ISBN 978-82-471-4119-9 (printed version)
ISBN 978-82-471-4120-5 (electronic version)
ISSN 1503-8181

NTNU – Trondheim
Norwegian University of
Science and Technology

Line Adde

Doctoral Thesis

Doctoral theses at NTNU, 2013:15

NTNU
Norwegian University of Science and Technology
Thesis for the degree of Philosophiae Doctor
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Electronics and
Telecommunications

NTNU

Doctoral theses at NTNU, 2013:15

Line Adde

A Discriminative Approach to Pronunciation Variation Modeling in Speech Recognition

NTNU – Trondheim
Norwegian University of
Science and Technology

Line Adde

A Discriminative Approach to Pronunciation Variation Modeling in Speech Recognition

Thesis for the degree of Philosophiae Doctor

Trondheim, January 2013

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Electronics and
Telecommunications

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Electronics and
Telecommunications

ISBN 978-82-471-4119-9 (printed version)
ISBN 978-82-471-4120-5 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2013:15

MILJØMERKET
241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Abstract

Put in the most general terms, this dissertation addresses the problem of automatic recognition of non-native proper names. Proper names in themselves tend to pose a severe challenge to speech recognition engines, as these names can typically be pronounced in a variety of ways, and do not necessarily follow generally governing pronunciation conventions. Non-native proper names add still further levels of complication, caused by such variables as the speaker's familiarity with the foreign name, proficiency in the foreign language, and tendency to adapt pronunciation of the name to the native language or, obversely, to adopt foreign speech characteristics in order to pronounce the name as faithfully as possible. When confronted with non-native proper names, it is therefore particularly important for an automatic speech recognition system to be able to handle a considerable amount of pronunciation variety. Traditionally, the more or less self-evident approach to cope with this variety has been simply to add pronunciation variants to the recognition lexicon. However, introducing such variants typically entails the risk of increasing confusability between different lexicon entries, as new variants of previously more distinct units are likely to augment phonetic similarities within the lexicon. It would seem crucial for recognition success, then, to optimize the balance between lexical coverage and confusability. In this work, we strive to attain such a balance by submitting pronunciation variants to selection procedures rather than adding variants to the recognition lexicon indiscriminately.

The selective addition of pronunciation variants to a recognition lexicon has a clear intuitive appeal. It is the objective of this dissertation to confirm that intuition experimentally by measuring the improvements in recognition accuracy yielded by various selection methods. Particularly, we propose a new pronunciation variant selection criterion that is directly related to the effective recognition error rate. To estimate the number of errors corrected by a particular variant, scores based on the Minimum Classification Error framework are calculated before and after the addition of the variant to the lexicon. Using this criterion, three different variant selection procedures are proposed in this work: a *single-pass* approach, an *iterative* approach and a *tree-search* approach. These selection methods aim to optimize the recognition lexicon in terms of size and recognition performance by adding to the lexicon only those pronunciation variants that effect an actual decrease in the error rate. We contrast these selection methods with more traditional approaches to populate the recognition lexicon, such as using all available variants indiscriminately, and selecting on the basis of the probabilities obtained during the generation of possible new pronunciation variants. Our experiments show that we can significantly reduce the error rate and the required number of variants per name by applying our proposed selection approaches.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *philosophiae doctor* (Ph.D.) at the Norwegian University of Science and Technology (NTNU). My supervisor has been Professor Torbjørn Svendsen at the Department of Electronics and Telecommunication at NTNU.

In addition to research activities, the work included compulsory course studies corresponding to one and a half semester of full-time studies, and one year of teaching assistant duties. This work was conducted in the period from September 2006 to October 2012 and has mainly taken place at the Department of Electronics and Telecommunication at NTNU, with the exception of a research visit to the Digital Speech and Signal Processing group (DSSP) of the department Electronics and Information Systems of Ghent University. The duration of this visit was approximately six months which I spent under the supervision of Professor Jean-Pierre Martens.

The work was funded by the Norwegian Research Council as part of the research project *Digitale utfordringer*. Part of this work was conducted in cooperation with ELIS, University of Ghent, in the context of the projects *Autonomata Too* (granted under the Dutch-Flemish STEVIN program) and *TELEX* (granted by Flanders FWO). The one-year period of teaching assistant duties was funded by the Department of Electronics and Telecommunication at NTNU.

Acknowledgements

I would first like to express my gratitude to my supervisor, Professor Torbjørn Svendsen, for his invaluable help, guidance and suggestions throughout this work. Prof. Svendsen never failed to procure promising angles and constructive insights where I discovered none, and without his advice I would not have succeeded in completing this doctoral research.

I am also forever indebted to Professor Jean-Pierre Martens of the DSSP

group at Ghent University. Prof. Martens made it possible for me to become a member of his research group, in the event providing a change of environment and insights which proved to be crucial for many of the findings described in this thesis. But beyond his role in giving my research project new momentum, Prof. Martens deserves my unending gratitude for continuously supporting me, motivating me and challenging me with a level of dedication he might perhaps have been expected to preserve for his own students.

My colleagues at both the Signal Processing Group at NTNU and the DSSP Group at the University of Ghent have my deepest thanks for making my working environment so enjoyable. I would particularly like to mention my office mate Dyre Meen, for his tireless IT support services and for showing me the way to the nearest coffee machine, and for countless discussions, both technical and non-technical. Dr. Ingunn Amdal deserves a special mention for her expert guidance concerning the construction of the corpus used in this work. Among other people who have my greatest appreciation are my friends and colleagues Bert Réveil, Trond Skogstad, Arild Brandrud Næss, Jarle Bauck Hamar, Dr. Alfonso Martinez del Hoyo Canterla, Dr. Sabato Marco Siniscalchi, Andreas Egeberg and Timo Mertens.

My current employer, Max Manus AS, deserves my gratitude for two reasons. Firstly, I was allowed to take sufficient time off to work on this thesis, and my colleagues were always understanding when I was unavailable in order to complete the task. Secondly, a number of my colleagues were so kind as to contribute to the corpus of speech data which I needed as a basis for my experiments. I am much obliged to the board of Max Manus AS for making this possible.

I would like to thank my family and friends for their unlimited support and patience. I would especially like to thank my parents, Sissel and Trond, for raising me with a curiosity for science and for always engaging, and supporting me, in all my pursuits. Finally, and most importantly, I would like to express my deepest gratitude to Dries. Over the last five years he has been my critical editor, my expert linguistic consultant, my human dictionary, my rehearsal audience, my motivator, my punishing bag, my shoulder to cry on, my personal support group, my travel companion, my mood-equalizer, my survival kit, my comfort, my date, and above all, my best friend. It is safe to say that without his unyielding love and confidence, I would have been lost. I love you Dries, this one is half yours.

Oslo, October 2012
Line Adde

Contents

Chapter 1

Introduction

Automatic speech recognition (ASR) may be defined as the process of retrieving the most probable word sequence from an acoustic speech signal by means of a computer, or, simpler still, as the conversion of speech to text. Even though it is probably fair to say that state-of-the-art ASR systems do not as yet perform on a par with the natural capability of most humans to understand spoken language, a vast amount of research over the past decades has resulted in a number of successful real-world ASR applications. Many of these applications, such as car navigation, travel assistance and directory assistance applications, rely on accurate recognition of proper names. Many proper names, however, pose a severe challenge to the recognition engine because of a considerable mismatch between the way the proper name is pronounced by the user and the way the name is represented in the ASR system through acoustic models and phonetic transcription lexicons. The main reason for this is that proper names have a tendency to deviate from conventional pronunciation rules, and often allow for a variety of valid pronunciations. As those pronunciation rules are used to populate the lexicons that are employed by the ASR engine, the irregularly pronounced proper names are at an instant disadvantage.

It is true that this disadvantage can be partially remedied by investing additional manual effort, viz. by constructing proper name dictionaries. However, while this can be extremely beneficial for relatively limited domains such as medical terminology (which also tends to diverge from generic pronunciation rules), the vast number of existing proper names makes it difficult to attain sufficient coverage for all possible user inputs in that domain. Furthermore, while domains such as medical terminology are typically governed by pronunciation conventions, proper names are quite likely to be pronounced in a variety of ways, making the manual dictionary construction effort even greater and more costly. The acoustic realization of a proper name is dependent not only on speaker characteristics such as gender, age and dialect, which may be said to influence all natural speech, but also on the speaker's familiarity with the name and the specific entity it refers to. To give but one example, only speakers who are in some way familiar with Cambridge University's Magdalen College will correctly pronounce the name as "maudlin".

Those observations are *a fortiori* true for non-native names, which are part and parcel of many real-life ASR applications. Referring back to the examples mentioned above: a navigation device will be of limited use if its ASR breaks down as soon as the car crosses a language border, a travel assistance application should cover more destinations than those in its own language area, and a directory assistance application should be able to handle the names of immigrants as well as native names. Such names are singularly challenging to ASR engines since they have an even larger variety of possible pronunciations. Over and above the speaker characteristics listed above, the acoustic realization of non-native proper names will depend on such factors as the speaker's proficiency in the non-native language, the speaker's tendency to adopt speech patterns from the non-native language, and the similarity of the native language and the non-native language.

An intuitive method to enable an ASR engine to handle such large-scale variation would be to add more variants to the recognition lexicon. Generating the variants manually might give the greatest increase in recognition performance, but such an effort is extremely costly and, given the amount of speaker-dependent variability, nigh on infinite. It may therefore be desirable to predict plausible pronunciations automatically, e.g. by allowing some controlled deviations from standard phonetic transcriptions. Such an approach has a foreseeable disadvantage, however: indiscriminately adding automatically generated transcriptions to a recognition lexicon is likely to increase the level of confusability within that lexicon. Newly introduced variants of previously distinct entries may be more phonetically similar, which effectively makes it harder for the recognition engine to tell the one from the other, so to speak. A method to optimize proper name lexicons is therefore in order: the ideal is to build lexicons that allow for the greatest degree of variation, at the lowest level of confusability.

The main objective of the present dissertation is to model the pronunciation variation observed when native Norwegian speakers pronounce English proper names. In an attempt to model this variation at the lexical level, we will investigate various methods to optimize our proper name lexicons. Our first approach will be to consider the population of our lexicon as a decision problem, where the decision criterion is directly associated with the system's recognition error rate. This approach allows us to add the variants that correct the most errors to the lexicon first. In our second approach a similar decision strategy will be employed in an iterative manner in order to evaluate the net effect of a lexicon change. Effectively, only variants that correct more recognition errors than they introduce are added to the dictionary when using this approach. A third approach will employ a tree-search algorithm to decide the order in which variants are added to the lexicon. In this way, we will prioritize the names where additional variation in the lexicon has the highest potential to improve recognition performance. The upshot of these approaches is a recognition lexicon that is optimized in terms of recognition performance as well as lexicon size. In the remainder of this introduction we will give a short summary of the main contributions of this work and outline the organization of this dissertation.

1.1 Contributions of this dissertation

This dissertation aims to improve pronunciation variation modeling in automatic speech recognition. We investigate the problem of pronunciation variation and highlight problematic areas in existing modeling techniques. To provide solutions to some of these problems, we define a set of criteria to determine a lexicon's effect on recognition performance, and propose some approaches to generating a lexicon with a positive effect. The main contributions of this dissertation are summarized below.

A new language resource for Norwegian containing annotated speech utterances of non-native names

This dissertation is specifically concerned with the pronunciation variation in English proper names as pronounced by native Norwegian speakers. When this work was started, no suitable language resource containing this kind of utterances was available. We have therefore compiled a new resource containing annotated speech utterances of English names spoken by Norwegians. This resource is used in our experiments, where we attempt to model the variation that is present in the speech data. A detailed description of the construction of this resource was published at the LREC conference in 2010 [1].

An initial study of pronunciation variation of non-native proper names

In an initial study on the nature of pronunciation variation in non-native proper names, we attempt to get a better understanding of the properties we might expect in a lexicon, and more generally in a pronunciation variation modeling scheme, that allows for good recognition performance. This study contains threshold and "cheating" experiments as well as two baseline experiments, and highlights several problem areas frequently observed in traditional pronunciation variation modeling methods.

A comparative study of different variant selection criteria

One of the most challenging tasks of lexical pronunciation variation modeling is to identify which pronunciation variants to include in the recognition lexicon. A particular pronunciation variant might correct a number of recognition errors, but if it is phonetically similar to a variant of another lexicon entry, it might also introduce new recognition errors. We perform a detailed comparative analysis of four different variant selection criteria, in order to find the criterion best suited for optimal variant selection. These four selection criteria are based on: probabilities obtained during variant generation, the acoustic log likelihood of the variant, the Maximum Entropy framework and the Minimum Classification Error framework. The analysis shows that the Minimum Classification Error (MCE) framework yields the most promising results, and we therefore develop this selection criterion further into a "breadth-first" approach and a "best-first" approach. This comparative study was published in an article presented at the IEEE Workshop on Spoken Language Technology in 2010 [2].

A breadth-first variant selection approach

The selection criterion in our "breadth-first" variant selection algorithm is directly related to the recognition performance: variants are only added to the lexicon on the condition that they correct recognition errors left unhandled by the initial lexicon. We start with a lexicon containing one variant per name, and use this lexicon to perform a recognition pass. For each variant in this lexicon, we calculate an MCE-score, which can be understood as a measure of the risk of misrecognizing a speech utterance. The larger this measure is, the larger the risk for an incorrect classification. In the next iteration, we successively add one variant to the lexicon, run a recognition pass with the extended lexicon, and calculate a new MCE-score for that variant. This process is then repeated for all available variants. When this iteration is done, we rank all variants per name by their MCE-scores. Applying our "breadth-first" approach, we then add the top-ranked variant for each name to the lexicon, provided its MCE-score is lower than that of the variant added for the same name in the previous iteration. In this way, we continue to iterate over the available variants, until we find we can no longer improve the recognition performance by adding variants to the lexicon. The breadth-first variant selection approach was presented at the Interspeech conference in 2010 [3]

A best-first variant selection approach

At the end of each iteration loop, our "breadth-first" approach evaluates for each name in the lexicon whether it is beneficial to add its top-ranked pronunciation variant. However, since the level of observed pronunciation variation strongly differs from name to name, we might do better with a "best-first" variant selection algorithm that prioritizes inclusion of variants for names where most variation is expected. To that end, the pronunciation variant selection task is recast as a tree search problem where the optimal recognition lexicon corresponds with the optimal path through a search tree. To guide the search algorithm, we again define an MCE-based discriminative evaluation function. In this approach, the evaluation function can be understood as a measure for the ratio of errors corrected by the addition of a variant versus errors still remaining unaddressed by the lexicon. This selection approach is described in an article published at the ICASP conference in 2011 [4].

1.2 Outline of this dissertation

This dissertation is organized as follows. Chapter 2 gives an overview of a typical ASR system and its basic components. It also contains a short description of the statistical framework on which a typical ASR engine is based, as well as an introduction of two alternative discriminative frameworks. Chapter 3 consists of three main parts. In the first part, the problem of accurate name recognition is described in greater detail. The second part of the chapter consists of a comprehensive survey of some of the methods reported in the literature to remedy these problems, whereby special emphasis is placed on lexical pronunciation variation modeling methods. The final part presents a general outline of the methods proposed in this dissertation. Chapter 4 describes the design and collection of the Norwegian NameDat corpus and gives an overview of the experimental set-up used in this work. This chapter starts the experimental part of this dissertation with an initial study, investigating the limitations of some existing lexical pronunciation variation modeling approaches. Chapter 5 contains an evaluation of four different criteria for variant selection. Chapter 6 describes a more sophisticated variant selection algorithm that amounts to a "breadth-first" approach to populate the recognition lexicon. In Chapter 7, the pronunciation variant selection problem is recast as a "best-first" tree search problem. Finally, Chapter 8 contains overall conclusions and some thoughts regarding future research within the field of lexical pronunciation modeling.

Chapter 2

Automatic Speech Recognition

This chapter presents an overview of a typical Automatic Speech Recognition (ASR) system. In the first section, the basic components of the ASR system will be described in some detail. In the second section, the statistical formulation of the speech recognition problem, based on the classical Bayes' decision theory, will be examined more closely. In this section, the speech recognition problem will first be redefined as a classification problem and then again as a simple distribution estimation problem. The classification problem will then be expanded to incorporate the case when a lexicon entry is represented by multiple alternative pronunciation variants. In the final part of this chapter, two discriminative alternatives to the classical Bayes' decision rule will be described.

2.1 The Automatic Speech Recognition system

Automatic speech recognition may be described as the process of retrieving the most probable word or word sequence $\hat{W}$ from an acoustic observation x. Most ASR systems today achieve this by implementing a procedure similar to the one illustrated in the block diagram in Figure 2.1. In this block diagram the preprocessing step maps the acoustic speech signal x to an alternative vector representation of the speech signal, which is called X. This mapping reduces the variability of the speech signal, making it better suited for ASR. This vector representation is then given as input to the decoder, which is the heart of the ASR engine. The decoder, then, retains the most probable word sequence from the speech vector by utilizing three different knowledge sources: a language model, a pronunciation lexicon and a set of acoustic models. These knowledge sources and the other components of the block diagram in Figure 2.1 will be described in further detail in this section.

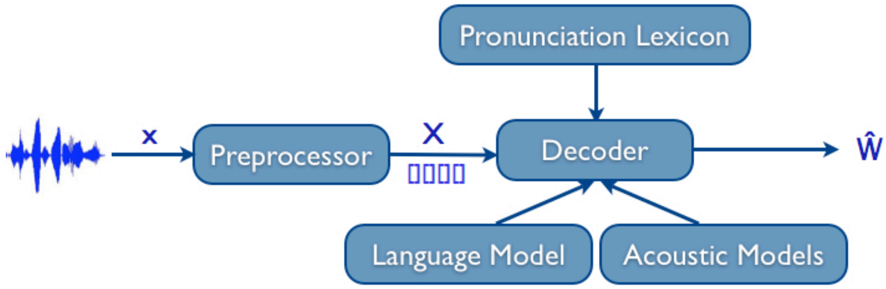

Figure 2.1: *Block diagram of an Automatic Speech Recognition system.*

2.1.1 The preprocessor

A speech signal usually contains a vast amount of variation due to various factors such as different acoustic environments, different speakers and different speaking styles. The main purpose of the preprocessing block in Figure 2.1 is to reduce this variation by extracting only the information that is relevant for the speech recognizer to retrieve the underlying sequence of words from the speech signal. This information is represented in a compact form as a sequence of feature vectors. By discarding irrelevant information such as speaker and environmental characteristics, the amount of training data needed to model the speech signal can be drastically reduced.

The extraction of feature vectors is typically performed by dividing the speech signal into a sequence of overlapping segments usually referred to as frames. The most common frame length is 25 milliseconds and the frame shift is normally around 10 milliseconds. A feature extraction procedure is then performed on each speech frame under the assumption that the speech signal is a *stationary random process* within this frame. This *short-time stationarity* assumption means that we assume that the statistical characteristics within this frame do not vary with time. The validity of this assumption, however, depends on several factors such as the speech sound and the relative placement of the frame. Vowels, for example, can be said to be fairly stationary sounds. Plosives, on the other hand, are highly non-stationary. If the frame is placed between two different sounds, the speech signal will also most likely be non-stationary within this frame. The result of this procedure is a sequence of feature vectors with a much lower dimension than the original speech signal.

To enhance the performance of the ASR system, temporal information, in the form of time derivatives of the feature parameters, is normally added to the set of feature parameters. The most common temporal parameters are the first order derivative (delta) and the second order derivative (delta-delta or acceleration). To further enhance the performance, a measure of the energy within a frame can also be added to the feature set to augment the spectral parameters derived during feature extraction.

There are several different kinds of feature extraction algorithms. The most commonly used algorithms are: the Mel-Frequency Cepstral Coefficients (MFCC) [5] algorithm, the Linear Predictive Coding (LPC) [6] algorithm and the Perceptual Linear Predictive (PLP) [7] algorithm. As these algorithms are all based on the estimation of the spectral envelope, they rely heavily on the stationary assumption to be valid for all frames. This is not always the case for frames containing speech segments, however, which poses a limitation for all the above-mentioned algorithms.

2.1.2 The language model

The language model has two main tasks in a speech recognition system, namely to define which words the recognizer should be able to recognize, the *vocabulary* of the system, and to constrain the numerous ways in which these words can be combined into word sequences. By introducing constraints on the way in which words can be combined into sentences, the language model simplifies the task of the decoder and thereby increases recognition performance. The downside of this is that putting constraints on sentence formation actively restricts the user's freedom of expression. The best compromise between the recognition performance and the user's freedom of expression is highly dependent of the application. For instance, when composing a personal letter, the complexity of the word sequences is fairly high and the user therefore needs a high degree of freedom. In a travel assistance application where the task is to recognize which train station the speaker wants to travel to and from, on the other hand, the complexity of the word sequence is low, allowing for a more restrictive language model.

Generally, language models are divided in two main categories; deterministic language models and probabilistic language models. The deterministic models (also known as grammar-based models) are the simplest form of language models and are normally employed when the number of sentences to be recognized is small and of low complexity. In these models, all words are typically deemed equally likely, as is the probability of any one word following any other word. Isolated word recognition tasks employ the most basic form of deterministic language models, allowing only a single word to be recognized and assuming all the words in the vocabulary to be equally likely.

