusion Matrix Formulation

CMC considers probability of the hypothesis and corresponding confidence scores at word level. The histogram of confidence scores are computed for each hypothesis-reference word pairs.

A confusion matrix is a 2 dimensional table that match the parameters related to hypothesis words against parameters related to reference words. This is computed for each ASR. Typically two confusion matrices are formulated, one for the probability of words and another for the confidence scores as shown in [Figure 9](#) and [9b](#). The confidence score bins are varied later to search for the optimum bin that gives best WER. The best binning is a trade off between sparseness of data and the granularity or fineness of the bins.

5.2.2 Validation of Confusion Matrix combination

As against the approach of BAYCOM, we do not distinguish between the probability of correct words (diagonal entries in confusion matrix) and the probability of error words or conflicting pairs of words. During validation, the confusion matrix entries are looked up for each hypothesis words and their respective confidence scores. A new confidence value is calculated for each conflicting hypothesis word pairs and the maximum value is chosen.

$$\left\{ \begin{array}{cccc} \cdot & H_1 & H_2 & \dots & H_m \\ R_1 & P(H_1/R_1) & P(H_2/R_2) & \cdot & P(S_{H_m}/R_1) \\ R_2 & P(H_1/R_2) & P(H_2/R_2) & \cdot & P(H_m/R_2) \\ \vdots & \cdot & \cdot & \cdot & \\ R_n & P(H_1/R_n) & P(H_2/R_n) & \cdot & P(H_m/R_n) \end{array} \right\}$$

(a) Probability of Hypothesis-Reference Pairs

$$\left\{ \begin{array}{cccc} * & H_1 & H_2 & \dots & H_m \\ R_1 & P(S_{H_1}/R_1) & P(S_{H_2}/R_2) & \cdot & P(S_{H_m}/R_1) \\ R_2 & P(S_{H_1}/R_2) & P(S_{H_2}/R_2) & \cdot & P(S_{H_m}/R_2) \\ \vdots & \cdot & \cdot & \cdot & \\ R_n & P(S_{H_1}/R_n) & P(S_{H_2}/R_n) & \cdot & P(S_{H_m}/R_n) \end{array} \right\}$$

(b) Probability of Confidences of Hypothesis-Reference Pairs

Figure 9: Building the Confusion Matrices

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.9
23385t	CMC	21.0

Table 10: Training on at6

5.2.3 Validation Issues in Confusion Matrix Combination

Some of the issues in validating Confusion Matrix combination are similar to BAYCOM:

- Missing reference-hypothesis pairs.
- Missing confidence scores of reference-hypothesis pairs.

5.3 CONFUSION MATRIX COMBINATION RESULTS

CMC was trained on at6 benchmark systems, [Chapter 2](#) as shown in [Table 10](#). CMC gives a WER of 21.0 lesser than the individual WERs of all the systems combined. Also, CMC outperforms ROVER and BAYCOM on the training set.

Validation sets for testing the system combination algorithms is done on ad6 sets which are 6 hours long. The STT benchmark systems, [Section 2.2.2](#) are used for validation. We see that ROVER performs better than CMC on validation set which is not tuned for missing parameters in [Figure 9](#) and [9b](#). The comparison of performance of the system combination algorithms is shown in [Table 11](#).

DETERMINING OPTIMUM NULL CONF AND BIN RESOLUTION
CMC is tuned for best bin resolution and null confidence. The bin resolution is varied between 0.01 and 0.2 and corresponding WERs are shown in [Table 12](#). Fixing the bin resolution at 0.2, the null confidences are varied between -10 and 5.0. The WER is not affected by the variation of nullconf as seen in [Table 13](#). Hence, nullconf is fixed at a value of -5.0 which is within the range of [-10,5.0].