A probabilistic language model aims to model the probability of an entire word sequence by estimating the probability of a certain word given the presence of the preceding words in the sequence. In principle, this means that any word can follow any other word in the vocabulary, but with a certain probability. This probability is calculated using the count of the particular word sequence in some given training text. The most common probabilistic language model is the N*-gram model*, which assigns a probability to a certain word given the $N-1$ previous words in the word sequence. The probability of the word sequence $\mathcal{W} = w_1, w_2, \ldots, w_m$ is estimated by the N-gram model as

$$\hat{P}(w_1, w_2, \ldots, w_m) \simeq \prod_{i=1}^{m} \hat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}), \tag{2.1}$$

where the conditional probabilities are usually estimated using the count $C(\mathcal{W})$ of a certain word sequence $\mathcal{W}$ in the training text

$$\hat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})}. \tag{2.2}$$

Obviously, the amount of probabilities that need to be estimated is dependent the size of N. Given that the probabilities are normally estimated from a bank of text data, it is currently deemed unfeasible to collect enough text data to satisfactorily train N-gram models of a higher complexity than three or four.

A problem that is common in language modeling is the case when no N-gram samples can be found in the training text. According to Equation (2.2), these N-grams will be given a probability of zero, although the actual probability should be higher. This is usually solved by redistributing a small probability mass from N-grams observed in the training data to N-grams not observed in the training data. This method is called *discounting*. The most commonly used discounting method is the Good-Turing discounting method [8].

Another common problem in language modeling is the lack of relevant training material (which is especially the case if N is large). If there is not enough evidence of a particular N-gram in the training texts, the probability estimation of this N-gram is likely to be somewhat unreliable. One way of solving this is to introduce a *back-off* scheme. Using a back-off scheme means that if an observed N-gram count is less than a predefined cutoff, the scaled probability of the shorter context (the $(N-1)$-gram) will be used instead and the N-gram will be deleted form the model. This method ensures that the language model always contains reliable N-gram probability estimates, in addition to making the model more compact.

2.1.3 The pronunciation lexicon

The main task of the pronunciation lexicon is to inform the recognizer of the most typical pronunciations of every word in the vocabulary. To achieve this, the pronunciation lexicon lists all the words in the vocabulary with at least one *phonetic transcription* describing the pronunciation of the word. A phonetic transcription is a sequence of phonetic symbols, each of which is a written representation of a particular predefined sound. These phonetic symbols can be sub-word units such as phones, diphones, triphones, syllables or larger units such as words, syllables and phrases. The small sub-word units have the benefit that the number of unique units necessary to model all the words in the vocabulary is significantly smaller compared to the larger units. This is a good property because it means that less training data is needed to get a satisfactory acoustic representation of the unit. The disadvantage of the small sub-word units, however, is that they do not contain as much contextual information as is comprised in the larger units. The pronunciation lexicon can contain several phonetic transcriptions for each word in the vocabulary. In this dissertation, we will refer to these phonetic transcriptions as *pronunciation variants* of a particular word.

2.1.4 The acoustic model

The objective of the acoustic model is to model the relation between the preprocessed speech signal and the underlying sequence of words represented by a string of phonetic symbols. Due to the high variability within the segment, the acoustic model must be able to handle both temporal and spectral variation within a speech segment, as well as differentiate between different speech segments.

A popular and well-suited statistical model that can model a series of discrete observations, as well as handle both temporal and spectral variations, is the hidden Markov model (HMM) illustrated in Figure 2.2. The HMM is commonly viewed as an extension of the Markov chain. The Markov chain is a model that consists of a set of states, $\mathcal{S} = \{1, \ldots, S\}$, a set of probabilities of starting in one of these states, $\boldsymbol{\pi} = \pi_i$, $i \in \mathcal{S}$, and a set of probabilities of moving between those states, $\mathbf{A} = a_{ij}$ where $i, j \in \mathcal{S}$. The number of states in $\mathcal{S}$ is largely dependent on the size of the sub-word units used by the recognizer. Most state-of-the-art ASR systems employ *context-dependent* acoustic models, where each sub-word unit is modeled using the unit's left and right context. For phone-based context-dependent systems, a three-state topology is typically employed, where the first state is used to model the beginning of the phone, the second state to model the middle part of the phone and third state to model the last part of the phone. These systems also normally employ a left-to-right topology which does not allow revisits to previous states, as indicated by the arrows in Figure 2.2. At every time unit t, the Markov chain changes state and generates an observation X_t. The observations generated from a Markov chain are deterministic in the sense that the exact same observation will be generated every time a given state is entered.

In the *hidden* Markov model, however, that is no longer the case. In this model, every time state i is entered, an observation vector X_t is generated from a probability distribution $\mathbf{b} = b_i(X_t)$, as illustrated in Figure 2.2. This means that the same state can generate different observation vectors each time it is entered. It is therefore not possible to say with certainty which state sequence generated the observed vector, i.e. the state sequence is hidden. For example, in Figure 2.2 the observation sequence $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ was generated by the state sequence $\mathbf{S} = \{1, 2, 2, 3\}$. By looking only at the observation sequence, there is no way of knowing whether the observations were generated by the state sequence $\mathbf{Q} = \{1, 2, 2, 3\}$, $\mathbf{Q} = \{1, 1, 2, 3\}$ or $\mathbf{Q} = \{1, 2, 3, 3\}$.

In automatic speech recognition it is generally assumed that the observed feature vector, $\mathcal{X} = \{X_1, X_2, \ldots, X_T\}$, is generated by an HMM defined by the parameters $\Phi = (\mathbf{A}, \mathbf{b}, \boldsymbol{\pi})$. In such a system, each state is usually defined to represent either a sub-word unit or a part of a sub-word unit. The goal, then, becomes to retrieve the hidden state sequence in order to reveal the underlying sequence of sub-word units that represents word W. Before describing how we can find this state sequence, let us first define the event of being in a state at time t as q_t and the probability of this event as $P(q_t)$. A state sequence of length T can then be given as $\mathcal{Q} = \{q_1, q_2, \ldots, q_T\}$. The *acoustic likelihood* of the observed feature vector can now be found by summing over all possible state sequences

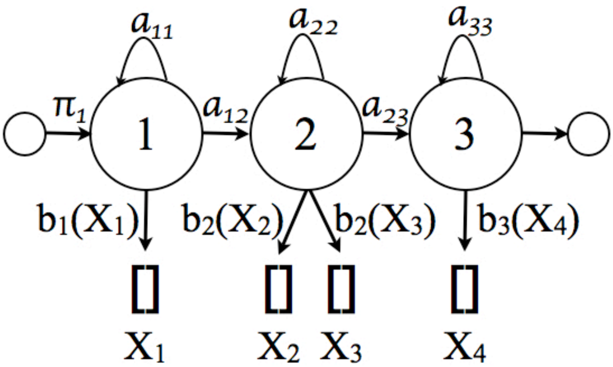

Figure 2.2: *The hidden Markov model.*

$$P(X|W,\Phi) = \sum_{Q} \pi_{q_1} b_{q_1}(X_1) \prod_{t=2}^{T} b_{q_t}(X_t) a_{q_t q_{t+1}}. \tag{2.3}$$

As calculating the likelihood in this manner can be computationally quite expensive, most ASR engines approximate the likelihood calculation by considering only the most likely state sequence

$$\hat{P}(X|W,\Phi) \approx \max_{Q} \Big\{ \pi_{q_1} b_{q_1}(X_1) \prod_{t=2}^{T} b_{q_t}(X_t) a_{q_t q_{t+1}} \Big\}. \tag{2.4}$$

Using the same state sequence definitions used above, we can define the probability for starting in state i, the initial state distribution π_i, as

$$\pi_i = P(q_0 = i).$$

The state transition probability a_{ij} is the probability of going from state i to state j and is defined as

$$a_{ij} = P(q_t = j|q_{t-1} = i).$$

This simple equation assumes that the transition probability at time t is only dependent on the previous state and not on any state sequences prior to $t = t - 1$. This assumption is also known as the *Markov assumption*

$$P(q_t|q_{t-1}, q_{t-2}, \ldots, q_0) = P(q_t|q_{t-1})$$

and any state sequence following this assumption is called a first order Markov chain.

Finally, in most HMM systems the output probability densities, **b**, are represented by the Gaussian Mixture Model (GMM). The GMM is a probability density function comprising several Gaussian mixture component densities combined together in a weighted sum. Given that the GMM contains M mixture components for all states, and that each of these components are weighted with a weight c_{im}, the output probability density for state j at time t can be defined as

$$b_j(X_t) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(X_t; \mu_{jm}, \Sigma_{jm})$$

where $\mathcal{N}$ is a multivariate normal distribution with mean vector μ_{jm} and covariance matrix Σ_{jm}.

Before the HMM model can be employed in any useful task, the parameters $\boldsymbol{\pi}$, **A** and **b** need to be estimated. An elegant solution to this problem is the Baum-Welch re-estimation algorithm [9][8]. This algorithm estimates the model parameters by choosing the parameters that maximize the likelihood of the training utterances. To make this selection, the algorithm first estimates some initial parameter values by making a rough guess of what the parameters might be. The likelihood of the training data is then calculated using these initial parameters. More accurate parameter values are then found by re-estimating the parameters iteratively using the likelihoods calculated in the previous step.

Limitations of the hidden Markov model

Although hidden Markov models have shown to be very well-suited for automatic speech recognition, there are some limitations of using this statistical framework to model the speech signal.

The HMM framework assumes that an observation is conditionally independent from its neighboring observations, which is clearly not true in the case of the speech signal, since most speech signals contain a large amount of correlation from one frame to the next. Furthermore, the Markov assumption stating that the probability of moving from one state to another is dependent only on the previous state, is also an approximation, since these dependancies normally extend over several states in a speech segment.

2.1.5 The decoder

The heart of an ASR system is the decoder. The decoder combines information from all the knowledge sources (i.e. the lexicon, the acoustic models and the language model) to create a recognition network of HMM states, connected with transitions. For an unknown speech utterance, each path trough this network represents a hypothesis of what was actually said. It is then the task of the decoder to identify the most likely path through this network. Perhaps the most obvious way to do this would be by calculating the likelihood of every path in the network and then simply choosing the most likely path. This, however, is unfeasible in practice due to the large amount of computations expected to be performed in close to real-time. A computationally efficient search algorithm, called the *Viterbi algorithm*, was therefore designed [9][8]. The Viterbi algorithm solves this problem by exploring only the most promising path using dynamic programming. After finding the most likely path, the algorithm backtracks trough the network and finds the corresponding word or word sequence.

N-best lists

In many cases, it is desirable to generate a list of the N most probable hypotheses, rather than just the most probable one. This list is called an *N-best list* and contains the N hypotheses deemed most likely by the decoder and their corresponding likelihood scores. The N-best list provides us with additional information about the recognition performance. Not only does it tell us whether an utterance was correctly recognized or not, it also tells us whether a misrecognized utterance was close to being correctly recognized or not. In most speech recognition systems, N is a user-defined parameter. Most state-of-the-art speech decoders, however, employ some sort of *pruning* technique to reduce the search space. In the HTK toolkit [10], which is used in this dissertation, the pruning is implemented by keeping a record of the log likelihood of this path through the recognition network, and excluding from the search any path with a log likelihood score that differs significantly from that of the best path. As a consequence, the actual value of N may vary from utterance to utterance. For example, if the recognizer is fairly confident in the recognition result, the N-best list will most likely contain relatively few hypotheses for the recognized utterance. An unreliable recognition result, however, will produce longer N-best lists, containing a relatively high number of hypotheses.

2.2 Bayes' decision theory

As mentioned in the beginning of this chapter, the task of an ASR system can be defined as retrieving the most probable word or word sequence $\hat{W}$ given an acoustic observation X. In this section, we will focus on finding the optimal word hypothesis in the case of isolated word recognition, by utilizing probabilities derived from the knowledge sources described in the previous section and Bayes' decision theory.

If $\mathcal{W} = \{W_1, W_2, \ldots, W_K\}$ is the set of possible words from which the optimal word hypothesis $\hat{W}$ is to be selected, the problem of finding $\hat{W}$ can be redefined as the classification of the acoustic observation X into one of K predefined classes. The classifier must identify a mapping, $C(X)$, from the parameter space to the discrete word space $\mathcal{W}$, that minimizes the number of misclassifications. This mapping is generally referred to as the speech recognizer's *decision rule.* Using this decision rule, it is possible to estimate the word $\hat{W}$ most likely to have been spoken in utterance X

$$\hat{W} = C(X).$$

To find the decision rule that minimizes the number of misclassification events, a measure of the classifier's performance must be defined. This performance measure is most commonly represented by a loss function, modeling the cost of classifying the incorrect word W as $\hat{W}$. For most speech recognition systems, this loss function is a zero-one function assigning equal loss to all misclassifications

$$l(\hat{W}, W) = \begin{cases} 0 & \hat{W} = W \\ 1 & \text{otherwise.} \end{cases}$$

Assuming that the true joint distribution $P(W, X)$ is known, it can be shown that the optimal classifier that minimizes the expectation of this loss function is the one that employs the following decision rule [11]:

$$C(X) = \operatorname*{argmax}_{W \in \mathcal{W}} P(W|X),$$

better known as the *maximum a posteriori (MAP)* decision rule. Thus, to implement the optimal MAP classifier, knowledge is required about the a posteriori probabilities $P(W|X)$. In practice, however, these probabilities are never exactly known, and they need to be estimated from a set of training examples. Since the a posteriori probabilities are relatively hard to estimate, the decision rule above can be rewritten using Bayes' rule

$$\begin{aligned} C(X) &= \operatorname*{argmax}_{W \in \mathcal{W}} P(W|X) \\ &= \operatorname*{argmax}_{W \in \mathcal{W}} \frac{P(X|W)P(W)}{P(X)} \\ &= \operatorname*{argmax}_{W \in \mathcal{W}} P(X|W)P(W). \end{aligned}$$

The last simplification can be done without loss of generality, since the distribution $P(X)$ is not dependent on the word sequence and is therefore not affecting the maximization operation. The *likelihood*, $P(X|W)$, and the *prior* probability, $P(W)$, are far less complicated to estimate than the posterior probability. In most classification problems, estimating the prior probabilities amounts to a straightforward computation. Estimating conditional likelihoods, on the other hand, tends to be a more difficult problem, especially when the dimensionality of the observation vector X is large. For that reason, the conditional likelihood function is often represented in a parametric form. For example, suppose that the true data distribution of the conditional likelihood is a normal density with mean μ and covariance Σ. This single piece of information reduces our estimation problem from estimating an unknown function $P(X|W)$ to estimating only two parameters, μ and Σ.

There are several ways in which to solve this estimation problem. The most common procedure is the *maximum-likelihood* estimation procedure. The general principle of the maximum likelihood procedure is to choose the distribution that maximizes the likelihood of obtaining the observed training samples. An attractive feature of the maximum likelihood estimation is its simplicity. Let us assume that a conditional likelihood function is in fact a normal density function described by the parameter vector $\theta = \{\mu, \Sigma\}$. To indicate the dependency of $P(X|W)$ on θ, we rewrite $P(X|W)$ as $P(X|W, \theta)$. Assuming that the parameters for each class (i.e. word) are functionally independent, we can further simplify the notation by removing the indications of class distinction. $P(X|W, \theta)$ then becomes $P(X|\theta)$. The problem, then, is to estimate the parameter vector θ using the information provided in the training samples. To illustrate, suppose we have a set of training samples $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ which are independently drawn from the density $P(X|\theta)$. The likelihood function $P(X|\theta)$ can then be defined as

$$P(X|\theta) = \prod_{n=1}^{N} P(X_n|\theta)$$

and the log likelihood function as

$$LLH(\theta) = \ln P(X|\theta) = \sum_{n=1}^{N} \ln P(X_n|\theta).$$

The solution to the maximum likelihood estimation problem can now formally be written as

$$\hat{\theta} = \operatorname*{argmax}_{\theta} LLH(\theta).$$

The maximum likelihood estimate for θ can then be obtained by solving the equations

$$\nabla_{\theta} LLH(\theta) = 0.$$

More details on maximum likelihood estimation may be found in [12].

In most present-day automatic speech recognition systems, the parametric form chosen to represent the speech signal is the hidden Markov model introduced in Section 2.1.4. The likelihood function most commonly used in ASR is therefore the one described by Equation (2.4). In this model, the parameters to be estimated are the initial states π, the state transition probabilities $\mathbf{A}$ and the output probability densities $\mathbf{b}$. As mentioned in Section 2.1.4, these parameters are usually estimated using the Forward-Backward estimation algorithm also known as the Baum-Welch re-estimation algorithm. More details of these algorithms can be found in [9][12]. The prior probabilities $P(W)$ are typically modeled by means of the N-gram model (Equation (2.1)). In isolated word tasks, however, where the prior probabilities are assumed to be uniformly distributed, the prior probabilities can be dropped without affecting the maximization operation.

2.2.1 Expanding the MAP decision rule to exploit alternative pronunciations

In many speech recognition applications, it is desirable to have several alternative pronunciation variants for every entry in the lexicon, in order to cover more than one possible pronunciation of the corresponding words. Let the set of words in the vocabulary be defined as $\mathcal{W} = \{W_1, W_2, \ldots, W_K\}$. Then, for every word $W_k \in \mathcal{W}$ there are $I(k)$ pronunciation variants $\mathcal{V}_k = \{V_{k1}, V_{k2}, \ldots, V_{kI(k)}\}$ representing W_k in the lexicon. These alternative pronunciation variants can be incorporated into the MAP decision rule as follows:

$$\begin{aligned}\hat{W} &= \operatorname*{argmax}_{W_k \in \mathcal{W}} P(X|W_k)P(W_k) \\ &= \operatorname*{argmax}_{W_k \in \mathcal{W}} \sum_{i=1}^{I(k)} P(X; V_{ki}|W_k)P(W_k) \\ &= \operatorname*{argmax}_{W_k \in \mathcal{W}} \sum_{i=1}^{I(k)} \frac{P(X; V_{ki}; W_k)}{P(W_k)} P(W_k) \\ &= \operatorname*{argmax}_{W_k \in \mathcal{W}} \sum_{i=1}^{I(k)} P(X|V_{ki}; W_k)P(V_{ki}|W_k)P(W_k). \qquad (2.5)\end{aligned}$$

Here, $P(X|V_{ki}; W_k)$ is the acoustic likelihood of the observation X given word W_k and the pronunciation variant V_{ki}, and $P(V_{ki}|W_k)$ is the probability of pronunciation variant V_{ki} being selected to represent word W_k. In the remainder of this dissertation, this probability will be referred to as the *pronunciation prior probability*. Both of these distributions are typically unknown and need to be estimated from training data. $P(W_k)$ is the probability of word W_k, which is generally uniformly distributed.

2.3 Discriminative approaches in ASR

As described in the previous section, the theoretical optimality of the MAP decision rule relies on the basic assumption that the posterior probability or, equivalently, the class conditional likelihood and the prior probability, are precisely known. This assumption, however, is often not valid in practical applications. There are three main reasons why this is so. Firstly, for computational reasons, the class conditional likelihood $P(X|W)$ often needs to be represented in a parametric form. This parametric form is normally limited by tractable computation and is therefore only an approximation of the true data distribution. Secondly, the estimation of the parameter set describing the data distribution might not always be optimal. Finally, lack of training data may result in a further mismatch between the estimated parameters and the true data distribution.

For these reasons, indirect learning of model parameters through Bayes' rule is in fact rarely optimal in practice. Discriminative classifiers aim to circumvent these problems by learning model parameters directly through maximizing the posterior probability or some other discriminative function. In the remainder of this chapter, two such discriminative classifiers will be described, namely the Maximum Entropy (ME) classifier and the Minimum Classification Error (MCE) classifier.

2.3.1 The Maximum Entropy Classifier

Maximum Entropy (ME) modeling was first proposed by Jaynes [13] in 1957 and has since been successfully applied in several areas within the field of speech technology. It has proved particularly effective in natural language processing where the ME model has been employed in areas such as language modeling (e.g. [14] and [15]), part of speech tagging, machine translation and language understanding (e.g. [16]). Some efforts have also been made to use Maximum Entropy to discriminatively train the acoustic parameters of automatic speech recognizers (e.g. [17] and [18]).

Most discriminative classifiers attempt to maximize the posterior probability

$$C(X) = \operatorname*{argmax}_{W \in \mathcal{W}} P(W|X)$$

directly. In the Maximum Entropy classifier, the posterior probability $P(W|X)$ is modeled to satisfy the *maximum entropy principle* [19]. This principle states that, given a set of distributions, we should always choose the distribution with the maximum entropy (the most uniform distribution) that satisfies a set of constraints. Entropy is defined as a measure of the uncertainty of a probabilistic distribution. The higher the entropy is, the more uncertain a distribution is, and the more information it contains. The maximum entropy principle ensures that by always choosing the distribution with the maximum uncertainty, the ME model makes as few assumptions about the true distribution as possible.

To introduce domain-specific knowledge into the model, a set of feature values can be extracted from a training set. The feature constraints are usually formulated using a set of binary feature functions. These feature functions return the value 1 if the predicted word $\hat{W}$ corresponds with the actual word W, and if the observation X satisfies some condition $b(X)$

$$f_i(X, W) = \begin{cases} 1 & W = \hat{W} \text{ and } b(X) \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

It can be shown [19] that the parametric model that satisfies the feature constraints while maximizing the entropy is a well-defined log-linear model given by

$$\tilde{P_\Lambda}(W|X) = \frac{1}{Z(X)} \exp\left[\sum_{i=1}^{F} \lambda_i f_i(X,W)\right]$$

where $f_i(X,W)$ are the feature constraints, F is the number of features in the set, $\Lambda = \{\lambda_i\}$ is a set of feature weights and $Z(X)$ is a normalization factor used to ensure that $\sum_{W \in \mathcal{W}} \tilde{P_\Lambda}(W|X) = 1$

$$Z(X) = \sum_{W \in \mathcal{W}} \exp\left[\sum_{i=1}^{F} \lambda_i f_i(X,W)\right]. \tag{2.6}$$

The feature weights λ_i are found by maximizing the log likelihood of the model $P_\Lambda(W|X)$

$$\lambda_i^* = \underset{\lambda}{\operatorname{argmax}}\, LLH(P_\Lambda).$$

Given a set of labeled training samples (X_n, W_n), the log likelihood of the model predicting the real distribution is defined by

$$LLH(P_\Lambda) = \sum_{n=1}^{N} \ln \tilde{P_\Lambda}(W_n|X_n) = \sum_{X,W} \tilde{P_\Lambda}(X,W) \ln \tilde{P_\Lambda}(W|X)$$

where the probability distribution $\tilde{P_\Lambda}(X,W)$ is the normalized number of occurrences of the pair (X, W) in the training samples.

For non-trivial problems, numerical methods need to be employed to find the optimal parameter set. The most commonly used algorithms are the Generalized Iterative Scaling (GIS) algorithm [20] and the faster Improved Iterative Scaling (IIS) training algorithm [21]. Recently, the Limited-Memory Variable Metric (L-BFGS) method [22] has also been found to be effective for optimizing maximum entropy parameters [23].

2.3.2 The Minimum Classification Error Classifier

The Minimum Classification Error (MCE) classifier was first proposed by Juang and Katagiri [24] to overcome the fundamental limitations of classifiers based on distribution-estimation described earlier. While the MCE method has proven to be effective within several areas of ASR (such as language model training [25][26], pronunciation variation modeling [27] and speaker identification and verification [28][29]), it has mainly been used to estimate the acoustic parameters of the speech recognizer. Juang *et al.* argued in [30] that although the HMM model is a reasonable model for speech observations, it cannot be explicitly proven that it is the true distribution form for speech, and for this reason using the discriminative MCE method for training HMM parameters is the most appropriate choice. Their study showed that training HMM parameters using the MCE method gave a superior performance over the traditional distribution-estimation method.

In the MCE framework, the classifier design is directly linked to the actual classification error rate. The goal of the classifier is therefore to correctly discriminate observations for the best recognition result, rather than to fit distributions to the observations. The main difference between conventional classifiers and the Minimum Classification Error classifier is that the latter is based on discriminant functions rather than distribution estimation. In theory, these discriminant functions may or may not be related to posterior probabilities or conditional probabilities. In many practical speech applications, however, these functions are often related to the log likelihood scores of class $C(X)$. The main purpose of the discriminant function is to determine the classifier's decision rule. For example, using a set of discriminant functions, $g_l(X;\Lambda)$, defined by the parameters Λ, a general form of the word classifier can be defined as

$$C(X) = W_k \quad \text{if} \quad g_k(X;\Lambda) = \max_l g_l(X;\Lambda) \tag{2.7}$$

where word W_k is recognized if the k-th discriminant function is the largest one for utterance X. The MCE classifier design is then performed in such a way that minimizing the expected loss of recognition accuracy directly relates to the minimization of the classification error rate. To achieve this, a three-step estimation procedure is employed.

In the first step, an error criterion expressing the decision rule of Equation (2.7) in a functional form is specified. To that end, there exist many possible functions. Juang *et al.* [30] used the *class misclassification measure*, $d_k(X_{kn};\Lambda)$, defined as

$$d_k(X;\Lambda) = -g_k(X;\Lambda) + \log\left[\frac{\sum_{j,j\neq k} e^{g_j(X;\Lambda)\eta}}{K-1}\right]^{\frac{1}{\eta}} \tag{2.8}$$

where K is the number of classes. This misclassification measure compares the discriminant function of the correct class (k) with that of the competing classes. A positive misclassification measure, therefore, indicates that the chance of a misclassification is higher than the chance of a correct decision and a misclassification measure less than zero indicates the reversed situation. In Equation (2.8), the parameter η is a positive number and acts as a tuning parameter. Increasing this parameter gives preference to competitors with higher likelihood scores, and setting this parameter to infinity will force the misclassification measure to only consider the best competitor.

In the second step, a loss function is defined by mapping the misclassification measure to a zero-to-one continuum, making it more suitable for optimization

$$l_k(X;\Lambda) = \frac{1}{1+e^{-d_k(X;\Lambda)}}. \tag{2.9}$$

If the loss is close to zero, the utterance is likely to be correctly recognized given the parameters Λ. The larger the measure is, the larger the risk for an incorrect recognition of the utterance.

Finally, to estimate the parameters of the classifier, the overall performance of the classifier needs to be defined. The expected loss of an arbitrary model Λ is generally used for this purpose. The expected loss is calculated as the sum of contributions from all classes $l(X_n;\Lambda)$ emerging from the available training utterances $X_n \in \mathcal{X}$:

$$\mathcal{L}(\mathcal{X};\Lambda) = \frac{1}{N}\sum_{n=1}^{N} l(X_n;\Lambda) \tag{2.10}$$

Following this three-step procedure results in a classifier design where the classifier parameters, Λ, can be optimized in terms of minimum expected loss. Various minimization algorithms can be used to minimize this function. Most studies rely on the powerful generalized probabilistic descent (GPD) algorithm to perform this task. More details concerning this algorithm may be found in [24].

Chapter 3

Proper name recognition

One of the most difficult and complex problems in Automatic Speech Recognition (ASR) today is posed by proper names. Many ASR applications, such as car navigation, travel assistance and directory assistance applications, rely on accurate recognition of proper names. In this chapter, we will describe several elements that can have a detrimental effect on the performance of these applications and give an overview of some of the methods reported to improve the performance of similar applications in the literature. In the first part of this chapter (Section 3.1), the problem of accurately recognizing proper names, and non-native names in particular, will be introduced. In Section 3.2-3.5, an overview of methods previously reported to solve some of these challenges will be discussed. These methods are divided into four main categories namely; methods that reduce the perplexity of the ASR system (Section 3.2), methods that model pronunciation variation at the lexical level (Section 3.3), methods that model pronunciation variation at the language model level (Section 3.4) and methods that model pronunciation variation at the acoustic level (Section 3.5). In the final part of this chapter (Section 3.6), an introduction and a general outline of the name recognition approach proposed in this dissertation will be given.

3.1 The challenge of accurately recognizing proper names

A common problem for most name recognition systems is that there often tends to be a mismatch between the way a proper name is pronounced by the user and the way the name is represented in the ASR system through acoustic models and phonetic transcriptions.

There are several reasons for this mismatch. The first reason is related to inaccurate name transcriptions at the lexical level. This is due to the fact that many proper names do not follow conventional pronunciation rules, which makes the prediction of reasonable name pronunciations a challenging task. This problem can be solved by incorporating manually constructed transcriptions into the lexicon. This, however, not only tends to be infeasible from a budgetary perspective, but it also requires expert knowledge in both the native language and languages of other origins present in the name set.

A second reason for this mismatch is that proper names usually have a large number of valid pronunciations. This pronunciation variation can be attributed to several factors such as; the origin of the name, the speaker's familiarity with the name, linguistic background of the speaker and the name, among other factors. This means that even if the lexicon contains an accurate representation of the name in question, the system will still fail to recognize the name if the speaker is not familiar with the name and chooses a different name pronunciation.

A third reason for the mismatch between the ASR system and the actual pronunciation of a name, is that many name recognition systems contain a considerable number of non-native names. Non-native proper names are singularly challenging since they have an even larger variety of valid pronunciations. An individual speaker's pronunciation of a non-native name is likely to be influenced by several sociocultural factors. Eklund and Lindström [31] mention regional background, gender, education and age as important in this regard. Other underlying factors mentioned by the same authors are: the speaker's knowledge of the target language and other foreign languages, the speaker's expectations of the listener's knowledge of the target language, the social status of the speaker and listener, the time and the place the name first appeared, the familiarity of the name through media and travel, the population of name bearers and the similarities between the languages involved. Both Fitt [32] and Eklund and Lindström [31] point out that speakers also tend to use non-native sounds to a varying degree, depending among other things on their knowledge of the name's origin and the origin language. Trancoso *et al.* [33] found that speakers actually can alter their pronunciation of a name towards the target language, or towards another foreign language they know well, even with a limited knowledge of the language in question.

Another problem one often comes across in name recognition applications is that the number of names to be recognized can be very large. Many name recognition systems can effectively contain several hundred thousand name entries, which makes the name recognition task a high perplexity problem. It is a well known fact that when the vocabulary of an ASR system increases, lexical entries start to resemble each other which decreases the recognizer's chance of selecting the phonetic transcription of the correct word. This was illustrated by Kamm *et al.* in [34] where they investigated the feasibility of having 1.5 million names in a directory assistance application. The study showed that the name recognition accuracy decreases logarithmically with the increasing number of names in the vocabulary.

Inaccurate lexical representations, variations in pronunciation, the use of non-native sounds and high perplexity pose severe challenges to the ASR engine. These issues can be partially remedied by either reducing the perplexity of the system or by modeling the pronunciation variation within the ASR system. This pronunciation modeling can be done either at the lexical level, at the acoustic level or at the language model level. Lexical pronunciation variation modeling has been thoroughly investigated the last decades and is still regarded as an important problem. The topic was given special attention in 1998 to 2002 through workshops organized in Rolduc (the Netherlands) in 1998, Sophia-Antipolis (France) in 2001 and Estes Park (Colorado, USA) in 2002, and will also be the main focus of the literature overview given in this chapter.

In the following sections we will give an overview of some of the lexical, acoustic and language model approaches to pronunciation variation modeling given in the literature. Although special attention will be given to the modeling of variation seen in proper names and non-native speech, relevant work in lexical pronunciation variation modeling will also be surveyed. Some attention will also be given to studies attempting to reduce the perplexity of large vocabulary name recognition tasks.

3.2 Reducing the perplexity of large vocabulary systems

As mentioned in the previous section, Kamm *et al.* [34] showed that the name recognition accuracy decreases logarithmically with increasing number of names in the vocabulary. Gao *et al.* [35] observed the same decrease in performance when they studied the performance of their directory assistance system which contained about 280,000 names. This study also showed that having more pronunciation variants in the lexicon (without changing the vocabulary size) can be beneficial. An error analysis of unsuccessful calls performed in this study revealed that over half of the errors made in their system was attributed to names either not in the vocabulary or to names which were hard to pronounce (many of which were of foreign origin). The authors tried to tackle these issues using techniques such as speaker clustering, massive acoustic adaptation of previous calls, unsupervised sentence adaptation and pronunciation modeling methods.

Other studies have tried to introduce restrictions on the user in order to improve the performance of large vocabulary directory assistance applications. In both Meyer and Hild [36] and Hild and Waibel [37], the caller was restricted to spell the first names and/or last names in a telephone-based name recognition system. Kellner *et al.* [38] attempted to circumvent the perplexity problem by introducing restrictions on the vocabulary using a hierarchical dialog structure which reduced the vocabulary considerably in every step of the dialog. Another approach to tackle the high confusability in directory assistance applications has been to use meta data such as caller ID [39] and available data such as lists of friends and relatives, place names and other relevant information [40], to attain prior information on which names are likely to appear. Both approaches reported large performance improvements on the name recognition task. Béchet *et al.* [41] [42] attained a noticeable improvement in name recognition performance by re-scoring N-best hypotheses generated by a directory assistance system developed at France Telecom R&D. In this approach, alternative word hypotheses were found by traversing a phoneme lattices generated from the re-scored N-best lists.

3.3 Modeling pronunciation variation at the lexical level

Modeling pronunciation variation at the lexical level usually entails adding multiple phonetic transcriptions of all the words in the vocabulary to the base lexicon. The rationale behind this is that having several pronunciation variants for each word in the lexicon considerably increases the recognizer's chance of selecting a variant corresponding to the correct word. When a lexicon covers a high amount of observed pronunciation variation, the lexicon is said to have a high phonetic coverage. Having too many pronunciation variants in the lexicon, however, can also introduce new errors since pronunciations of different words start to resemble each other. This effect is commonly referred to as lexical confusability. Several studies have confirmed that having a high lexical confusability will introduce more recognition errors and increase the decoding time ([43], [34], [44], [35] and [45]). Studies performed by Yang and Martens [44] and Kessens *et al.* [45] showed that adding multiple pronunciation variants to the lexicon is only beneficial up to a certain point. In fact, both studies showed that when the number of variants per word exceeds 2.5, the system starts performing worse than a system using only one variant per word.

A successful lexical modeling scheme should therefore construct a lexicon that contains a good balance between phonetic coverage and confusability. In an attempt to simulate such a lexicon, McAllaster *et al.* [46] conducted a "cheating" experiment where they added all the phonetic transcriptions of the test set to the base lexicon. When tested on data fabricated from acoustic models, the experiment showed a considerable performance increase of when compared to using only the base lexicon. In a similar experiment, Saraçlar *et al.* [47] achieved a relative performance improvement of 43% by using a lexicon comprising phonetic transcriptions extracted from performing phoneme recognition on the test set. These experiments show that having the correct variants in the recognition can improve the performance substantially given that there is a match between the acoustic models and the pronunciations in the lexicon.

To optimize the recognition lexicon, most lexical modeling schemes today employ a two-phase optimization process comprising a *generation* phase which generates a set of pronunciation candidates and a *selection* phase which selects the pronunciation candidates most likely to correct more errors than they introduce. The generation phase is often divided into three sub-phases. In the first phase, linguistic evidence is retrieved from either existing knowledge sources or from actual speech data. In the second phase, formalizations are usually derived from the information obtained in the previous phase. These formalizations are then used in the third sub-phase to generate a set of pronunciation variants. The selection phase then aims to retrieve from this set the variants most likely to perform well. In the work described in this dissertation we will mainly focus on this last phase; the selection phase. In the remainder of this section we will therefore only give a brief overview of the literature regarding the generation phase and give more attention to the selection phase.

3.3.1 Generation phase

Gathering information

To model pronunciation variation accurately it is necessary to obtain some evidence of the variation that is most likely to occur. This information can either be collected from existing knowledge sources (*knowledge-based approaches*) or from actual speech data (*data-driven approaches*). Both of these approaches can be divided into *direct approaches* and *indirect approaches.* The direct approaches extract new pronunciations for the words in the vocabulary directly from available data sources. Indirect approaches, on the other hand, aim to infer a set of formalizations from the data and use these to generate pronunciation variants for both seen and unseen words.

In direct knowledge-based methods, alternative pronunciations can be extracted from several existing lexical and linguistic resources. One direct knowledge-based approach is to simply retrieve pronunciation variants from existing electronic pronunciation lexicons. Obviously, this approach is limited in that it will not be able to model pronunciation variation for words not in these lexicons. An example of an indirect knowledge-based approach, is to generate pronunciation variants from a set of pronunciation rules derived from linguistic knowledge. These rules can be built manually or extracted from existing knowledge sources using techniques such as Classification And Regression Trees (CART). Knowledge-based approaches to pronunciation variation modeling are often criticized for being expensive in terms of human resources, having an insufficient coverage of pronunciation variation and being unable to model the frequency of the variations found. Rules derived from knowledge sources are also commonly known to be quite ill-suited for modeling non-standard speech e.g. dialect, conversational and non-native speech. Moreover, for many languages (such as Norwegian) not many suitable lexical and linguistic resources are available.

Data-driven approaches to pronunciation variation modeling aim to exploit real speech data by extracting the pronunciations observed in a data set. These transcriptions can then either be used directly in recognition lexicons or indirectly by deriving a set of formalizations from them. Indirect approaches often rely on smaller segments than words and will therefore also give more reliable estimates and generalize better. One advantage of data-driven approaches, as opposed to knowledge-based approaches, is that they enable us to compute probabilities for the variants. A disadvantage of these approaches is that they can suffer from generalization problems and produce variants that are too specific for the employed data set.

Deriving formalizations

As discussed above, when employing indirect pronunciation variation modeling, pronunciation variants are generated using a set of formalizations. These formalizations are usually derived by aligning a set of reference transcriptions with a set of alternative transcriptions. These alternative transcriptions can be derived either from existing knowledge bases or from speech data. The reference transcriptions can be manually derived transliterations of a speech utterance or a phonetic transcription retrieved from an available pronunciation dictionary. The alignments are usually performed by means of some form of dynamic programming algorithm. After the alignment, differences between the reference transcription and the alternative transcription, such as substitutions, insertions and deletions, will be used to derive pronunciation rules, train neural networks, train decision trees and calculate confusion matrices [48]. Of these approaches, the most common procedure is to derive a set of rules. These pronunciation rules are normally phone-to-phone mappings as well as phone insertion and deletion rules. The phone mappings can also be associated with a cost based on the statistical co-occurrence of phones, as introduced in [49]. The rules are typically defined in a certain context and most approaches use the left and right neighboring phones of the target phone. Rule probabilities are often calculated using frequency counts and help to assess the accuracy of the rule and to later calculate probabilities for the variants generated using a specific set of rules. Finally, a pruning step can be performed to exclude rules with a probability under a predefined threshold.

Generating pronunciation variants

Pronunciation variants can be generated manually, extracted from various available lexical resources or generated automatically using one of the formalisms described above. There are many different strategies for variant generation reported in the literature (see [48] for an overview), many of which use one of the following procedures; rules, artificial neural networks, phoneme recognizers, decision trees, grapheme-to-phoneme (g2p) converters or Finite-State Transducers (FST).

Several studies rely on context-dependent decision trees to generate a set of pronunciation variants (e.g. [50], [51], [52], [53], [54], [55], [56]). These trees are used on a phone-by-phone basis to generate alternative pronunciation variants from canonical pronunciations. The canonical pronunciations can be hand-labeled phoneme sequences [55], phoneme sequences emerging from a phoneme recognizer ([50], [51], [52] and [56]) or a sequence of articulatory features obtained using articulatory feature models [54] and [53]. An approach using decision trees to generate foreign accented pronunciation variants, was proposed by Goronzy *et al.* in [56]. In this study, English-accented variants were generated for German words by decoding German speech with an English phoneme recognizer and training a decision tree on a set of standard German reference transcriptions and the variants emerging from the phoneme recognizer. The resulting decision tree was then used to find English-accented variants from German transcriptions which later were added to a German base lexicon. Experiments with native English speakers uttering German words, showed that this procedure achieved better results than the base lexicon comprising only German base variants. The study further suggested that even better results might be achieved if speaker-dependent decision trees were to be used.

Grapheme-to-phoneme (g2p) conversion, also commonly referred to as letter-to-sound (l2s) conversion, is an important component of many ASR and text-to-speech applications for predicting pronunciations of lexicon entries. Traditionally, these converters have been created manually using a set of phonological rules derived from linguistic knowledge. Today, many state-of-the-art g2p converters are based on statistical models representing the relationship between graphemes and phonemes, and the relationship between different graphemes (usually modeled by N-gram models). These statistical models can be trained using available pronunciation lexicons that contain large amounts of word-pronunciation pairs. Most g2p converters, however, are not able to predict accurate pronunciations for words that do not follow conventional pronunciation rules (e.g. proper names and accented or non-native speech). Some studies have tried to solve this problem by retraining the g2p converters (e.g. [57], [58]). Badr *et al.* [59] recently proposed a way of learning new pronunciations from a set of spoken utterances by integrating a known g2p conversion technique with acoustic examples. Despite the fact that the acoustic material was quite noisy (it was collected using an Internet-based crowdsourcing method), this technique achieved a performance surpassing a lexicon comprising manually created baseforms.

Recently, several efforts have been made to implement phonological rules using Finite-State Transducers (e.g. [60], [61], [62], [63]). In these studies, phonological rules were used to transform baseform pronunciations into a graph of alternative pronunciations which were then incorporated into an FST-based recognition system. These rules described phonological variations such as place and voicing assimilation, gemination, silence insertion, alveolar stop flapping, schwa deletion, vowel devoicing, etc. ([60], [62]) and were mainly derived on the basis of expert knowledge. Hazen *et al.* [61] derived rules using information from various levels in the linguistic hierarchy such as morphology, part-of-speech, tense, lexical stress, syllable structure and phonemic content. Livescu and Glass [63] derived simple context-independent rules from a limited set of training data to model phonetic confusions in non-native speech. A non-native pronunciation graph was then generated using these rules, and incorporated as an additional resource in an FST recognizer. Using simple rules and a limited set of training material, an absolute word error rate reduction of 2.1% was achieved in this study.