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	24.6
21993dw	MPFE BBN	24.6
Limsi	MMI	28.8
21993dr	ROVER	22.6
23385d	CMC	24.4

Table 11: Validation on ad6

EXPT. NO	BIN RESOLUTION VALUE	WER
31446	0.02	21.1
	0.04	21.1
	0.06	21.1
	0.08	21.0
	0.1	21.0
	0.15	20.9
	0.2	20.9

Table 12: Varying bin resolution between 0.02 and 0.2

EXPT. NO	NULLCONF VALUE	WER
24507	-10 to 5.0	20.9

Table 13: Varying Nullconf between -10.0 and 5.0

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.9
31446	CMC	20.9

Table 14: Training on at6 - optimum nullconf and bin resolution

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	24.6
21993dw	MPFE BBN	24.6
Limsi	MMI	28.8
21993dr	ROVER	22.6
24501	CMC	23.4

Table 15: Validation on ad6 - optimum bin resolution of 0.2 and null confidence -5.0

The WERs for optimum nullconf and bin resolution are shown in [Table 14](#) and [Table 15](#)

5.4 FEATURES OF CMC

CMC is based on the pattern recognition theory on the lines of BAYCOM. CMC considers parameters related to specific reference-hypothesis pairs as compared to BAYCOM, which deals with probability parameters related to ASR systems that are employed in system combination. CMC demands more memory as the confusion matrix for each ASR needs to be stored prior to running the experiments. The drawback of CMC over previous methods is that whenever the hypothesis-reference pair parameters are missing, smoothing is necessary. However, with a large available data set significant improvements can be obtained as seen in [Table 14](#).

RESULTS

6.1 ANALYSIS OF RESULTS

ROVER and BAYCOM training was performed on two independent system combination experiments.

1. Combining 2 MPFE and 1 MMI system.
2. The benchmark STT system

6.1.1 *System Combination Experiment Combining 2 MPFE and 1 MMI System*

ROVER was trained on the 3 systems in Table [Table 16](#). Optimum nullconf and alpha are determined. The optimum combination of nullconf and alpha gives the lowest output WER. The optimum parameters were determined in this manner. The values are shown in [Table 17](#)

6.1.2 *BAYCOM Experiment Combining 2 MPFE and 1 MMI System*

BAYCOM was trained on the same 3 systems used to train ROVER [Chapter 3](#). The performance of BAYCOM and ROVER are compared in [Table 18](#). We see that BAYCOM outperforms rover tuned for optimum parameters with an output WER of 13.3 as compared to ROVER's output WER of 13.7.

While training BAYCOM probability of confidence scores of 0 are encountered for a particular ASR implying that the

EXPT. NO	STT SYSTEM	WER
21503tn5	Morpheme based MPFE	14.4
21247tn5	Word based derivative MMI	14.6
21478tn5	Word based MPFE	14.0
21544	Rover combining the above systems	13.7

Table 16: ROVER on MPFE and MMI

Parameters	WER(min)
a = 0.45, c = 0.9	13.7

Table 17: Optimum values of a and c

EXPT. NO	STT SYSTEM	WER
21503tn5	Morpheme based MPFE	14.4
21247tn5	Word based derivative MMI	14.6
21478tn5	Word based MPFE	14.0
21544	Rover combining the above systems	13.7
23338	BAYCOM	13.3

Table 18: BAYCOM combining MPFE and MMI

hypothesis score corresponding to the bin is not encountered. This does not provide correct estimates of BAYCOM confidence for the following reasons:

- 0 probability values cannot be used as, log probabilities are computed in BAYCOM
- The probability of confidence scores depend largely on bin resolution. A higher bin resolution can lead to frequent occurrence of 0 probability scores.

Hence, it is necessary to estimate this value.

DETERMINING OPTIMUM NULLCONF: One method is to check null confidence scores in a range and decide on the best value corresponding to the lowest WER. Table 19 shows WER corresponding to varying nullconfs. A nullconf value of 0.99 gives lowest WER.