3.3.2 Selection phase

Many of the variant generation approaches described in the previous subsection result in an exceedingly large number of new pronunciation variants. As discussed in the beginning of this section, adding multiple pronunciation variants to the lexicon increases the recognizer's chance of selecting the correct word at the risk of introducing unwanted confusion between lexicon entries. Over the past few decades, many studies have been conducted to reduce this confusability. These methods can usually be divided into two groups; *indirect selection methods* and *direct selection methods*. The indirect selection methods assess the pronunciation rules used to generate a pronunciation variant whereas the direct selection methods assess the variant directly.

Indirect selection methods aim to prune out unlikely pronunciation rules during variant generation hence reducing the number of variants generated. One indirect method of reducing confusability in this way was proposed by Cremelie and Martens [64] and was later used with some success by Yang and Martens [65]. In this approach rules were organized in a hierarchical manner and frequency counts were used to eliminate specific rules that could be covered by more general rules. Another approach using frequency of occurrence to select the best performing rules was described by Kessens *et al.* in [45]. Amdal [49] used improvement in log likelihood scores as a measure to assess the performance of a set of pronunciation rules. This measure compared the acoustic log likelihood scores of phonetic transcriptions affected by a pronunciation rule with the corresponding score for the reference transcription. The resulting log likelihood ratio score was then used as a rule pruning measure. Applying this rule pruning measure in the generation of new variants, yielded a result outperforming a traditional rule pruning method based on rule probability, using even fewer rules.

Many studies have employed a direct method to determine which pronunciation variants to include in the lexicon and consequently various variant selection criteria have been proposed. In the selection approach proposed by Riley *et al.* in [55], a forced alignment procedure was used on a set of training utterances and the pronunciation variants that were chosen often by the recognizer were then included in the lexicon. Slobada and Waibel [66] used a phone confusion matrix to eliminate the variants that only differed in confusable phones from the lexicon. In Torre *et al.* [67], a confusability matrix was used in combination with word confusions to reject highly confusable variants. Confidence measures have been used by many authors (e.g. [50] and [68]) to select the variants most likely to cause the least confusion. In Holter and Svendsen [69], a maximum likelihood criterion was used to assess the pronunciations generated from a phoneme recognizer. While several approaches have been based on the maximum likelihood criterion, this study aimed to maximize the *joint likelihood* of the training data. This approach resulted in an error rate reduction of up to 18.4% relative.

Some efforts have also been made to define metrics that can calculate the confusability of a lexicon. Wester and Fosler-Lussier [70] defined such a metric by using a forced alignment procedure on a set of training utterances and a lexicon comprising the phonetic transcriptions under evaluation. This procedure resulted in a phone transcription for every utterance in the training data. This phone transcription was then used to obtain every sequence of variants in the lexicon that matched any substring in this phone transcription, producing a lattice of possible word sequences. Finally, the confusability was calculated by summing all "confused" phones for each phone and dividing this number by the total number of phones in the alignment. Although there was no direct correlation between this confusion metric and the error rate, it was useful as a variant selection criterion, providing an 8% relative reduction in word error rate and a substantial decrease in decoding time. This confusability metric was later used by Fosler-Lussier *et al.* [71] to create a framework for predicting and simulating speech recognition errors of unseen words in an isolated word task. The authors' goal was that this framework could help predict what phonological variations are likely to increase the lexical confusability and to train new and better pronunciation and language models in the future. A recent approach employing this framework to simulate recognition errors to be used in discriminative language model training was proposed by Jyothi and Fosler-Lussier [72].

It has been shown that pronunciation variation is highly dependent on prosodic factors (such as speaking rate, phone duration, fundamental frequency, signal-to-noise ratio, etc.) in addition to phonetic context [73]. This dependency is often used to argue for the use of dynamic lexicons, where the pronunciation of a word is determined dynamically during recognition using linguistic context information. Finke and Waibel [74] proposed a dynamic lexicon where the pronunciation prior probability varied as a function of the speaking style. In this study, phone duration was found to be the most important cue to pronunciation variability. Fosler-Lussier and collaborators have done a great deal of work on building dynamic lexicons using decision trees to select the pronunciation variants that are most likely to perform well in the current linguistic context ([75], [50], [51] and [52]). In this work, the linguistic context is modeled by building one decision tree for every basic recognition unit. In phone-based decision tree systems, each decision tree is related to a particular phone and predicts how the phone is realized in context. These trees use context cues such as the identity of the neighboring units, duration of a unit, speaking rate and word predictability to generate a finite-state grammar for the unit. These grammars are then concatenated during recognition to form one large grammar for the entire utterance. The best pronunciation is then found dynamically by employing this model in an acoustic re-scoring decoder which re-scores the initial list of hypotheses.

However, there are some problems with the approaches mentioned above. Firstly, in many of these approaches the number of variants per lexicon entry is pre-determined and equal for all entries. This does not seem to be the best approach for the name recognition task, where the variation in pronunciation varies substantially depending on which name is being uttered. Secondly, to select the best performing pronunciation variants from a set of candidates it is desirable to have a way of measuring the effect a particular pronunciation variant will have on the error rate. In most variant selection approaches there is no direct relationship between the selection criterion and the actual recognition error rate, which makes it hard to measure this effect. Finally, it can be argued that the optimal lexicon should not only contain pronunciation variants that correct more errors than they introduce, it should also contain complementary pronunciation variants, correcting different types of recognition errors. The optimal variant selection criterion is therefore a criterion that identifies the pronunciation variants that both corrects the most recognition errors *and* corrects different recognition errors than the variants already in the lexicon.

Vinyals *et al.* [27] proposed to adopt the Minimum Classification Error (MCE) criterion for selecting the most distinctive pronunciation variants. In this study a phone recognizer was employed to generate a set of pronunciation candidates. These candidates were then evaluated by calculating a MCE score for every candidate on the basis of likelihood scores emerging from the phone recognizer. By using the MCE criterion to select variants, this approach uses a selection criterion that is directly related to the recognition performance. This procedure was tested in a large vocabulary setting using short utterances, many containing proper names such as street and city names, yielding an overall sentence error rate reduction of 2.6% absolute.

3.3.3 Lexical pronunciation variation modeling: the case for proper names

Automatic recognition of native and non-native proper names is a notoriously difficult problem due to the large amount of variation seen in the pronunciation of these names. It can therefore be especially beneficial to model this variation at the lexical level, as described in the preceding sections. The majority of the studies on this topic in the literature concentrate on generating new, or improving existing, phonetic transcriptions to accurately represent names in the recognition lexicon.

A knowledge-based approach, generating a set of additional phonetic variants for city names from five European languages, was proposed by Schaden ([76] and [77]). In these studies, foreign accented phonetic transcriptions were obtained by applying a set of phonological rewrite rules to available native canonical transcriptions. These rules were context-based phoneme/allophone mapping tables that modeled systematic mispronunciations commonly occurring when non-native speakers uttered native city names.

A data-driven approach to modeling the variation seen in proper names, is to generate a set of alternative pronunciations from audio samples. Ramabhadran *et al.* [78] described an algorithm that automatically generated speaker-dependent pronunciation variants from acoustic samples of names in a name dialing application. In this algorithm, the variants were generated from the audio samples using an HMM-based ballistic labeler which constructed a lattice of sub-phone units from the speech utterance. The probability of moving from one node (phone) to another in this lattice was then determined by weighting the probability stored in a phone transition model (trained on a database of names) with the score obtained from the HMM. Adding the newly created variants to the recognition lexicon resulted in a performance surpassing that of a lexicon comprising hand-written variants. The same approach was later reused in another large vocabulary name recognition task in Gao *et al.* [35]. In this study, adding the derived pronunciations to the recognition lexicon gave an improvement in error rate of 2.28% absolute when tested in a large vocabulary setting on a test set comprising 5,700 native and non-native name utterances.

Several studies over the last years have relied on general purpose grapheme-to-phoneme (g2p) transcriptions, or modified versions of these transcriptions, to represent native and non-native proper names in the lexicon. A pronunciation-learning algorithm utilizing both audio samples and linguistic information contained in g2p transcriptions was reported in Beaufays *et al.* [79]. In this algorithm, alternative proper name pronunciations were generated from a set of audio samples using a speech recognizer and acquired linguistic knowledge. The algorithm started by applying a g2p converter to all the names in the vocabulary. These variants were then forced aligned with training utterances of that name and posterior probabilities were calculated for each phone. The phone which had the lowest posterior probability was then successively replaced by each phone in the phone set to form a set of alternative pronunciation variants. To select the most promising variants among these variants a joint optimization procedure was employed. This procedure simultaneously maximized the acoustic likelihood of the variant as well as its linguistic probability, which was calculated using the initial pronunciation probability and a linguistic model derived from an existing ASR dictionary. Adding the resulting variants to the recognition lexicon yielded an error rate reduction of up to 44% relative compared to a reference g2p lexicon.

In [33], Trancoso *et al.* were faced with the problem of decoding French place names spoken by German speakers (and vice versa) in a car navigation system. They noticed that their general purpose German and French g2p converters covered only a small portion of the pronunciation variation observed in their database. To solve this problem, alternative “nativized” pronunciations for the non-native names were generated using special g2p rewrite rules derived statistically from data and previous know-how from the ONOMASTICA project [80].

In another study, performed by Cremelie and ten Bosch [81], the task of accurately recognizing non-native proper names was solved by using state-of-the-art g2p converters for multiple languages (Dutch, English and French). Using these converters, three g2p transcriptions were generated for each of the 500 names in the vocabulary and added to a pronunciation lexicon. Using optimized language-dependent transcription probabilities the authors achieved a considerable relative reduction in name error rate (40% for native Dutch speakers, 45% for English speakers and 70% for French speakers). Réveil *et al.* [82] achieved similar results when including English and French g2p transcriptions to a native Dutch g2p lexicon. This lexicon was then evaluated on three different test sets comprising English and French name utterances spoken by; native Dutch speakers, English and French speakers and finally Turkish and Moroccan speakers. All three test sets achieved a substantial gain of over 20% with respect to a baseline system containing only Dutch g2p transcriptions.

A similar approach, using eight g2p converters trained on eight different languages and a language identification scheme, was employed by Maison *et al.* in [83]. In this study, alternative pronunciations were generated for all the names in the vocabulary using these g2p converters. A language identification scheme was then used on each of these names to find the two languages most likely to be the source of the name. Then, the two most probable g2p transcriptions generated by the g2p converters of these two languages were added to two different baseline lexicons; one containing manually generated transcriptions for each name and one containing native US English g2p transcriptions for each name. The method was evaluated in a large vocabulary setting (44k names) on three different test sets; US names spoken by US speakers, foreign names spoken by US speakers and foreign names spoken by native speakers of the language in question. For the first test set the reduction in sentence error rate was not significant. For the second and the third test set, however, this method resulted in a 10% and 25% absolute reduction in the sentence error rate respectively.

Although g2p converters contain valuable linguistic knowledge for most words, they naturally perform rather poorly on word pronunciations that deviate from what conventional pronunciation rules would predict. Much effort has therefore been invested in adapting the g2p transcriptions to handle category-specific pronunciation variation, such as that seen in proper names. One example of this is the grapheme-to-phoneme (g2p) phoneme-to-phoneme (p2p) tandem proposed by Yang *et al.* [84]. This approach aims to correct the mistakes made by the g2p converter by using a category specific phoneme-to-phoneme converter trained on target transcriptions of actual proper name pronunciations. The p2p converter is trained using a four step training procedure. First, the target transcriptions are automatically aligned with the corresponding g2p transcription and the orthographic transcription of the name. In the second step, phone transformations accounting for the systematic errors made by the g2p converter are derived from the discrepancies observed in the alignment. These transformations are then used in the third step to generate a set of training samples from which general pronunciation rules are learned. Finally, pronunciation rules are induced from these training examples. In the variant generation phase, the p2p converter takes the initial g2p transcription and the orthographic transcription as input and aligns these transcriptions. This alignment is then searched for a phonemic or orthographic context in which one of the learned rules applies. If such a context is found, the corresponding stochastic rule is applied and the converter continues searching the alignment for other context for which a rule can be applied. Preliminary experiments using three p2p converters trained on first names, last names and geographical names showed significant improvements in both word error rate and the number of improved erroneous initial transcriptions. Using the same g2p-p2p tandem on a set of names of foreign origin, van den Heuvel *et al.* [85] were able to automatically generate pronunciation variants for Dutch, English, French and Moroccan proper names which yielded a better performance than the initial g2p variants.

In an effort to adapt the grapheme-to-phoneme conversion of names in a name recognition system, Li *et al.* [58] employed a set of acoustic name samples. In this work, the original g2p converter was based on a N-gram *graphoneme* model, where a graphoneme unit was defined to be a pattern of graphemes connected with the corresponding pattern of phonemes. Acoustic likelihood scores calculated by a speech recognizer when decoding the adaptation data was then used to adopt the original n-gram parameters. Two different training strategies were employed in this study: a maximum likelihood approach (ML) and a discriminative approach (DT). In the maximum likelihood approach, the most likely phoneme sequence was found by maximizing the likelihood that the phoneme sequence was generated from the given grapheme sequence and acoustic sample. This phoneme sequence was then used together with the corresponding grapheme sequence to re-estimate the graphoneme model. This procedure was repeated until convergence. In the discriminative training approach, the goal was to model the graphoneme parameters in such way that the variants generated using the updated g2p converter minimized the actual error rate. To that end, the conditional likelihood of the grapheme sequence given the acoustic samples was maximized. When tested in a large vocabulary setting (58k names) using a test set containing 2844 name utterances (first name, last name), the ML training approach resulted in a 7% relative reduction in sentence error rate (SER), whereas the DT training approach yielded a 12% reduction in SER.

3.4 Modeling pronunciation variation at the language model level

As described above, pronunciation variation is often handled at the lexical level by adding multiple pronunciation variants to the recognition lexicon. After adding these variants in the lexicon the question becomes how to handle these variations in the language model. Strik and Cucchiarini [86] describe three ways of handling this in their survey on pronunciation variation modeling. The first approach is to do nothing and leave the language models unchanged. This entails using the same word probability for all pronunciations of that word. The second approach is to treat the variants themselves as words and use them directly to calculate new N-gram scores, an approach employed by Kessens *et al.* [87], among others. The third approach is to introduce pronunciation variant probabilities in addition to word N-gram probabilities in the decoding process.

A common strategy to calculate this probability is to give word pronunciations that occur frequently in some training set a higher probability than pronunciations that occur infrequently. The rationale behind this is that words that occur frequently are more important for the overall recognition performance than infrequent words, but also that frequent words often contain more variation [73]. Probabilities for rule-generated pronunciation variants can easily be obtained by utilizing the probabilities of the rules involved in generating the variant, as was done by Cremelie and Martens [64]. An alternative approach, optimizing the pronunciation prior probabilities in respect to the word error rate, rather than estimating priors by relative pronunciation frequencies observed in training data, was proposed by Schramm and Beyerlein [88]. In this approach the pronunciation priors were optimized discriminatively using the Discriminative Model Combination (DMC) framework. Incorporating these pronunciation weights into the lexicon gave a relative error rate reduction of 7.9%.

3.5 Modeling pronunciation variation at the acoustic level

Most state-of-the-art ASR systems today employ phones and context-dependent phones as basic recognition units. In these systems pronunciation variation is usually modeled by introducing additional pronunciation variants in the lexicon. These additional variants are commonly generated by applying some form of substitution, insertion or deletion rules to a baseform pronunciation. This pronunciation modeling strategy is often referred to as *explicit* modeling. An alternative pronunciation modeling approach is to model the variation *implicitly*, within the acoustic model. In this section we will give a brief overview of the most common implicit modeling approaches.

3.5.1 Modeling pronunciation variation implicitly

An implicit pronunciation variation modeling approach was proposed by Hain in [89]. In this study, Hain argued that by stepwise reducing the number of pronunciation variants per word to one representative variant, it is possible to model most types of pronunciation variation within the acoustic model. A lexicon compiled in this way, containing one single pronunciation per word, was tested on read and conversational speech and the results were compared to that of the full lexicon, containing multiple pronunciation variants per word.[1] The results showed that the proposed lexicon performed equally well or better than the lexicon using multiple pronunciations per word, given that the starting lexicon is of good quality.

It is often argued, however, that modeling pronunciation variation at the phonemic level is suboptimal due to the fact that pronunciation variation often is a result of coarticulation and assimilation effects between different phones. Sethy *et al.* argued in [90] that using syllables as the basic acoustic unit can reduce the need for multiple pronunciation variants, as pronunciation variation observed across multiple phones can be modeled within the acoustic unit. This study compared a system using context-dependent phones with a system constructed using syllables for the task of English proper name recognition. The authors argued that modeling larger contexts within the acoustic model could reduce the need of having multiple pronunciation variants of each word. A performance analysis comparing the two systems showed that increasing the number of names in the vocabulary had a much smaller effect on the syllable-based system compared to the system using context-dependent phones. In fact, the study reported a substantial increase in name recognition accuracy, on a large vocabulary task (10k names), when using syllables instead of context-dependent phones as the basic recognition unit. The downside of using longer recognition units, however, is that the number of basic models increases with the vocabulary size, which can often lead to a low coverage in the training data, especially for larger vocabularies. For the particular case of proper name recognition, some languages have the advantage that the number of names actually used is relatively small. In Korean, for example, the 50 most frequent surnames covers 95% of the population. For these languages, whole word models can have a significant effect on the error rate, as was shown in a study by Kim *et al.* [91]. This is unfeasible for most languages, however, as the number of commonly used proper names tends to be tens of thousands.

Recent studies (e.g. [54], [53], [92], [93]) have looked into representing word pronunciations as a sequence of underlying articulatory feature streams rather than as a sequence of phones. The argument behind this is to avoid the notion that words are realized as “beads-on-a-string”, i.e. as a sequence of non-overlapping phones and that pronunciation variation is just a series of phoneme-level insertions, deletions and substitutions [94]. Saraçlar argues in [95] that some types of pronunciation variation are the result of more continuous and gradual changes rather than just a simple replacement of one phone with another. Some studies further argue that pronuncia-

[1] Context-dependent phones were used in this study.

tion variation can be better modeled by using articulatory features which are a more fine-grained representation of a word pronunciation. Livescu and Glass describe in [54] a feature-based pronunciation model where each baseform in the lexicon is represented by multiple streams of underlying feature values. These context-independent features were described as linguistic properties such as degree of lip opening and lip rounding, the location of the tongue tip along the palate. Dynamic Bayesian networks (DBN) were then used to model the relationship between these linguistic features and words. In this pronunciation model, pronunciation variation can be seen as a result of an asynchrony between different features, e.g. the situation where one feature transitions to the next state before all the remaining features. Pronunciation effects such as vowel nasalization, for example, is typically a result of the velum feature changing state before the other articulatory features. In experiments recently performed by Bowman and Livescu [53] and Jyothi *et al.* [93], the context-independent features of [54] were replaced with context-dependent features, modeled using decision trees. In the latter of these studies, context-dependent models based on articulatory features were evaluated on a lexical access task and compared with the performance of context-dependent phone models on the same task. The result of this experiment showed that the feature-based models performed significantly better than the phone-based models. Furthermore, with the inclusion of context, the best feature model generally ranks the correct word hypothesis higher in the hypothesis list compared to the baseline phone-based models.

3.5.2 Iterative adaptation of existing models

An acoustic adaptation technique that is often used in conjunction with a lexical pronunciation variation modeling scheme, is the iterative re-training of existing acoustic models (e.g. [96],[55], [66], [97], [98]). In this approach, new, and hopefully improved, pronunciation variants from the lexical modeling scheme are used to re-train the acoustic models used a previous iteration. The rationale behind this is that by using these improved variants, a better match between the basic acoustic units and the speech signal can be achieved, which will most likely lead to more accurate acoustic models. In the next iteration, the new set of acoustic models are used in the lexical modeling scheme to generate new and improved pronunciation variants, which are again used to re-train the acoustic models. This procedure is iterated until no further gain is observed. The success of this scheme has been variable. Some studies have reported increased performance ([96],[55], [66]), whereas other studies such as [97] and [98] observed only a limited or no improvement at all.

3.5.3 Modeling non-native speech

As discussed previously in this chapter, non-native speakers tend to use non-native sounds to a varying degree. Speakers with a high proficiency in the target language may have a near-perfect realization of non-native sounds whereas other speakers might have a heavily “nativized” realization of the same sounds. Several methods for handling non-native speech have been proposed in the automatic speech recognition literature. Perhaps the most intuitive approach has been simply to train the acoustic models using relevant non-native speech. Réveil *et al.* showed in their comparative study [82] that substantial gains could be achieved by using acoustic models trained on multilingual speech data. In this study, the authors compared the performance of standard monolingual acoustic models models trained on native (Dutch) speech and state-of-the-art multilingual acoustic models trained on Dutch, UK English, French and German speech. The underlying phoneme set of the monolingual models consisted of 45 phonemes, while 80 phonemes were used in the multilingual case. These models were tested on the recognition of Dutch, English, French, Moroccan and Turkish proper names spoken by native speakers of the same five languages. The experiments showed substantial performance gains for all non-native names spoken by all speakers. These gains were, as could be expected, at the expense of the recognition performance of native names uttered by native speakers.

In many cases, however, training multilingual models is not an option since non-native speech is rarely available in the quantities necessary to train accurate acoustic models. Another approach successfully employed by a number of studies (e.g. [35], [99] and [100]), is to model the acoustic variation in non-native speech by applying well-known acoustic adaptation schemes such as the MLLR approach and the MAP approach.