DETERMINING OPTIMUM BIN RESOLUTION: Fixing the optimum value of nullconf at 0.99 the bin resolution is varied. Optimum value of bin resolution is checked in similar fashion as nullconf by varying the values in a range of 0 to 1 and searching for the minimum corresponding WER. An optimum bin resolution is a good estimate of probability of confidence scores. Similar to training BAYCOM for various null confidences, the bin resolution values are tested in a range. The combination

EXPT. NO	NULLCONF VALUE	WER
20171	0.0	14.6
	0.1	14.6
	0.2	14.5
	0.3	14.3
	0.4	14.3
	0.45	14.2
20101	0.5	14.2
	0.6	14.1
	0.7	14.1
	0.8	14.0
	0.9	14.0
	0.99	14.0

Table 19: Varying Nullconf between 0.0 and 1.0

of optimum bin resolution and null confidence gives the best performance of BAYCOM. Null confidences were fixed at 1 and best bin resolutions are found to be 0.04, 0.05 and 0.08 as shown in [Table 20](#)

[Table 21](#) and [Table 22](#) shows the summary of training and validation results on at6 and ad6 respectively. It is observed that all the 3 system combination algorithms, ROVER, BAYCOM at and CMC give improvements over each of the three individual ASRs. Observing [Table 21](#), BAYCOM performs better than ROVER on the training sets. CMC outperforms BAYCOM and ROVER on the training sets, however ROVER performs best on the val-

BIN RESOLUTION - EXPT 23338	WER
0.01	13.2
0.02	13.2
0.03	13.2
0.04	13.2
0.05	13.2
0.1	13.3
0.2	13.4

Table 20: Varying bin resolution between 0.0 and 1.0

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.9
31446	CMC	20.9

Table 21: Training on at6 - optimum nullconf and bin resolution

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	24.6
21993dw	MPFE BBN	24.6
Limsi	MMI	28.8
21993dr	ROVER	22.6
24501	CMC	23.4

Table 22: Validation on ad6 - optimum bin resolution of 0.2 and null confidence -5.0

validation sets as shown in [Table 22](#). This is due to sparseness in the validation sets. However, improvements on the training sets indicate that with a larger available training sets better improvements can be achieved. Smoothing is another area that can improve WER when missing word data are encountered.

It is observed that types or errors are distinguished as substitution, deletion or insertion. If it were to be possible to determine the estimate of missing parameters necessary for system combination either during training or validation, then errors can be minimized. Estimating the missing parameters is called smoothing. It was observed that some of the traditional smoothing techniques such as backing off methods as shown in [Table 23](#), or considering the mean of the probabilities as in [Table 24](#), did not provide improvements.

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.9
31446	CMC	20.9
31430	CMC(backing off)	24.2

Table 23: Training on at6 - Smoothing by backing off method

6.2 SMOOTHING METHODS FOR SYSTEM COMBINATION ALGORITHMS

6.3 BACKING OFF

Backing off methods are one of the most popular smoothing methods. Backing off method approximates a conditional probability of preceding words whenever a missing word is encountered. However, backing off did not improve the performance of CMC as shown in Table 23. The WER increased from 20.9 to 24.2 when CMC was trained on at6 benchmark system. 24.2 was the best WER for optimum nullconf of -5.0 and bin resolution of 0.2.

6.4 MEAN OF PROBABILITY OF CONFIDENCE SCORE BINS

Another method of smoothing missing probabilities of confidence scores is to substitute the mean of the histogram of the confidence score distribution, whenever available. , ?? for bin resolutions in the range [0.02 0.2].

$$p_{\text{missing}} = \sum_{i=1}^N \frac{p(S_i)}{N}$$

EXPT. NO	TRAINING MODEL TYPE	WER
21993tm	MPFE BBN System	26.3
21993tw	MPFE BBN	26.0
Limsi	MMI	27.0
21993tr	ROVER	24.2
25519	BAYCOM	23.9
31446	CMC	20.9
24775	CMC(Mean)	21.0

Table 24: Training on at6 - Smoothing by mean of probability of confidence scores

EXPT. NO	VALIDATION MODEL TYPE	WER
21993dm	MPFE BBN	24.6
21993dw	MPFE BBN	24.6
Limsi	MMI	28.8
21993dr	ROVER	22.6
24501	CMC	23.4
31454	CMC(Mean)	23.6

Table 25: Validation on ad6 - optimum bin resolution of 0.2 and null confidence -5.0

CONCLUSIONS

The contribution in the thesis has been mainly towards the theory underlying system combination. The evolution of system combination techniques began by rescoring ASR outputs by other systems leading to a linear combination technique, ROVER and progressing towards application of bayesian techniques. However, as the complexity of the system combination algorithms increases, sparsity of parameters are likely to increase and hence, requires effecient smoothing methods. It was observed that some of the traditional smoothing methods such as backing off, did not provide improvements. ROVER still remains as a powerful system combination technique given that it is simple, and has well established smoothing techniques, such as tuning the null confidence. It gives good improvements once trained for an optimum null confidence. However, improvements in accuracy in the order of 0.1% are considered significant when data set is large, in the order or 100s or 1000s of hours. Though BAYCOM and CMC increase in complexity they provide significant improvements over ROVER on the training sets. Since arabic vocabulary is sparse, the future scope lies in building a larger vocabulary for the algorithms and exploring different smoothing methods to expect good improvements over validation sets from these algorithms. The confusion matrices are sparse as hypothesis-reference pairs may not exist for each of the ASRs. Another technique to explore is word clustering. Identify similar words and forming word clusters among them reduces the dimension. The word probability estimates of the word clusters also helps towards reducing sparsity.

BIBLIOGRAPHY

- [1] M. Ostendorf Et. al. Integration of diverse recognition methodologies through reevaluation of nbest sentence hypotheses. In *Proceedings DARPA Speech and Natural Language Processing Workshop*, page 83–87, 1991. (Cited on page 7.)
- [2] L. Barrault, D. Matrouf, R. De Mori, R. Gemello, and F. Mana. Characterizing feature variability in automatic speech recognition systems. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 5:V–V, May 2006. ISSN 1520-6149. (Cited on page 10.)
- [3] Robert Bringhurst. *The Elements of Typographic Style*. Version 2.5. Hartley & Marks, Publishers, Point Roberts, WA, USA, 2002. (Cited on page 46.)
- [4] Fritjof Capra. *The Science of Leonardo*. Doubleday, USA, 2007. (Cited on page vii.)
- [5] Lin Chase. Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. Eurospeech '97*, pages 815–818, Rhodes, Greece, 1997. URL citeseer.ist.psu.edu/337088.html. (Cited on page 5.)
- [6] Yen-Lu Chow and Richard Schwartz. The n-best algorithm: an efficient procedure for finding top n sentence hypotheses. In *HLT '89: Proceedings of the workshop on Speech and Natural Language*, pages 199–202, Morristown, NJ, USA, 1989. Association for Computational Linguistics. ISBN 1-55860-112-0. doi: <http://dx.doi.org/10.3115/1075434.1075467>. (Cited on page 7.)
- [7] G. Evermann and P. C. Woodl. Posterior probability decoding, confidence estimation and system combination. 2000. (Cited on page 6.)
- [8] G. Evermann and P.C. Woodland. Large vocabulary decoding and confidence estimation using word posterior probabilities. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 3:1655–1658, 2000.

- doi: <http://doi.ieeecomputersociety.org/10.1109/ICASSP.2000.862067>. (Cited on page 5.)
- [9] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354, 14–17 Dec 1997. (Cited on pages 6, 9, 15, and 17.)
- [10] L. Gillick, Y. Ito, and J. Young. A probabilistic approach to confidence estimation and evaluation. *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 2:879–882 vol.2, Apr 1997. doi: 10.1109/ICASSP.1997.596076. (Cited on page 5.)
- [11] D. Giuliani and F. Brugnara. Experiments on cross-system acoustic model adaptation. *Automatic Speech Recognition and Understanding, 2007. ASRU. IEEE Workshop on*, pages 117–122, Dec. 2007. (Cited on page 9.)
- [12] Guru. Mmi training for automatic segmentation of conversational telephone speech ,northeastern university,northeastern university. *MS Thesis*, 2005, August 2006. (Cited on page 1.)
- [13] Dustin Hillard, Bj
"orn Hoffmeister, Mari Ostendorf, Ralf Schl
"uter, and Hermann Ney. irover: Improving system combination with classification. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 65–68, Rochester, New York, April 2007. (Cited on page 9.)
- [14] Hui Jiang. Confidence measures for speech recognition: A survey. In *Proc. Speech Communication*, pages 455–470, 2005. URL <http://www.elsevier.com/locate/specom>. (Cited on page 5.)
- [15] B. Lecouteux, G. Linares, Y. Esteve, and J. Mauclair. System combination by driven decoding. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 4:IV–341–IV–344, April 2007. ISSN 1520-6149. (Cited on page 9.)
- [16] J. Makhoul. Speech processing at bbn. *Annals of the History of Computing, IEEE*, 28(1):32–45, Jan–March 2006. ISSN 1058-6180. doi: 10.1109/MAHC.2006.19. (Cited on page 1.)