A comparative study of different acoustic adaptation techniques was performed by Wang *et al.* [99] for the task of recognizing English words spoken by German native speakers. In the first approach described in this study, acoustic models were trained using 34 hours of native English speech pooled together with 52 minutes of German speech. This approach resulted in a slight performance gain of 0.8% absolute compared to the baseline English models (WER 43.5%). The authors argued that this result most likely was an effect of the moderate amount of non-native training data compared to the extensive amount of native training data. In the second approach, native English models were adapted using a MAP adaptation technique and 52 minutes of non-native speech. This resulted in a performance gain of 6% absolute compared to using the native English models. The third approach explored an interpolation technique using the weighted average of the probability density functions of the native and the non-native acoustic models. Using this technique the authors achieved a considerable gain of 7.5% absolute. In the final approach, a Polyphone Decision Tree Specialization method [101] was adopted to port the native decision tree to the non-native language to represent the context of the non-native speech more accurately. Preliminary experiments using this approach showed a performance gain of 8% absolute compared to using the baseline English models.

A different, well proven approach to model the variation observed in non-native sounds, is to extend the native phone set with phones specific to the non-native language. Stemmer *et. al* [102] used such an approach to model native German speakers uttering English movie titles. In this study, phones shared by the non-native language (English) and the native language (German) were trained using both English and German speech data. Phones unique to only one of the two languages were trained using only language specific data. This approach improved the recognition accuracy by 16.5% absolute. Extending the phone set, however, may in some cases reduce the discriminative power of the models and will augment the number of pronunciation variants needed in the lexicon, which will in turn increase the confusability between lexicon entries. A common way to avoid expanding the phone set is to map all foreign phonemes to the best native phone equivalent. The downside of this approach is that foreign phonemes may have quite different phonological characteristics than their native equivalent, which means that these phonemes may be inaccurately represented by the native acoustic model. Stouten and Martens [103] and [104] aimed to avoid this by introducing the concept of “foreignizable” phonemes, which effectively are native phonemes attached to a foreign phoneme. In this study, the phonemes were modeled acoustically by combining scores from a standard acoustic model with scores from a phonological inspired back-off acoustic model which was trained on native speech. Since these scores also were trained purely on native data, this method entirely eliminates the need for non-native training data. In a small-scale test experiment, this approach yielded a relative improvement of 11% compared to using purely native acoustic models.

Gao *et al.* [35] showed that speaker clustering can be effective to improve recognition accuracy of a speaker independent telephone-based name dialing system. By clustering speakers with similar acoustic characteristics, speaker dependent characteristics as well as channel and noise conditions, the authors managed to reduce both the word error rate and the sentence error rate in a preliminary large vocabulary experiment.

3.6 Novel discriminative approaches to variant selection in lexical modeling

Proper names pose a severe challenge to the ASR engine and the reasons for this are manifold, as discussed in the beginning of this chapter. One successful solution to this problem has been to include a set of alternative pronunciation variants in the recognition lexicon. However, this entails the risk of introducing unwanted confusion between lexicon entries. In the work described in this dissertation, we aim to increase the recognition performance of non-native proper names by optimizing the lexicon in such a way that the lexical confusion is minimized. To achieve this, several novel variant selection approaches will be proposed. This section gives a general outline of the basic steps common to all the proposed approaches as well as an introduction to some general concepts and frameworks relevant to the interpretation of the proposed algorithms.

3.6.1 Main steps of the proposed variant selection approaches

We assume that we have a set of names $\mathcal{W} = \{W_1, \ldots, W_K\}$, and that for any name $W_k \in \mathcal{W}$ we have a set of training utterances $\mathcal{X}_k = \{X_{k1}, \ldots, X_{kN}\}$, a grapheme-to-phoneme transcription G_k and a set of auditorily verified transcriptions $\mathcal{T}_k = \{T_{k1}, \ldots, T_{kN}\}$ of the utterances in $\mathcal{X}_k$.[2] Using these auditorily verified transcriptions and the grapheme-to-phoneme transcription G_k, we generate a set of pronunciation candidates, $\mathcal{V}_k = \{V_{k1}, \ldots, V_{kI}\}$. An initial lexicon Λ_c, comprising one general-purpose g2p variant for each name in $\mathcal{W}$, is then constructed. By replacing the variant corresponding to name W_k in Λ_c with variant V_{ki}, we form the temporary lexicon Λ_{ki}. To obtain some evidence of the performance of this temporary lexicon, a recognition pass is executed using Λ_{ki} and the training utterances $X_{kn} \in \mathcal{X}_k$. This procedure is then repeated for all variants $V_{ki} \in \mathcal{V}$. The goal, then, is to find the optimal set of pronunciation variants for name W_k based on the evidence observed in the training material. The five main steps of the proposed variant selection approaches are illustrated in Figure 3.1. In some of the later selection approaches proposed in this thesis (specifically in Chapter 6 and Chapter 7), the optimized lexicon is used in an iterative manner as input to the selection algorithm. This iterative behavior is indicated by the dashed line in Figure 3.1.

In the work described in this dissertation, we will mainly focus on the

[2] These transcriptions are manually created transliterations of what a human expert actually heard when listening to the individual training utterances.

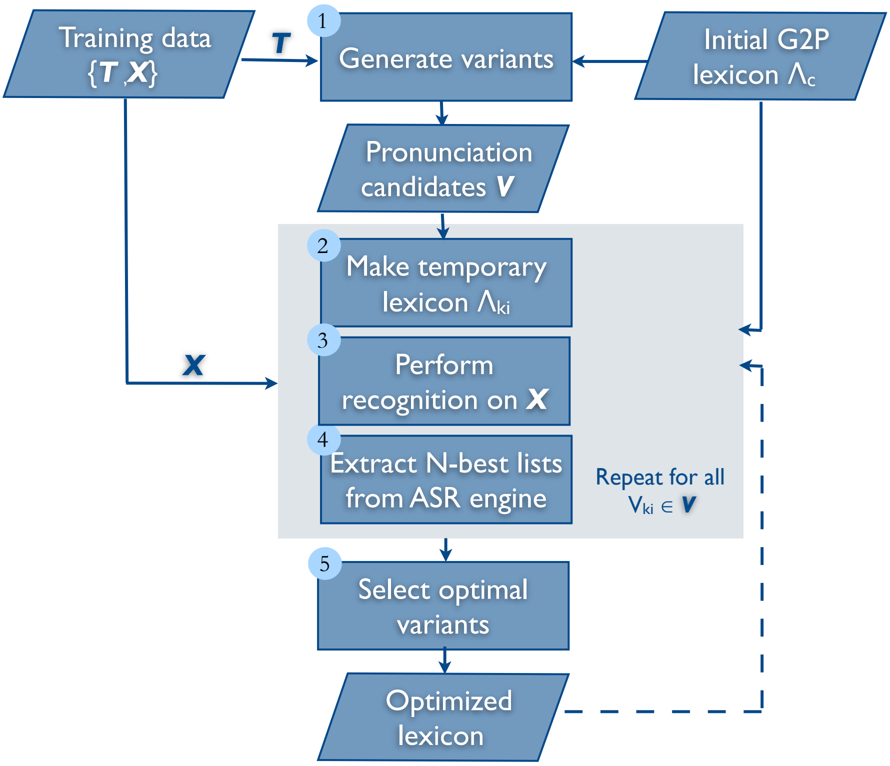

Figure 3.1: *Basic steps of the proposed pronunciation variation modeling approaches.*

variant selection phase (step 5) of Figure 3.1. We will do this by considering the variant selection problem as a decision problem, where the most promising variant V_k^* can be found by maximizing a decision rule $S(\mathcal{X}_k, \Lambda_{ki})$

$$V_k^* = \operatorname*{argmax}_{V_{ki} \in \mathcal{V}_k} S(\mathcal{X}_k, \Lambda_{ki}). \tag{3.1}$$

Since we aim to optimize recognition lexicons by selecting a minimal amount of maximally effective variants, the decision rule S should be designed in such a way that the resulting recognition lexicon contains a good balance between phonetic coverage and lexical confusability. To achieve this, the decision rule must be composed of a *variant selection criterion* that reflects a variant's potential to correct recognition errors. To that end, several criteria have been proposed in the literature, as discussed in Section 3.3.2. In the remainder of this section, we will introduce three such variant selection criteria which will form the basis of the variant selection approaches proposed in this dissertation.

3.6.2 A Maximum Likelihood variant selection criterion

One conventional way of assessing the performance of a speech recognition system is to utilize the acoustic log likelihood scores calculated by the recognition engine. This score is the likelihood of a path through the recognition network when recognizing utterance X_{kn}, calculated using probabilities stored in the HMM model and in the language model. The likelihoods scores for the most likely paths (name hypotheses in our case) are then given in the N-best list, H_{kn}, by the recognizer. If name W_k is in this list, the log likelihood score of the name can be directly extracted from H_{kn}. If it is not in this list, the name is given a fixed score much lower than the score of the least likely hypothesis in H_{kn}. The *total log likelihood* score (LLH) for name W_k, is then defined as the sum of the log likelihood scores calculated for all the training utterances $X_{kn} \in \mathcal{X}_k$ of name W_k:

$$LLH(\mathcal{X}_k, \Lambda_{ki}) = \sum_{n=1}^{N} LLH(X_{kn}, \Lambda_{ki}). \tag{3.2}$$

Since the log likelihood score gives an indication of the expected recognition performance of a variant, it seems like a good candidate for our variant selection criterion. By prioritizing the selection of the variants with the maximum likelihood scores, we may achieve a considerable level of lexicon optimization.

3.6.3 A discriminative Maximum Entropy (ME) variant selection criterion

A discriminative way of assessing the performance of a pronunciation variant $V_{ki} \in \mathcal{V}_k$, is to model the probability of an utterance of name W_k being correctly recognized by the recognition engine when using lexicon Λ_{ki} (comprising variant V_{ki}). A well-suited framework for modeling this probability, is the Maximum Entropy framework described in Section 2.3.1. As described in this section, the Maximum Entropy principle states that we should choose the most uniform distribution that satisfies a set of constraints. Applying this principle to the pronunciation variation modeling task, the problem can be defined as finding the probability distribution $P(\Lambda_{ki}|c_k)$ of lexicon Λ_{ki} that maximizes the entropy under a set of constraints c_k. In this work we will use one single constraint, namely whether or not an utterance $x_k \in \mathcal{X}_k$ is recognized correctly when the lexicon Λ_{ki} is employed. Using this constraint, the Maximum Entropy model, $P(\Lambda_{ki}|W_k)$, for a particular lexicon will estimate the probability that the lexicon resulted in a correct classification given name W_k. This probability is calculated using the number of times a lexicon resulted in a correct classification in the training material

$$\hat{P}(\Lambda_{ki}|W_k) = P(\Lambda_{ki}|c_k) = P(\Lambda_{ki}|x_k \text{ is recognized correctly})$$

To train the ME model, $P(\Lambda_{ki}|c_k)$, we define a set of I binary features, one for every pronunciation variant. These features represent the constraint described above and can be defined as

$$f_j(c_k, \Lambda_{ki}) = \begin{cases} 1 & \Lambda_{kj} = \Lambda_{ki} \text{ and } c_k \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where the feature of variant V_{kj} is given a value of 1 if the variant V_{ki} under evaluation is equal to variant V_{kj} and if this variant resulted in a correct classification of utterance x_k. In this way, we also constrain the ME model to model c_k with the same frequency as was observed in the training data.

The pronunciation prior distribution that satisfies the feature constraints while maximizing the entropy is then given by

$$P(\Lambda_{ki}|c_k) = \frac{1}{Z(c_k)} \exp\left[\sum_{j=1}^{F} \lambda_j f_j(c_k, \Lambda_{ki})\right] \tag{3.4}$$

where $f_j(c_k, \Lambda_{ki})$ are the feature constraints, F is the number of features in the set (equal to I in our case), λ_j are the feature weights and $Z(c_k)$ is a normalization factor $Z(c_k) = \sum_{i=1}^{I} \exp[\sum_{j=1}^{F} \lambda_j f_j(c_k, \Lambda_{ki})]$. The feature weights λ_j are found by employing the Improved Iterative Scaling training algorithm [21].

After training this parametric model for every name W_k using the constraints c_k, the probability of every variant $V_{ki} \in \mathcal{V}_k$ can be extracted from the model. The probability of variant V_{ki} is then directly correlated with the number of correctly recognized name utterances in the training set when lexicon Λ_{ki} is employed. Selecting the variants with the *highest* probability for lexicon inclusion, is thus the same as selecting the variants proven to correct many recognition errors in the training set.

3.6.4 A discriminative Minimum Classification Error (MCE) variant selection criterion

Another discriminative way of assessing the performance of a pronunciation variant V_{ki} is to calculate the *expected loss of recognition accuracy* observed for a set of training utterances $\mathcal{X}_k$ when using a lexicon Λ_{ki} comprising variant V_{ki}. There are several ways of defining this loss, one of which is to use the expected loss function defined in the Minimum Classification Error framework described in Chapter 2.3.2.

Using this function to model the performance of lexicon Λ_{ki}, we get the following expected loss function

$$\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki}) = \frac{1}{N} \sum_{n=1}^{N} l_k(X_{kn}; \Lambda_{ki}). \tag{3.5}$$

The loss function, $l_k(X_{kn}; \Lambda_{ki})$, can be defined by means of a set of discriminant functions $g_l(X; \Lambda_{ki})$ with $l = 1, \ldots, K$. Most modern speech recognition applications rely on log likelihood scores to make a decision, and consequently, these scores will act as discriminant functions in this work. If H_{kn} is the set of likely name hypotheses proposed by the recognizer for utterance X_{kn} when employing lexicon Λ_{ki}, one can define a *misclassification measure* $d_k(X_{kn}; \Lambda_{ki})$ as

$$d_k(X_{kn}; \Lambda_{ki}) = -g_k(X_{kn}; \Lambda_{ki}) + \log \left[\frac{\sum_{j,j \neq k} e^{g_j(X_{kn};\Lambda_{ki})\eta}}{\mathrm{card}(H_{kn}) - 1} \right]^{\frac{1}{\eta}} \tag{3.6}$$

where η is a positive number. In order to map the misclassification measure of Equation (3.6) to a zero-to-one continuum, the loss function is defined as

$$l_k(X_{kn}; \Lambda_{ki}) = \frac{1}{1 + e^{-d_k(X_{kn};\Lambda_{ki})}}. \tag{3.7}$$

If the loss is close to zero, it means that the utterance is likely to be correctly recognized using lexicon Λ_{ki}. The larger the measure is, the larger the risk for an incorrect recognition of the utterance.

The expected loss function $\mathcal{L}_k$ can then be used to extract the variants that have, on average, the largest difference between the log likelihood score of the correct hypothesis and the log likelihood of the competing hypotheses, i.e. the variants that have the smallest risk of being misrecognized. Thus, selecting the variants with the *lowest* expected loss, is equivalent to selecting the variants proven to have the best recognition performance on the training set and posing the smallest risk of causing a misrecognition.

Chapter 4

Experimental set-up and baseline experiments

To improve the recognition performance of proper names by modeling the pronunciation of the names the lexicon, several language-specific resources are required. Firstly, it is crucial to have access to a corpus that contains the type of variation to be modeled. Secondly, since it is our goal to investigate different methods to select optimal pronunciation variants, we must have a pool of candidate variants from which to select. Finally, a recognition system is needed to evaluate the performance of the candidate variants and of the final lexicons.

In the first section of this chapter, the creation of these three language-specific resources is described in detail. The second section describes the experimental set-up adopted in all the experiments conducted in this dissertation. The final section of this chapter presents the results of initial threshold experiments and two baseline experiments conducted using the resources described in this chapter.

4.1 The NameDat corpus

When we started to work on this dissertation, the only available resource concerning Norwegian pronunciation of non-native names was the Onomastica [80] corpus. Unfortunately, this is a purely lexical resource that includes only a single "nativized" transcription for each name in the lexicon, and no recorded speech. These limitations of the Onomastica corpus make it unusable for either lexical or acoustic pronunciation variation modeling. It was therefore deemed necessary to collect a new resource for Norwegian containing annotated speech utterances of non-native names.

This section will describe the design, recording and annotation of the NameDat database, a small-scale database containing English proper names spoken by native Norwegians. The database was designed as an additional resource to the large vocabulary speech recognition engine SVoG[1] and its main purpose was to reveal which typical phonetic patterns appear when native Norwegians pronounce English proper names. For additional details on the collection of this corpus, the reader is referred to [1].

4.1.1 Corpus design

The speech data presented in the NameDat corpus was collected from 33 native Norwegian speakers of between 18 and 60 years of age. The speakers were recruited among colleagues, friends and family. The quality of the recordings is highly dependent on the speakers and their ability and experience with reading aloud. An effort was made to cover some distribution in terms of speaker gender, education and age. As for the parameter of provenance, for such a limited amount of speakers it was unfortunately unfeasible to cover the numerous dialectal regions in Norway. The last design parameter presented in the database is language proficiency, which was determined by means of a self-assessment poll of the speakers. Table 4.1 gives an overview of the speakers following these parameters.

Criterion	Speakers	
Age	Over 40	Under 40
	12	21
Gender	Male	Female
	17	16
Higher education	Yes	No
	26	7
English proficiency	Intermediate	Good
	10	9
	Very good	Fluent
	11	3

Table 4.1: *Speaker distribution of the NameDat corpus.*

Each of the 33 speakers read a manuscript consisting of 125 sentences where each sentence contained one, two or three names of English origin.

[1] `http://www.sintef.org/Home/Information-and-Communication-Technology-ICT/Acoustics/Communication-acoustics/ Speech-technology/`

There were five different manuscripts in the corpus, each containing 221 names, yielding a total of 1105 unique names. The first four manuscripts were read by seven speakers, while the fifth was read by the remaining five speakers. The manuscripts contained mostly place names from English speaking areas and a smaller amount of common US and UK person names.

Three features were especially emphasized in the corpus design. Firstly, it was deemed desirable that the corpus contained both well-known names and names unknown to the speaker. In order to achieve this, two selection criteria were applied, viz. the name's frequency of occurrence in a large text corpus from the news domain, and in the case of city names, the city's number of inhabitants. These criteria were taken as a rough indication of the familiarity of the names through the media and travel. The second feature was to have a considerable amount of "difficult" names in the corpus, where a "difficult" name was intended to be a name that a general automatic speech recognizer would have trouble classifying correctly. The Levenshtein distance between a transcription generated automatically by an English grapheme-to-phoneme converter and a transcription made by a human expert was used to identify these names. Finally, the third desirable feature was to have a good coverage of non-native sounds in the corpus. Therefore, a special effort was made to include names that feature English phonemes in their pronunciation which are not part of the native Norwegian phoneme alphabet. As such, these particular names supply a good coverage of English sounds that typically have a large pronunciation variation when uttered by Norwegian speakers.

4.1.2 Recordings

Due to logistic reasons, the recordings were made in two different acoustic environments. The recordings with the majority of the speakers were made in a soundproof acoustic laboratory, while the recordings of the other speakers were made in an office environment.[2] Prior to the recording session the speakers were briefed about the purpose of the project and what was expected of them. They were informed that they would be asked to read 125 Norwegian sentences, all of which contained at least one English name. They were explained that the purpose was not to record the "correct" English pronunciations, but rather to record how they would actually pronounce the names in everyday speech. They were instructed to try to pronounce all names, even if they had no idea how to pronounce them.

[2] The choice was made out of necessity: 14 speakers were located in the Oslo area, where we had no acoustic laboratory at our disposal.

The recording script was presented to the speakers using the audio recording software Speechrecorder[3]. In order to avoid hesitations, the speakers were instructed to read through the sentence presented on the screen and decide how to pronounce the names in the sentence prior to making a recording.

The recording chain consisted of a Sennheiser HMD 25-1 dynamic headset microphone and Shure FP23 microphone amplifier connected to the line-in port on a MacBook Pro. The signal-to-noise ratio of the recording chain was measured to be 51 dB and the frequency response of the chain was measured and found to be reasonably flat.

The MacBook Pro was used to digitize the speech. For all recordings a sample rate of 48kHz was used and the samples were stored in 16-bit linear PCM wav format.

4.1.3 Broad phonetic annotation

The purpose of the broad phonetic annotation was to document all the different name pronunciations perceived in the recordings and to detect common linguistic features in English proper names spoken by Norwegians. The annotations were later to be used in pronunciation modeling of English names, so consistency and accuracy were naturally essential qualities in the annotation process. Due to budgetary reasons, names from 125 sentences were manually annotated for 19 out of the 33 speakers.

Annotation format and tools

The phoneme set used for the annotations was in the SAMPA[4] format, with the Norwegian phoneme inventory as the core set. In order to represent phonemes occurring in English names and loan words which lack an equivalent in the Norwegian inventory, the Norwegian phoneme set was extended with symbols from the British English SAMPA phoneme inventory. These phonemes are listed in Table 4.2.

The annotations were made in Praat[5] and consisted of five tiers: *auto*, *phone*, *phone comment*, *word*, and *utterance*. Provisional annotations were available for the whole sentence. For the carrier sentence, the annotations were automatically generated using a Norwegian Text-to-Speech front-end, and for most of the English names, expert transcriptions were available from

[3] `http://www.phonetik.uni-muenchen.de/Bas/software/speechrecorder/`

[4] `http://www.phon.ucl.ac.uk/home/sampa`

[5] Version 5.0.46, available at `http://www.praat.org/`

Symbol	Example word
eI	r**ai**se
aU	r**ou**se
@U	n**o**se
r	**wr**ong
w	**w**asp
z	**z**ing
Z	mea**s**ure
D	**th**is
T	**th**in
tS	**ch**in
dZ	**g**in

Table 4.2: *Non-native phoneme extensions.*

the Onomastica Consortium [80] and an in-house pronunciation dictionary. The alignments were obtained using forced alignment.

The provisional annotations were presented to the annotator in the *auto* tier and the corrections were made in the *phone* tier. The annotation was mainly phonetic, but boundaries were corrected where the alignments were clearly misplaced in the provisional annotation. Only the names and name boundaries were corrected. The *phone comment* tier was aligned with the *phone* tier and was used to comment on frequently occurring variations[6].

In the *word* tier, names could be marked as unusable or as mispronunciations. A name was marked as unusable if it contained long pauses or was corrupted by background noise. A name was marked as a mispronunciation if the realization of the name was clearly a reading mistake. For instance, pronouncing the name *'Gilmilnscroft'* as *'Gilmilnsoft'* is obviously a misreading and would be marked as a mispronunciation. However, articulation errors and errors made due to the speaker's insufficient knowledge of English were not marked as mispronunciations. A log file and a pre-defined set of tags were available to the annotator to comment on any uncertainties. The log file can easily be queried by means of the tags.

[6] Typical comments were e.g. devoicing of voiced phone, uncertain phone identity, phone is realized as an approximant, missing or unknown phone, typical "nativized" pronunciation of an English phone.

Annotation procedure

For consistency reasons, the annotations were performed by one single expert annotator. The annotator was given a set of guidelines and a test session was performed where the annotator received feedback on his annotations. The annotator was instructed to check the provisional transcriptions of the names in the carrier sentences and modify them if necessary. Corrections were made according to general guidelines for Norwegian annotation. In addition, the annotator was instructed to pay special attention to non-native sounds and decide whether or not they were pronounced in a "nativized" manner.

4.2 Experimental set-up

In this section, the experimental set-up used throughout this dissertation is described in detail. The section is divided in three parts: the first part describes the data set extracted from the NameDat corpus, the second part describes the generation of transcription variants and the third part describes the employed recognition engine.

4.2.1 Data set

For our experimental study, we extracted annotated name utterances from 19 of the 33 speakers in the NameDat corpus. Name utterances from the remaining 14 speakers were not used in our study. The choice of speakers was made out of necessity as only the names spoken by 19 of the speakers in the corpus were manually annotated. The name utterances were obtained from three different manuscripts, each containing 221 unique names. The two first manuscripts were read by seven speakers, while the third manuscript was read by five speakers. For each speaker, 16 sentences were withheld from the data set and used to form an adaptation set. Removing the name utterances used in the adaptation set and name utterances deemed unusable by the annotator yielded the final data set illustrated in Table 4.3.

4.2.2 The recognition engine

The recognition engine used in this dissertation was the large vocabulary continuous speech recognizer SVoG which is based on the Hidden Markov Toolkit (HTK) [10]. This engine employs word-internal tri-state, left-right triphone models without skips where each state uses 2-64 Gaussian mixture components. Additionally, context-independent models with matching

Manuscript	Names	Speakers	Name utterances
Manuscript 1	206	7	1436
Manuscript 2	202	7	1400
Manuscript 3	209	5	1039
Total	617	19	3875

Table 4.3: *The final data set.*

topology and a three state silence model (with feedback from the third to the first state) are available to the recognition engine. The monophone models use 32 Gaussian mixture components for each state. The feature vectors used by the SVoG recognizer are traditional MFFCs with 13 cepstral coefficients (including C_0) plus the corresponding delta and acceleration coefficients. When extracting these features, the frame length was set to 25ms and the frame shift was 10ms.

To compensate for acoustic dissimilarities from the recording environment and to "tune" the acoustic models to the NameDat speakers, speaker adaptation was performed on the model parameters. For this purpose, a standard Maximum likelihood linear regression (MLLR) algorithm was applied to the mean values of the gaussian mixture components. The MLLR algorithm aims to adapt the model parameters by applying a parametric transformation to the parameters. In this work, instead of applying the same transform to all the mixture component mean values, a regression class tree was used to apply different transforms to different parts of the model. This enables the adaptation algorithm to dynamically specify the number of transformations to be generated depending on the amount of available adaptation data. The regression class tree was generated using 32 leaf nodes and an unsupervised clustering technique incorporated in the HTK tool HHEd. The transform was then estimated in a maximum likelihood sense using this regression tree and the withheld adaptation data.

Two different grammars were used in the experiments described in this thesis. For our controlled environment experiments, a small vocabulary grammar containing a loop of the 617 names comprised in the NameDat corpus was used. For our open environment experiments, a large vocabulary grammar containing a loop of the same 617 names plus 15,428 other English proper names was used. These 15,428 names were simply "filler names" included to extend the vocabulary size.

4.2.3 Three-fold cross validation procedure

Due to the limited size of our data set, a three-fold cross validation strategy was employed for all the experiments in this dissertation, rather than the more commonly used five- or ten-fold cross validation. This strategy entailed dividing the full data set in a test set and a training set three times so that the three test sets each comprised utterances by different speakers. Since we plan to use discriminative measures for variant selection, which can only be calculated for names for which we have training utterances, we can only assess the positive effect of the selected variants on the recognition of these names if the test set also contains utterances of these names. Therefore, the full data set was divided so that each test set comprised one third of the utterances of each unique name and the corresponding training set comprised the remaining utterances. The division was made in such a way that there was also no overlap in speakers between a test set and the corresponding training set. Table 4.4 shows the number of unique names, the average number of utterances per name and the total number of name utterances for each test and training set. Since the first two manuscripts were read by seven speakers while the remaining manuscript was only read by five speakers, the average number of utterances per name varies somewhat between the three test and training sets.

Data set	Names	Avg. utterances per name	Name utterances
Test set 1	617	2.32	1430
Train set 1	617	3.96	2445
Total	617	6.28	3875
Test set 2	617	2.01	1239
Train set 2	617	4.27	2636
Total	617	6.28	3875
Test set 3	617	1.95	1206
Train set 3	617	4.33	2669
Total	617	6.28	3875

Table 4.4: *Test and training sets used in the three-fold cross validation scheme.*

Each training procedure was then performed three times, once for every training set, and the result of every procedure was tested using the corresponding test set. The average of the three test results was taken as the final result. This holds for all experimental results presented in this dissertation: the numbers should always be understood as averages over three folds.

4.2.4 Transcription variants

Most current large vocabulary speech recognition systems rely on automatically generated transcriptions to model the pronunciation of proper names in the lexicon. These general purpose transcriptions are commonly generated from language specific grapheme-to-phoneme (g2p) converters. As the name indicates, a g2p converter converts a grapheme string into a sequence of phoneme symbols using a set of language-specific rules. However, since the pronunciation of proper names often deviates from what conventional pronunciation rules would predict, g2p transcriptions are normally not very accurate descriptions of the actual pronunciation. As described in Section 3.3, several efforts have been made to solve this problem by automatically generating a large set of transcriptions in order to cover a wider range of pronunciation variation. This strategy, however, has been found to introduce unwanted confusion between lexicon entries which often introduces new recognition errors. In the work described in this dissertation, we aim to address this problem by investigating several methods of identifying the optimal set of pronunciation variants from a large pool of pronunciation candidates. But before we can do so we need to create a pool of candidate transcription variants from which we can choose.

As described in Section 3.3, there are several ways to generate a pool of pronunciation variants, depending on the desired type of variants and the available data. In this work, we have two types of transcriptions at our disposal: Norwegian and English g2p transcriptions on the one hand, and manual annotations of the utterances in our data set on the other. These manual annotations are the best nativized transliterations of what a human expert actually heard when listening to the utterances. In the remainder of this dissertation, these transcriptions will be referred to as *auditorily verified (AV)* transcriptions. The English g2p transcriptions were created using the English g2p-converter embedded in the Nuance RealSpeak text-to-speech system[7], and the Norwegian g2p transcriptions were created using the Norwegian TTS engine Arne®[8]. Having these resources available to us, we decided to create an additional pool of transcription variants that should be more robust to the pronunciation variation inherent in non-native proper names. We therefore trained a set of *phoneme-to-phoneme-converters.*

[7] `http://www.nuance.com/realspeak/`

[8] `http://www.lingit.no`

Generating transcriptions using a Phoneme-to-Phoneme converter

The phoneme-to-phoneme (p2p) converter is a publicly available[9] tool developed by Qian Yang and her colleagues when she was a PhD student at Ghent University [105]. The tool was specifically developed to enhance the performance of proper name transcriptions.

The p2p converter aims to automatically correct the mistakes made by the g2p converter by learning conversion rules on the basis of auditorily verified transcriptions, g2p transcriptions, and orthographic transcriptions. The p2p converter focuses primarily on modeling pronunciation effects which are typical for proper names and is therefore not dependent on large amounts of training material. As illustrated in Figure 4.1, the p2p converter employs a two-step procedure: the g2p converter first generates an initial transcription, and the p2p converter subsequently tries to correct this initial transcription. The result of this procedure is a set of alternative transcriptions, each describing plausible variants of the initial g2p transcription. The rationale behind this approach is that the p2p converter can benefit from the knowledge of the g2p converter and can therefore attain a good performance without having access to large amounts of manually corrected transcriptions.

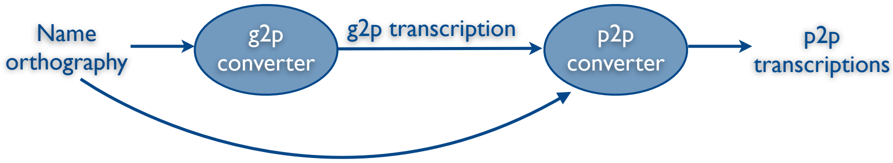

Figure 4.1: *p2p variant generation.*

When applied, the p2p converter aligns the initial g2p transcription with the orthographic transcription and examines where one of the learned conversion rules can be applied. Each conversion rule expresses the following: if a particular phonemic pattern (*rule input*) occurs in the initial transcription in a particular phonemic and orthographic context (*rule condition*), then transform the rule input to an alternative phonemic pattern (*rule output*) and assign a certain probability to the transformation. More detailed information about the p2p converter and the training of the converter can be found in [105] and [84].

For the experiments described in this thesis, three Norwegian and three English phoneme-to-phoneme converters were trained, one English and one

[9] http://www.inl.nl/en/tools/autonomata-g2p-toolkit

Norwegian converter for each training set. Conventionally, p2p converters are trained on g2p and AV transcription pairs of names not present in the test set, but since the same names occur both in our test sets and in our training sets, we were forced to train the p2p converters using names included in the test set. It should be noted, however, that only AV transcriptions of name utterances in the training set were used to train the p2p converters. This set-up is likely to produce somewhat more accurate p2p transcriptions than in the case of using unseen data exclusively. Nevertheless, since the main objective of this dissertation was not to assess the quality of these variants nor the variant creation process, this was considered to be of minor importance, although it should be kept in mind when interpreting the results. During the variant generation stage, each p2p converter was allowed to generate up to 10 variants per name, but only if their probability exceeded a threshold which was specified as a fraction (we used 0.02) of the probability of the best variant. If the input g2p transcription was not among the created variants, it was added to the candidate pool a posteriori with a probability equal to the above threshold.

To create p2p variants for the 15,428 filler names used in our open vocabulary experiments, one Norwegian and one English p2p converter was trained using g2p and AV transcription pairs for all name utterances in the three training sets. Applying these converters to Norwegian and English g2p transcriptions for the filler names yielded a set of 149,973 p2p transcriptions.

Lexicon	Train 1	Train 2	Train 3	Avg
NO g2p	617	617	617	617
EN g2p	617	617	617	617
NOEN g2p	1227	1227	1227	1227
NOEN p2p-g2p	5486	5458	5720	5555
AV	1686	1860	1870	1805
AV NOEN p2p-g2p	5917	6001	6257	6058

Table 4.5: *Number of pronunciation variants for different lexicons.*

Table 4.5 shows the lexicon size of six different lexicons containing the following variants: Norwegian g2p transcriptions (NO g2p), English g2p transcriptions (EN g2p), unique Norwegian and English g2p transcriptions (NOEN g2p) pooled together, unique p2p and g2p transcriptions generated using the p2p converters described in this section (NOEN g2p-p2p), unique AV transcriptions encountered in the three different training sets (AV) and finally non-overlapping AV transcriptions pooled together with NOEN p2p-g2p transcriptions (AV NOEN p2p-g2p).

4.3 Initial threshold and baseline experiments

In this section, preliminary experiments using lexicons comprising variants generated by the g2p and p2p converters and auditorily verified transcriptions will be described. These experiments aim to assess the quality of the automatically generated transcriptions and to set some performance thresholds for the experiments conducted in this dissertation. We will also evaluate two different pronunciation variant selection strategies. The performance of these selection strategies will serve as a baseline for our work.

As previously discussed, the confusability between lexicon entries generally increases with the vocabulary size. To evaluate our variant selection approaches in environments with different levels of lexical confusion, all experiments described in this dissertation will be conducted in two different environments: in a controlled setting of 617 names and in an open setting, using a much larger vocabulary of 16,045 names. In the large vocabulary experiments, 15,428 of the names will be "filler names" while the remaining 617 names will be the same names as in the controlled experiments. Since our variant selection methods can only select pronunciation variants of names for which we have training utterances, the variants selected for the filler names will not be optimized in terms of recognition performance. The number of variants generated for each filler name will, however, equal the average number of variants for the 617 names in the data set so as to simulate the behavior of the evaluated variant selection approach. For the same reason, the performance of the large vocabulary lexicons will only be evaluated on a subset of the names in the lexicon. Moreover, since the p2p converters generating variants for the filler names were trained on unseen names (as opposed to the names in the data set), the variants generated for the filler names are likely to be somewhat less accurate than the variants generated for the 617 names in the data set. For these reasons one should be very careful when interpreting the large vocabulary results. However, when put in context, these experiments can give a fairly good indication of the effect an increased vocabulary can have on the recognition performance.

Although it may be clear that this is a somewhat artificial set-up, there are some real-life applications. For instance, if a large vocabulary recognition lexicon contains an identifiable subset of "difficult" entries, improving the recognition performance for these words specifically will increase the performance of the entire system considerably. By using only a modest amount of training examples and a lexicon optimization algorithm similar to the ones described in this dissertation, a system-wide performance increase can be achieved with limited cost.

Throughout this dissertation, recognition performances will be given in terms of the *Name Error Rate (NER)*. In this performance measure, a name (e.g. New York) is only considered correct if all of its constituents (words) are correctly recognized. When comparing the NER of speech recognition system A with the NER of speech recognition system B in this dissertation, the term “significant” is only used when system A achieves an NER that is outside the 95% confidence interval calculated for the NER of system B. The confidence intervals for selected name error rates are calculated in Appendix A.

4.3.1 Threshold experiments

In order to set some performance thresholds, we conducted recognition tests in both a controlled and open environment using the lexicons given in Table 4.5. This section describes the results of these experiments.

Testing in a controlled environment

The results of the three-fold cross validation procedure conducted in a small vocabulary environment are displayed in Table 4.6. This table shows for each lexicon the average NER as well as the average lexicon size, defined as the average number of pronunciation variants contained in the three lexicons.

Lexicon	Size	NER
NO g2p	617	34.44%
EN g2p	617	23.79%
NOEN g2p	1,227	15.74%
NOEN g2p-p2p	5,555	12.50%
AV	1,805	10.66%
AV NOEN g2p-p2p	6,058	10.17%

Table 4.6: *Lexicon size and NER for the reference lexicons in a controlled environment.*

As the results in this table illustrate, the Norwegian g2p transcriptions perform quite poorly. This is mainly due to the fact that the pronunciation of English proper names deviates significantly from what Norwegian pronunciation rules predict. Moreover, the Norwegian g2p converter is not a state-of-the-art converter, such as the English g2p converter which performs considerably better. The English and Norwegian g2p lexicons do, however, correct different recognition errors, as illustrated by the recognition result for the NOEN g2p lexicon. Since the experiments described in this thesis aim to select variants from the pool of g2p-p2p variants, it is interesting to note the performance of simply adding all available pronunciation variants to the lexicon (NOEN g2p-p2p). Using this lexicon further reduced the NER, but the lexicon now contained over four times as many pronunciation variants as the NOEN g2p lexicon. The last two recognition tests (AV and AV NOEN g2p-p2p) are "cheating" experiments performed to attain some reference performances when using ideal lexicons comprising manually corrected transcriptions of the utterances in the training set.

Testing in an open environment

To obtain threshold performances in the case of an open environment, we created six new lexicons in the same way as before. For the Norwegian and English g2p and g2p-p2p lexicons, Norwegian and English g2p and g2p-p2p variants were added for the filler names. Since there were no AV transcriptions available for the filler names, the three most probable g2p-p2p transcriptions were included for each filler name when testing the AV lexicon. The reason why we selected three variants to represent every filler name was that there are, on average, three unique AV variants representing each name in the data set. The lexicons were tested using the same three-fold test procedure as in the case of the small vocabulary experiment. The results are given in Table 4.7.

Lexicon	Size	NER
NO g2p	16,045	55,21%
EN g2p	16,045	39,01%
NOEN g2p	32,072	29,11%
NOEN g2p-p2p	155,141	22.40%
AV	47,928	19.29%
AV NOEN g2p-p2p	156,516	18.40%

Table 4.7: *Lexicon size and Name Error Rate (NER) for the reference lexicons in case of a 16k vocabulary.*

Table 4.7 confirms that the general performance of the recognition system deteriorates when the vocabulary size increases and that the differences in performance between the lexicons are larger than in the case of a small vocabulary. As in the controlled experiment, the NOEN g2p lexicon considerably outperforms both the Norwegian and the English g2p lexicons. Adding p2p variants to this lexicon (NOEN g2p-p2p) resulted in a further performance increase, in spite of containing almost five times as many variants as the NOEN g2p lexicon. The AV lexicon decreased the NER even more, which illustrates the positive effect of having a lexicon consisting of a few high quality pronunciation variants rather than many less accurate variants. A somewhat more surprising result was the performance improvement observed when adding AV variants to the g2p-p2p lexicon (AV NOEN g2p-p2p). Although this experiment is rather artificial in the sense that we only add AV variants to the names that we are testing on, we expected the performance of the last experiment to be closer to that of the NOEN g2p-p2p lexicon due to the large number of variants in the lexicon. A closer inspection of the small vocabulary results shows that the same effect is in fact also present in small vocabulary settings, though this is not equally surprising as the number of variants in the lexicon is much smaller. In any case, both of these results do seem to indicate that having accurate transcriptions in the lexicon is more important than having a compact lexicon with minimal lexical confusion.

To investigate this further, we performed an additional experiment using a lexicon comprising all the g2p-p2p variants of the filler names (149,973 variants) and only the AV variants for the 617 names in our data set (1805 variants). We will refer to this lexicon as the "AV + Filler" lexicon. Testing this lexicon in a large vocabulary environment resulted in an NER of 23.36%. Puzzled by this, we inspected the errors made by this lexicon and compared them with the errors made by the NOEN g2p-p2p lexicon and by the AV NOEN g2p-p2p lexicon. The first thing we noticed was that the chance of a name utterance being incorrectly recognized as a filler name was considerably reduced when using a lexicon containing g2p-p2p and AV variants for the names in the data set. To investigate this further, we compared the N-best lists produced by the recognizer when decoding the name utterances that resulted in different recognition results for the three lexicons. This confirmed our suspicion that in many cases where the "AV + Filler" lexicon resulted in a misclassification, the correct name was somewhere in the top part of the N-best list, with an acoustic likelihood score close to that of the best hypothesis. In the AV NOEN g2p-p2p lexicon, then, these names were represented by a slightly more accurate variant, which was enough to outperform the main (incorrect) competitor.

Even though it is difficult to draw definite conclusions from this experiment, we would like to put forward some tentative observations. It seems that lexicons with a high level of lexical confusion generally benefit more from having a larger set of high quality transcriptions, as the chance of getting a good acoustic match between these variants and a name utterance is higher than if the lexicon contained fewer variants of lower quality. An optimal recognition lexicon should therefore contain as few pronunciation variants as possible, but at the same time contain enough high quality variants to accurately describe different pronunciation effects. These contradictory lexicon properties must be at the heart of our considerations when designing improved variant selection criteria in the next chapters.

4.3.2 Baseline experiments

In order to assess the performance of different variant selection methods, we evaluated the performance of two baseline selection approaches. The first selection approach simply consisted of adding a predefined number of randomly selected pronunciation variants to the recognition lexicon. The second selection approach used the variant probabilities generated by the p2p converters to decide in which order variants should be added to the lexicon. In both approaches, the variants were selected from our pool of automatically generated p2p-g2p pronunciation candidates.