- [17] M. Padmanabhan and M. Picheny. Large-vocabulary speech recognition algorithms. *Computer*, 35(4):42–50, Apr 2002. ISSN 0018-9162. doi: 10.1109/MC.2002.993770. (Cited on pages 1 and 6.)
- [18] R.D. Peacocke and D.H. Graf. An introduction to speech and speaker recognition. *Computer*, 23(8):26–33, Aug 1990. ISSN 0018-9162. doi: 10.1109/2.56868. (Cited on pages 1 and 2.)
- [19] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*. Pearson Education, first indian edition, 2004. ISBN 81-297-0272-X. (Cited on page 1.)
- [20] A. Sankar. Bayesian model combination (baycom) for improved recognition. *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 1:845–848, March 18–23, 2005. ISSN 1520-6149. doi: 10.1109/ICASSP.2005.1415246. (Cited on pages 6, 10, and 21.)
- [21] S. Srinivasan and E. Brown. Is speech recognition becoming mainstream? *Computer*, 35(4):38–41, Apr 2002. ISSN 0018-9162. (Cited on page 1.)
- [22] Vincent (2005) Vanhoucke. Confidence scoring and rejection using multi-pass speech recognition. *INTERSPEECH*, 2005. (Cited on page 9.)
- [23] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *Proc. ICASSP 1997*, pages 887–890, Munich, Germany, 1997. URL citeseer.ist.psu.edu/weintraub97neuralnetwork.html. (Cited on page 5.)
- [24] F. Wessel, R. Schluter, and H. Ney. Using posterior word probabilities for improved speech recognition. *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, 3:1587–1590 vol.3, 2000. doi: 10.1109/ICASSP.2000.861989. (Cited on page 5.)
- [25] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 9(3): 288–298, Mar 2001. ISSN 1063-6676. doi: 10.1109/89.906002. (Cited on page 5.)

COLOPHON

This thesis was typeset with $\text{\LaTeX}2_{\epsilon}$ using Hermann Zapf's *Palatino* and *Euler* type faces (Type 1 PostScript fonts *URW Palladio L* and *FPL* were used). The listings are typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera". (Type 1 PostScript fonts were made available by Malte Rosenau and Ulrich Dirr.)

The typographic style was inspired by [Bringhurst's](#) genius as presented in *The Elements of Typographic Style* [3]. It is available for \LaTeX via CTAN as "`classicthesis`".

NOTE: The custom size of the textblock was calculated using the directions given by Mr. Bringhurst (pages 26–29 and 175/176). 10 pt Palatino needs 133.21 pt for the string "abcdefghijklmnopqrstuvwxy". This yields a good line length between 24–26 pc (288–312 pt). Using a "double square textblock" with a 1:2 ratio this results in a textblock of 312:624 pt (which includes the headline in this design). A good alternative would be the "golden section textblock" with a ratio of 1:1.62, here 312:505.44 pt. For comparison, DIV9 of the typearea package results in a line length of 389 pt (32.4 pc), which is by far too long. However, this information will only be of interest for hardcore pseudo-typographers like me.

To make your own calculations, use the following commands and look up the corresponding lengths in the book:

```
\settowidth{\abcd}{abcdefghijklmnopqrstuvwxy}
\the\abcd\ % prints the value of the length
```

Please see the file `classicthesis.sty` for some precalculated values for Palatino and Minion.

Final Version as of April 28, 2009 at 14:50.

DECLARATION

Put your declaration here.

Boston, April 2009

Harish Kashyap
Krishnamurthy