Random variant selection

In this baseline experiment, the lexicons were constructed by randomly selecting variants from the pool of p2p-g2p pronunciation variants. In order to prevent outlier selections, the experiment was repeated five times. We started each of these five experiments with a lexicon comprising one randomly selected variant per name. This variant was then removed from the candidate pool. In each subsequent iteration, the lexicon was supplemented with another randomly selected variant for each name still present in the candidate pool, until there were no more variants available.[10]

This selection method was tested in a controlled and in an open environment. In Figure 4.2 and Figure 4.3, the obtained recognition results are illustrated as a function of M, the maximum allowed number of variants per name.[11] The five random experiments are illustrated in blue, while the

[10] As the number of variants generated by the p2p converters is name-dependent, the total amount of iterations differed from name to name. The amount of available variants ranged from 1 to 22.

[11] As the candidate pool does not contain the same number of variants for each name, not all lexicon entries are represented with M transcription variants in the lexicons evaluated here.

average performance of these experiments is illustrated in red.

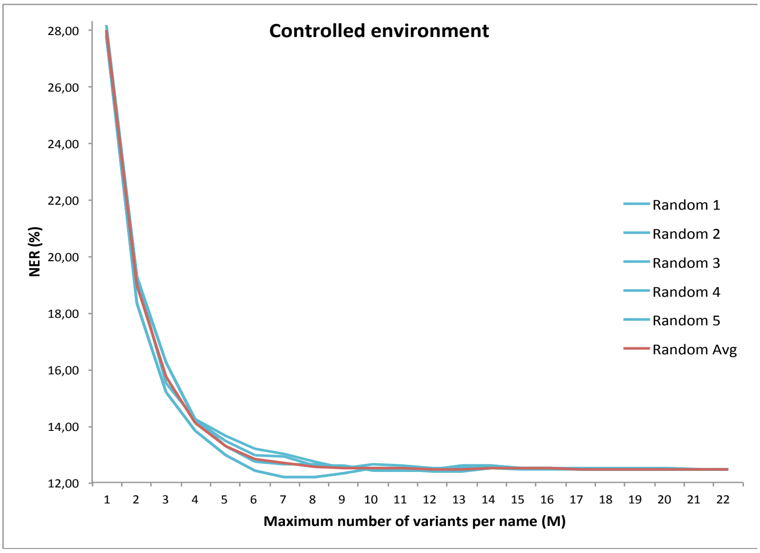

Figure 4.2: *Random variant selection in a controlled environment.*

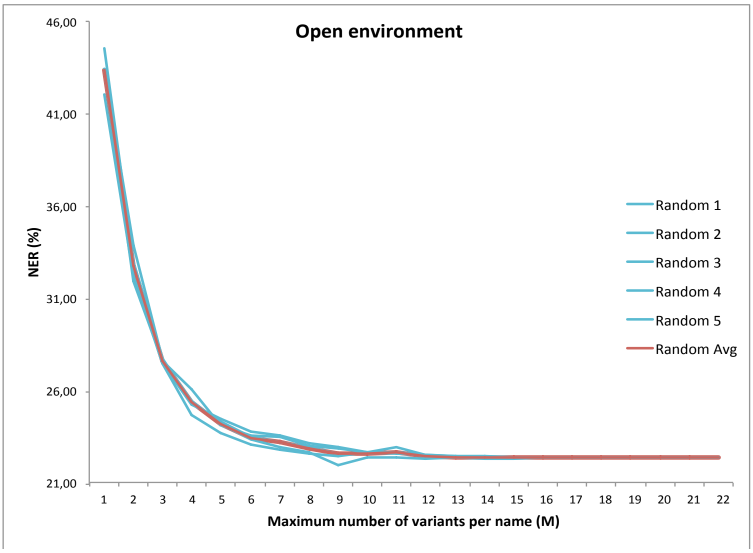

Figure 4.3: *Random variant selection in an open environment.*

As might be expected, these results show a drastic reduction in name error rate in each iteration for the first few variants. After iteration M=4, the steepness of the curve is gradually reduced until it becomes completely flat after around 10 variants per name. Moreover, no gain could be observed in favor of the lexicons containing fewer variants (i.e. the lexicons expected to have low lexical confusion) compared to simply using all available variants. A reasonable hypothesis of why this is the case is that when adding variants randomly, some of the added variants correct errors and some introduce errors. The lexical confusion is therefore more or less uniformly distributed across the variant space. However, if we were able to identify which variants correct errors and which variants introduce errors we could prioritize adding the former. The result of this would probably be a curve that begins at a much lower NER and where the NER decreases further until we start adding the variants that introduce more errors than they correct. At this point, we would expect the NER to start increasing again and the curve to bend upwards. In order to test this hypothesis, we need to find a variant selection criterion that reflects a variant's potential to correct recognition errors.

Probability-based variant selection

One interesting feature of the p2p converter is that it associates a probability with every new variant it generates. This numerical value can be interpreted as an indication of the expected quality of the variant. As such, it might be interesting to test the probabilities as a first approximation of the variant selection criterion we are trying to develop in this dissertation. We would expect that the variants with the higher probabilities have the greater potential to correct recognition errors, and that lexicons containing only variants with high probabilities will lead to increased recognition performance. We investigated this hypothesis in our second baseline experiment.

As in the random baseline experiment, our starting point was a lexicon containing only one variant per name, but now we specifically selected the variants for which the p2p converter estimated the highest probabilities. In the subsequent iterations, we added one variant per name in decreasing order of probability. For the large vocabulary lexicons the procedure was the same, only this time the selection was made from the extended pool of g2p-p2p variants. The recognition results obtained with this method in a controlled and in an open setting are illustrated in blue in Figure 4.4 and Figure 4.5 respectively. Again we see the Name Error Rate as a function of M, the maximum number of variants per name. For reference, the average result of the five random selection experiments performed in the previous section is illustrated by the red curve.

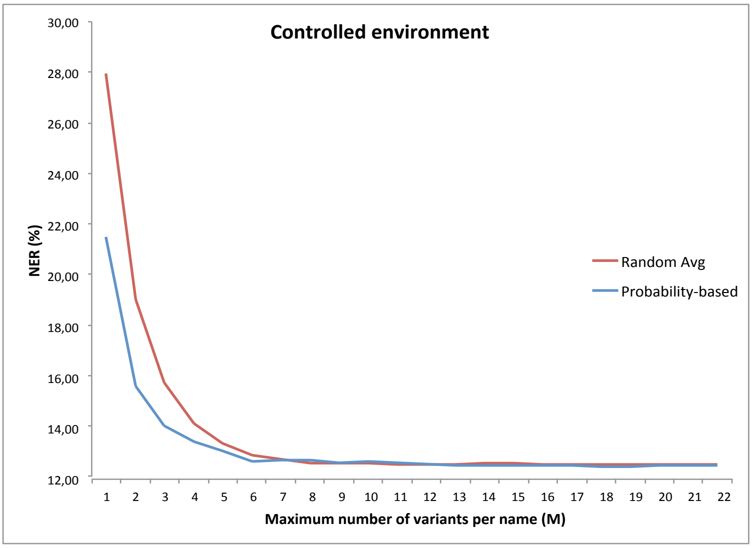

Figure 4.4: *Probability-based variant selection in a controlled environment.*

These figures show that the probability-based variant selection method performs better than the random selection method for M less than four. The main benefit from using this method compared to the random selection method seems to be its ability to make a better selection for the first variants. As in the case of the random experiment, no gain could be observed in favor of the lexicons containing a small number of variants compared to the lexicon comprising all available variants. This result is somewhat surprising, as previous experiments reported in the literature (e.g. [44] and [45]) state that increasing the number of variants in the lexicon is helpful only up to a certain point (typically around 2-3 variants per lexicon entry). After this point the increased lexical confusion tends to counteract the performance gain obtained from having more accurate variants in the lexicon. According to these findings, we expected to find a dip in performance in Figure 4.4 and Figure 4.5 around this point, which is obviously not the case.

One plausible reason why this expected performance dip does not occur, is that the probabilities generated from the p2p converter are not sufficiently accurate, or at least not sufficiently correlated with the actual recognition performance. A closer inspection of the probabilities generated by the p2p converter reveals that the less probable variants seem to have very similar

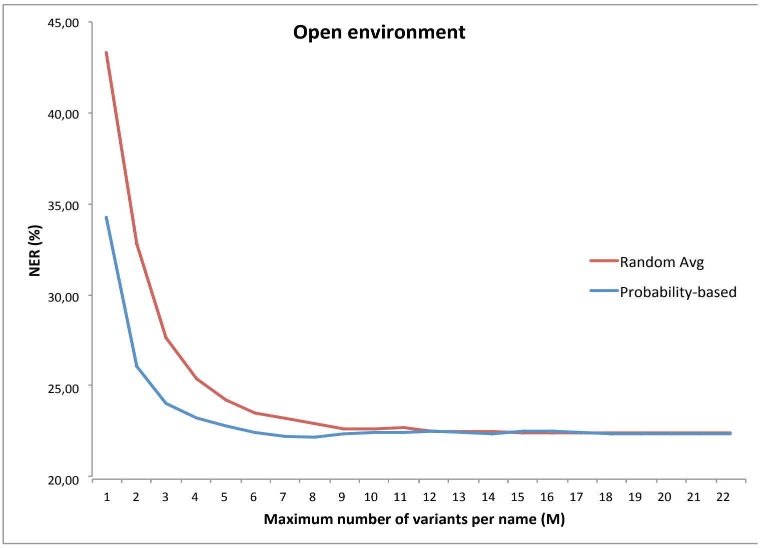

Figure 4.5: *Probability-based variant selection in an open environment.*

probabilities. This usually happens after the third or fourth most probable variant. This can signify that the p2p converter does not have enough evidence to adequately estimate probabilities for these variants, making their internal ranking subject to random effects. Another factor which may influence the probability-based selection method is the fact that the g2p-p2p variant pool contains variants generated both by English and Norwegian g2p-p2p tandems. The most probable variants in this pool are therefore likely to be one or two variants from the English converter and one or two variants from the Norwegian converter. In cases where the pronunciation of a name is similar in English and in Norwegian, the two (or even the four) most probable pronunciation variants in the pool are likely to represent the same pronunciation effects, and are thus likely to correct the same recognition errors. In this way, selecting variants according to their predetermined probability can result in a lexicon containing many overlapping variants. For these reasons, it was deemed necessary to define a new selection criterion, which is more closely related to the actual recognition performance.

4.4 Conclusion

In this chapter we have described the design and collection of the NameDat database as well as the generation of alternative name transcriptions using Norwegian and English g2p and p2p converters. Together with the manually corrected transcriptions from the NameDat database, these alternative transcriptions were used in a set of preliminary experiments in order to get some performance thresholds. As expected, these experiments showed that the general system performance decreased considerably when we increased the vocabulary from 617 names to 16,045 names. Furthermore, the experiments illustrated the positive effect of adding more transcription variants to the lexicon to supplement the automatically generated g2p variants. The "cheating" experiments (using manually corrected transcription variants) showed that lexical confusability can in fact be reduced by the presence of accurate transcription variants in the lexicon. This means that having high quality transcription variants in the lexicon can be just as important, or even more important, than reducing the number of variants when it comes to reducing confusability and achieving high performance rates.

The baseline experiments showed that the selection criterion used for variant selection (which determines the order in which the variants are added) has a considerable effect on the performance, especially when the maximum number of variants per name is small. These experiments further showed that neither of the baseline selection criteria investigated in this chapter were able to create a lexicon performing better than a lexicon comprising all available transcription variants, which is somewhat surprising considering similar studies reported in the literature. This is likely to be attributed to the poor ability of our baseline selection criteria to select the variants yielding optimal performance.

These findings inspired a search for a new variant selection criterion that identifies and selects only the transcription variants that effectively reduce the Name Error Rate. It is our working hypothesis that such a selection criterion can result in a lexicon that is doubly optimized, both in terms of its size and in terms of its recognition performance. In the following chapters we therefore aspire to create a variant selection criterion that:

- identifies the most accurate pronunciation variants;
- identifies variants that correct more errors than they introduce;
- identifies pronunciation variants covering different pronunciation phenomena;

- optimizes the order in which variants are added to the lexicon;
- optimizes the number of variants in the lexicon.

Chapter 5

Finding the optimal variant selection criterion

In the previous chapter, two baseline variant selection approaches were evaluated. In the first approach, pronunciation variants were selected randomly from a pool of candidate transcriptions. In the second approach, variants were selected on the basis of probabilities learned during variant generation. The evaluation of the two selection approaches revealed that both approaches performed quite poorly. This was not very surprising in case of the random selection approach, but for the probability-based approach we expected a somewhat better performance. This lack of performance gain was attributed to inaccurate variant probabilities and low correlation between the variant probabilities and the actual recognition performance.

In this chapter, we propose two different discriminative selection criteria both of which are directly related to the recognition performance. The two criteria are based on a set of prior probabilities which are estimated by means of scores calculated using the discriminative Maximum Entropy (ME) framework and the Minimum Classification Error (MCE) framework. To compare the performance of these selection criteria to that of the probability-based approach, we estimate a third set of prior probabilities using the probabilities generated by the p2p converter (P2P). The three different sets of pronunciation priors are then evaluated in isolation (prior selection) and together with acoustic log likelihood scores generated by the recognizer (posterior selection). Parts of the work described in this chapter was also presented in [2].

5.1 Decision rules and variant selection algorithm

In this chapter we will consider the variant selection problem as a decision problem, where the task is to decide which pronunciation variants result in the best performing recognition lexicon. In Section 5.1.1, we define the decision rule to be used in this selection algorithm and Section 5.1.2, then, describes the actual variant selection algorithm.

5.1.1 Decision rule

Again, let us assume that we have a set of names $\mathcal{W} = \{W_1, \dots, W_K\}$, and that for some name $W_k \in \mathcal{W}$ we have a set of training utterances $\mathcal{X}_k = \{X_{k1}, \dots, X_{kN}\}$ and a set of candidate pronunciation variants $\mathcal{V}_k = \{V_{k1}, \dots, V_{kI}\}$. Additionally, we construct a start lexicon Λ_s containing English g2p transcriptions for all items in our set of names $\mathcal{W}$. As we aim to find variant selection approaches that are directly related to the actual recognition performance, evidence of the performance of all available variants is needed. To that end, we create a temporary lexicon Λ_{ki} for each candidate pronunciation variant V_{ki} in the candidate pool. This temporary lexicon is constructed by copying our start lexicon Λ_s and replacing the g2p transcription of name W_k by the candidate pronunciation variant V_{ki}.

From Section 2.2.1 we know that the optimal word hypothesis according to the MAP decision rule is

$$\hat{W} = \operatorname*{argmax}_{W_k \in \mathcal{W}} \sum_{i=1}^{I} P(\mathcal{X}_k | \Lambda_{ki}; W_k) P(\Lambda_{ki} | W_k) P(W_k).$$

Assuming that the contribution from the best performing pronunciation variant will be considerably larger than the remaining contributions, this sum can be approximated by

$$\hat{W} \cong \operatorname*{argmax}_{W_k \in \mathcal{W}} \operatorname*{argmax}_{V_{ki} \in \mathcal{V}} P(\mathcal{X}_k | \Lambda_{ki}; W_k) P(\Lambda_{ki} | W_k) P(W_k).$$

Suppose, then, that we already know that word $W = W_k$. Then we can use this decision rule to find the optimal pronunciation variant V_k^* by selecting the variant that maximizes

$$V_k^* \cong \operatorname*{argmax}_{V_{ki} \in \mathcal{V}_k} P(\mathcal{X}_k | \Lambda_{ki}; W_k) P(\Lambda_{ki} | W_k) P(W_k). \tag{5.1}$$

Here, $P(\mathcal{X}_k | \Lambda_{ki}; W_k)$ is the acoustic likelihood of the training utterances of name W_k, $P(\Lambda_{ki} | W_k)$ is the pronunciation prior probability of the temporary lexicon Λ_{ki} and $P(W_k)$ is the probability of word W_k. In this chapter we evaluate two different versions of this decision rule: a simplified version, which we will refer to as the *prior* variant selection approach, and a full version, which we will call the *posterior* variant selection approach. In the prior variant selection approach, the pronunciation prior probabilities are used alone to select which variants to include in the lexicon:

$$V_k^* = \operatorname*{argmax}_{V_{ki} \in \mathcal{V}_k} \hat{P}(\Lambda_{ki}|W_k). \tag{5.2}$$

In the posterior variant selection approach, the pronunciation variants are selected by means of a combination of the prior probabilities and the total acousthic log likelihood scores generated by the recognizer

$$V_k^* = \operatorname*{argmax}_{V_{ki} \in \mathcal{V}_k} \left\{ \sum_{n=1}^{N} \log P(X_{kn}|\Lambda_{ki}; W_k) + \gamma \log \hat{P}(\Lambda_{ki}|W_k) \right\}. \tag{5.3}$$

Again, $P(X_{kn}|\Lambda_{ki}; W_k)$ is the acoustic likelihood of name utterance X_{kn} given that variant V_{ki} is in the lexicon, γ is a scaling factor and $P(\Lambda_{ki}|W_k)$ is the pronunciation prior probability. The scaling factor was determined using a systematic trial and error procedure on the training set. This is a suboptimal approach, which may have influenced the results to a small extent, but was deemed necessary as our data set was too small to contain an independent development set. The scaling factor was determined to be 20 for the small vocabulary experiments and 2.5 for the large vocabulary experiments. This rather large difference in scaling factors can be attributed to the fact that the difference in acoustic likelihood between different hypotheses in the N-best list tends to be smaller in the large vocabulary case compared to the small vocabulary case, due to the increased confusability.

In both the prior and the posterior decision rules, we have disregarded the word probability distribution $P(W_k)$ since every word in our corpus is assumed to be equally likely.

5.1.2 Variant selection algorithm

To find the optimal pronunciation variants to represent name W_k in the end lexicon, we extracted a pronunciation candidate set $\mathcal{V}_k$ and a set of training utterances $\mathcal{X}_k$ from the training data and performed the following single-pass selection procedure for each name $W_k \in \mathcal{W}$:

1. for each candidate $V_{ki} \in \mathcal{V}_k$:
    - (a) create a temporary lexicon Λ_{ki} by *replacing* the g2p transcription in the start lexicon Λ_s by the candidate V_{ki}
    - (b) create an empty set $\mathcal{H}_{ki}$ to be used as a storage container for recognition hypotheses
    - (c) for each training utterance $X_{kn} \in \mathcal{X}_k$:
        - i. perform an isolated word recognition on X_{kn} using the temporary lexicon Λ_{ki},
        - ii. add the N-best list H_{kn} of the most likely name hypotheses[1] together with their normalized likelihood scores to the set $\mathcal{H}_{ki}$,
    - (d) retrieve the likelihood scores for name W_k from the N-best lists in $\mathcal{H}_{ki}$ and take the logarithm[2]
    - (e) calculate the total log likelihood score of name W_k given lexicon Λ_{ki} by summing these log likelihood scores
    - (f) estimate the prior probability $\hat{P}(\Lambda_{ki}|W_k)$ by using one of the methods proposed in the following subsection
2. select for each name the M variants maximizing
    - (a) Equation (5.2) (prior variant selection) and add these variants to the prior end lexicon[3]
    - (b) Equation (5.3) (posterior variant selection) and add these variants to the posterior end lexicon[3]

In the next subsection we propose three different ways of estimating the pronunciation prior probabilities namely using variant probabilities, Maximum Entropy scores and Minimum Classification Error scores.

[1] The maximum number of hypotheses in H_{kn} was set to 20.

[2] If W_k was not in H_{kn}, it was given a likelihood score of -200. This value was lower than any of the likelihood scores in the N-best lists, but not so low as to skew the total likelihood score computed in the next step.

[3] If two variants have equal values, choose the variant with the highest total log likelihood score.

5.1.3 Estimating pronunciation priors

In this subsection the pronunciation prior probability, $P(\Lambda_{ki}|W_k)$, will be estimated in three different ways using variant information based on: probabilities learned during variant generation, calculated Maximum Entropy scores and calculated Minimum Classification Error scores.

Using variant probabilities as pronunciation priors

In the experiments conducted in this thesis, the pool of candidate pronunciation variants $\mathcal{V} = (\mathcal{V}_1, \ldots, \mathcal{V}_K)$ was generated using phoneme-to-phoneme (p2p) converters (as described in Section 4.2.4). These p2p converters were trained using an automatically generated grapheme-to-phoneme transcription G_k of every name W_k and a set of auditorily verified transcriptions $\mathcal{T}_k = \{T_{k1}, \ldots, T_{kN}\}$ of the utterances in the training set $\mathcal{X}_k$. The p2p converter assigns a probability, $P(V_{ki}|\mathcal{T}_k; G_k)$, to every pronunciation variant $V_{ki} \in \mathcal{V}$ by utilizing probabilities learned during training. Using this variant probability as an estimate of the pronunciation prior probability, $\hat{P}(\Lambda_{ki}|W_k)$, will serve as a direct reference to the probability-based baseline experiment conducted in the previous chapter

$$\hat{P}(\Lambda_{ki}|W_k) = P(V_{ki}|\mathcal{T}_k; G_k).$$

Obtaining prior probabilities for the p2p approach was exceedingly straightforward, as they were taken to be the probabilities generated by the p2p converters during variant generation. A detailed description on the generation of these probabilities can be found in [105] and [84].

Using Maximum Entropy scores as pronunciation priors

Entropy is defined as a measure of the uncertainty of a probabilistic distribution. As discussed in Section 3.6.3, the maximum entropy principle states that we should choose the distribution with the maximum entropy (the most uniform distribution) that satisfies a set of constraints.

In this work we use one single constraint namely whether an utterance x_k is recognized correctly or not when the lexicon Λ_{ki} is being used. Employing the Maximum Entropy model to estimate the pronunciation prior probability we get

$$\hat{P}(\Lambda_{ki}|W_k) = P(\Lambda_{ki}|c_k) = \frac{1}{Z(c_k)} \exp\left[\sum_{j=1}^{F} \lambda_j f_j(c_k, \Lambda_{ki})\right] \tag{5.4}$$

as described in Section 3.6.3. Now, if we have a set of training utterances $\mathcal{X}_k$ for each name W_k, we expect this pronunciation prior probability to reflect the number of correct classifications when using lexicon Λ_{ki}. To incorporate this into the Maximum Entropy model, the best hypothesis in the hypothesis list H_{kn} was extracted for all the name utterances $X_{kn} \in \mathcal{X}_k$. When one of these hypotheses was equal to name W_k, the binary feature in Equation (3.3) was set to 1 for variant V_{ki} and name utterance X_{kn}. In this way, the Maximum Entropy model was constrained to model c_k with the same frequency as was observed in the training data. The parametric model in Equation (5.4) was then trained for every name using these features. The pronunciation prior probabilities were finally extracted from this model for every variant $V_{ki} \in \mathcal{V}_k$.

Using MCE scores as pronunciation priors

Another discriminative framework that can be used to estimate the pronunciation prior probability, is the Minimum Classification Error framework described in Section 3.6.4. In this framework, the performance of lexicon Λ_{ki} is modeled using the expected loss of recognition accuracy given in Equation (3.5). This value is the expectation of the zero-one loss values $l_k(X_{kn};\Lambda)$ (Equation (3.7)) calculated for the available training utterances $X_{kn} \in \mathcal{X}_k$.

The pronunciation prior probability was then estimated by simply taking the complement of the loss function $l_k(X_{kn};\Lambda_{ki})$

$$\begin{aligned}\hat{P}(\Lambda_{ki}|W_k) &= P(\Lambda_{ki}|x_k \text{ is correctly recognized}) \\ &= \frac{1}{N}\sum_{n=1}^{N}(1 - l_k(X_{kn};\Lambda_{ki})). \qquad (5.5)\end{aligned}$$

In the work described in this thesis, acoustic log likelihoods extracted from the hypothesis list H_{kn} were used as discriminant functions $g(X_{kn};\Lambda)$ for utterance X_{kn}. As a consequence of the pruning parameter employed by the HTK recognizer (see Section 2.1.5 for details), the number of hypotheses in the list H_{kn} varies for each utterance in the training set. A long list means high confusability whereas a short list means a lower level of confusability between lexicon entries. Since long H_{kn} lists normally entail a lower average likelihood score among the competitors when calculating the misclassification measure in Equation (3.6) compared to shorter H_{kn} lists, we carefully tuned the η parameter in Equation (3.6) using a systematic trial and error procedure on the training set.[4] The value that was found to give a good competitor weight was $\eta = 6$.

To calculate the total loss value for lexicon Λ_{ki}, Equation (3.7) was calculated for every name utterance $X_{kn} \in \mathcal{X}_k$ using the corresponding list of hypotheses H_{kn}. If name W_k did not appear in H_{kn}, it was given a loss value equal to 1 for that utterance. Finally, the pronunciation prior probability was retained by using these loss values and Equation (5.5).

5.2 Experiments and results

In the experiments described in this section we compare three prior and four posterior selection criteria and evaluate their ability to select the best performing pronunciation variants for inclusion in the recognition lexicon. The experiments were conducted in two different environments: in a controlled setting of 617 names and in an open setting, using a much larger vocabulary of 16,045 names. The three-fold cross-validation procedure described in Section 4.2.3, using data extracted from the NameDat database, was used in both experiments. The results in this section are presented as the average name error rate (NER) of the three test sets and are given as a function of M, the maximum allowed number of variants per name. Since virtually no variation in performance was observed in the baseline experiments for large values of M, the results in this chapter are given for up to ten variants per name ($M = 10$).

5.2.1 Testing in a controlled environment

The prior and posterior selection criteria described in the previous section were first tested in a controlled environment, using a single-word grammar containing only the 617 names of the NameDat corpus. By using only the pronunciation prior probabilities as the selection criterion (Equation (5.2)), we obtained the results listed in the columns on the left-hand side of Table 5.1. The right-hand side of the table shows the results obtained using the posterior probabilities as the selection criterion (Equation (5.3)). In both cases, the p2p variant probabilities (P2P), the Maximum Entropy score (ME) and the Minimum Classification Error score (MCE) described above were used as estimates of the pronunciation prior probability. The additional column on the posterior selection side of the table shows the results

[4] As previously noted, this is a suboptimal approach which may have influenced the results to some small extent, but considered necessary due to the limited size of our data set.

obtained using only the acoustic log likelihood scores: NPP stands for no prior probability.

	Prior selection			Posterior selection			
M	P2P	ME	MCE	NPP	P2P	ME	MCE
1	21.51%	13.77%	13.53%	13.67%	16.16%	13.73%	13.67%
2	15.64%	13.22%	12.85%	13.12%	13.95%	13.25%	12.99%
3	14.08%	12.37%	12.26%	12.43%	13.38%	12.35%	12.44%
4	13.45%	12.33%	12.22%	12.27%	12.86%	12.23%	12.24%
5	13.07%	12.20%	12.23%	12.29%	12.74%	12.20%	12.23%
6	12.66%	12.30%	12.20%	12.33%	12.60%	12.33%	12.33%
7	12.71%	12.34%	12.25%	12.33%	12.60%	12.34%	12.33%
8	12.68%	12.34%	12.31%	12.37%	12.60%	12.34%	12.39%
9	12.61%	12.34%	12.37%	12.37%	12.55%	12.37%	12.42%
10	12.63%	12.47%	12.34%	12.42%	12.58%	12.47%	12.40%

Table 5.1: *NER of lexicons created with the prior selection method and the posterior selection method ($\gamma = 20$).*

The figures on the left-hand side of Table 5.1 reveal that the discriminative prior selection approaches (ME and MCE) significantly outperformed the probability-based P2P method for the first three values of M. For values of M larger than four, the discriminative selection approaches also outperformed the P2P method, but with a somewhat smaller margin. To better demonstrate the differences in performance between the various selection approaches, the results of Table 5.1 are illustrated in Figure 5.1 and 5.2 as a function of the lexicon size. In these figures the x-axis is represented on a logarithmic scale, as will be the case for all performance graphs throughout this dissertation. Figure 5.1 shows the results of the prior selection approaches and Figure 5.2 illustrates the results of the posterior selection approaches.

When studying Figure 5.1, we observe a marginal performance gain in favor of the MCE approach. Figure 5.2 shows that when compared to an approach using no prior probabilities (NPP), none of the posterior selection approaches seems able to exploit the additional information contained in the priors. In fact, the differences in performance are so small as to be virtually invisible in Figure 5.2. When compared to their respective prior approaches, the P2P posterior approach shows some improvement, whereas the ME and MCE posterior approaches stay at the same performance level.

Interestingly, when comparing the results illustrated in Figure 5.1 and

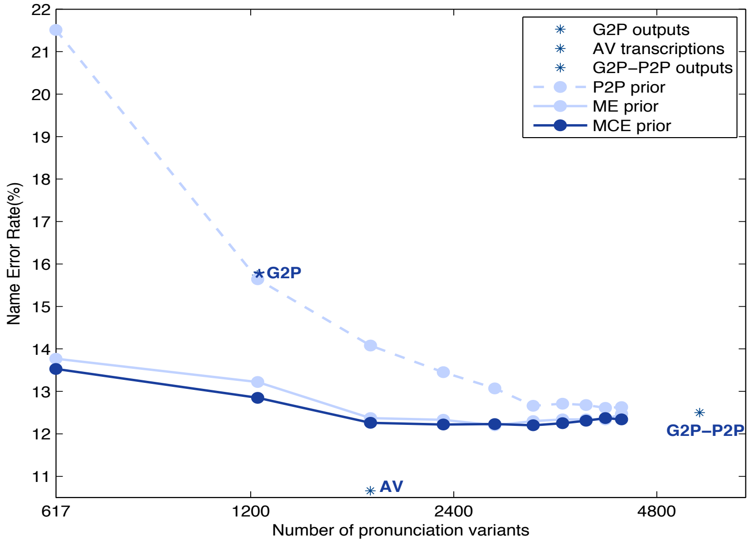

Figure 5.1: *NER of lexicons created with the prior selection method.*

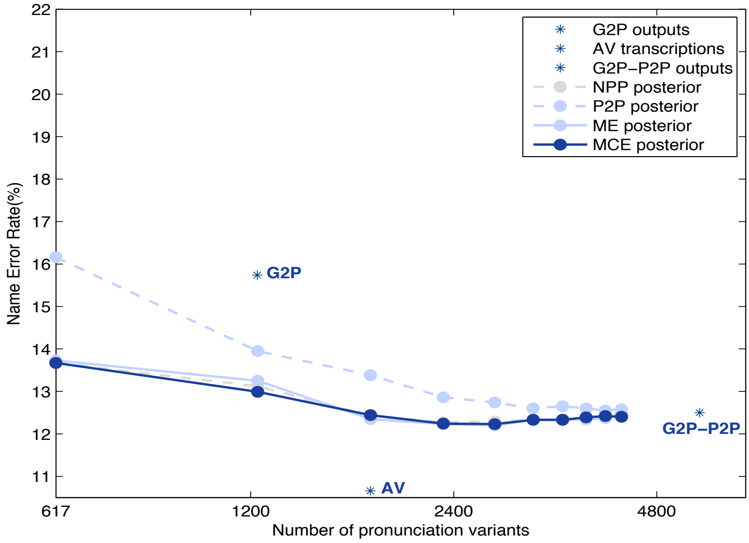

Figure 5.2: *NER of lexicons created with the posterior selection method.*

Figure 5.2 with that of the threshold experiments given in Section 4.3.1

(marked as asterisks in these figures), we observe that it sufficed for all selection approaches, except for the probability-based approaches, to select no more than three pronunciations per name to attain a performance equivalent to that of a much larger lexicon comprising all g2p and p2p transcriptions (G2P-P2P 12.5% NER). However, none of the proposed selection approaches were able to attain a performance comparable to that of a lexicon containing only auditorily verified transcriptions (AV 10.66% NER).

Including AV variants in the candidate set

Since the pronunciation candidates generated by the P2P converters are created using an automatic procedure, they are far from optimal. It is therefore interesting to investigate how the proposed selection criteria behave when the pool of pronunciation candidates also contains variants of a higher quality. Therefore, we repeated the experiment, adding the auditorily verified transcriptions of the training utterances to the pool of pronunciation candidates. Since there is no accurate way of assigning probabilities to AV variants, the P2P prior and the P2P posterior selection approaches were not evaluated in this experiment. Table 5.2 shows the results of this experiment.

	Prior selection		Posterior selection		
M	ME	MCE	NPP	ME	MCE
1	13.05%	12.54%	12.62%	12.82%	12.57%
2	11.70%	11.58%	11.77%	11.64%	11.59%
3	10.90%	11.15%	11.10%	10.94%	11.18%
4	10.77%	10.65%	10.86%	10.72%	10.66%
5	10.51%	10.37%	10.43%	10.41%	10.45%
6	10.39%	10.34%	10.46%	10.34%	10.42%
7	10.40%	10.27%	10.45%	10.35%	10.27%
8	10.27%	10.23%	10.40%	10.32%	10.37%
9	10.25%	10.19%	10.25%	10.22%	10.27%
10	10.35%	10.18%	10.37%	10.37%	10.35%

Table 5.2: *NER of lexicons created with the prior selection method and the posterior selection method when using AV transcriptions in the candidate set ($\gamma = 20$).*

This table shows that the differences in performance between the five selection approaches is still vanishingly small. It is worth noticing, however, that the best performing lexicons reach name error rates comparable to that of a lexicon comprising all variants generated by the g2p and p2p converters and all AV variants (G2P-P2P-AV 10.17% NER), using fewer variants in the lexicon ($M = 9$). Still, as in the original experiment, none of the selection approaches were able to outperform a lexicon comprising only the AV variants (AV 10.66% NER) when using the same amount of variants in the lexicon ($M = 3$). As opposed to the original experiment, however, none of the lexicons generated in this experiment were significantly outperformed by the AV lexicon.

5.2.2 Testing in an open environment

In an attempt to simulate an environment where the discriminative potential of the prior probabilities might be more fully exploited, we repeated the previous recognition experiment in a large vocabulary setting of 16,045 names. The results of this experiment are given in Table 5.3 and illustrated in Figure 5.3 and Figure 5.4.

Since the proposed variant selection methods could only be applied on names for which training utterances were available, we conducted experiments where only the variants selected for the original 617 names were optimized in terms of recognition performance. The filler names were left with their M most probable variants according to the Norwegian and English p2p converters.

	Prior selection			Posterior selection			
M	P2P	ME	MCE	NPP	P2P	ME	MCE
1	34.32%	22.65%	22.78%	22.56%	24.04%	22.57%	22.64%
2	26.13%	22.00%	21.50%	21.79%	22.24%	21.86%	21.78%
3	24.11%	20.93%	21.00%	21.06%	21.52%	21.11%	21.06%
4	23.30%	21.02%	21.09%	21.06%	21.51%	20.99%	21.09%
5	22.86%	21.15%	21.17%	21.30%	21.37%	21.31%	21.25%
6	22.47%	21.35%	21.46%	21.51%	21.57%	21.43%	21.43%
7	22.30%	21.59%	21.54%	21.62%	21.54%	21.56%	21.57%
8	22.39%	21.81%	21.77%	21.85%	21.87%	21.82%	21.79%
9	22.40%	22.14%	22.43%	22.10%	22.15%	22.14%	22.10%
10	22.48%	22.27%	22.27%	22.25%	22.32%	22.27%	22.25%

Table 5.3: *NER of lexicons created with the prior and posterior selection methods in the case of a 16k vocabulary ($\gamma = 2.5$).*

Table 5.3 shows that the P2P prior selection method was again significantly outperformed by the discriminative ME- and MCE-based prior selection methods for the first four values of M. Moreover, the performance gain was relatively larger than in the case of a small vocabulary. However, still virtually no performance gain could be observed for the posterior ME- and MCE-based selection approaches when compared to the NPP posterior approach, as illustrated in Figures 5.3 and 5.4. Furthermore, none of these results give any evidence as to which of the discriminative selection approaches is the better performing one. The difference between the methods actually seems to decrease compared to the results obtained using a small vocabulary.[5]

These figures also reveal that the recognition performances no longer seem to improve with higher values of M. In fact, for lexicons generated by one of the discriminative approaches, three or four variants per name were sufficient to obtain a better recognition performance than using ten variants per name, which is more in agreement with other results reported in the literature (e.g. [44] and [45]). Moreover, for all lexicons but the prior P2P lexicon, no more than two variants per name were needed to achieve a performance surpassing that of a lexicon comprising all variants in the candidate set (G2P-P2P 22.40% NER).

Including AV variants in the candidate set

As in the controlled recognition experiment, we added the auditorily verified transcriptions of the training set to the pool of pronunciation candidates in order to observe what effect high quality variants would have on the performance of the different selection approaches. The results of this experiment are shown in Table 5.4.

As in the other experiments conducted in this chapter, there is no clear evidence which of the proposed selection approaches generates the best performing lexicon. Nevertheless, all the proposed selection approaches needed to select no more than four or five variants per name to obtain a performance equal to that of a lexicon comprising all g2p-p2p variants and all auditorily verified variants (G2P-P2P-AV 18.40% NER). Testing in an open environment, where the lexical confusion is higher, all selection approaches even performed somewhat better than a lexicon containing all auditorily verified variants (AV 19.29% NER) using approximately the same number of variants per name ($M = 3$).

[5] The differences between the posterior approaches are so minute that the corresponding graphs in Figure 5.4 are effectively indistinguishable.

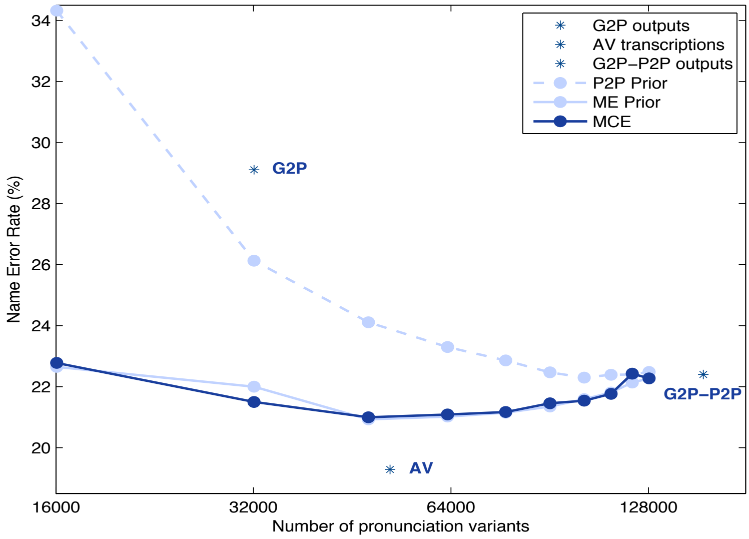

Figure 5.3: *NER of lexicons created with the prior selection method in the case of a 16k vocabulary.*

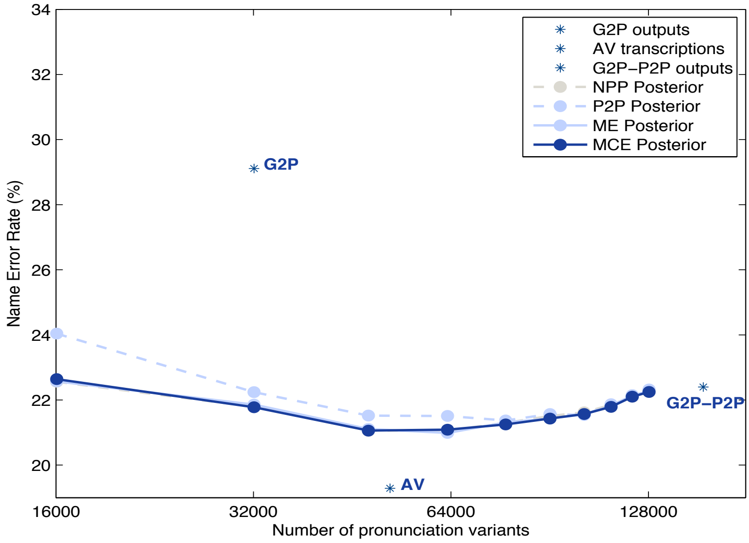

Figure 5.4: *NER of lexicons created with the posterior selection method in the case of a 16k vocabulary.*

	Prior selection		Posterior selection		
M	ME	MCE	NPP	ME	MCE
1	21.26%	20.94%	20.65%	21.16%	20.83%
2	19.88%	19.41%	19.72%	19.88%	19.41%
3	18.56%	18.55%	18.75%	18.72%	18.60%
4	18.36%	18.41%	18.34%	18.47%	18.26%
5	18.02%	18.17%	17.98%	18.07%	18.18%
6	18.15%	18.23%	18.23%	18.21%	18.15%
7	18.12%	18.06%	18.16%	18.14%	18.05%
8	18.24%	18.18%	18.34%	18.31%	18.32%
9	18.29%	18.29%	18.24%	18.29%	18.32%
10	18.42%	18.39%	18.41%	18.42%	18.42%

Table 5.4: *NER of lexicons created with the prior and posterior selection methods in the case of a 16k vocabulary when using AV transcriptions in the candidate set* ($\gamma = 2.5$)*.*

5.3 Discussion

In this chapter, we have proposed two discriminative variant selection criteria and evaluated them using both prior and posterior decision rules. The performance of the proposed selection criteria was compared to that of an approach using p2p probabilities as prior probabilities and to that of an approach using no prior probabilities at all. The experiments performed in the previous section showed that the p2p prior selection method was significantly outperformed, both in a controlled and in an open environment, by all the evaluated prior and poste rior selection criteria, when using a lexicon containing fewer than three variants per name. Moreover, the proposed selection approaches needed to select no more than three (in case of a controlled experiment) or two (in case of an open experiment) variants per name to attain a performance surpassing that of a lexicon containing all p2p and g2p transcriptions. There was no clear evidence, however, of which of the decision rules performed best, nor was there enough evidence to determine the most suitable selection criterion.

In order to identify potential areas of further improvement for our proposed variant selection approaches, then, we require a better understanding of the behavior of our decision rules and the characteristics of our selection criteria. In Section 5.3.1, we therefore analyze the proposed decision rules in greater detail. This analysis contains an in-depth comparison of the prior and posterior decision rules, where we take a closer look at the individual variants selected by the prior decision rule compared to the NPP posterior criterion. In Section 5.3.2, we describe a number of limitations that our single-pass selection procedure is faced with, and identify some prerequisite characteristics of a more successful approach, which we will investigate further in the following chapters. In Section 5.3.3, we reflect on the specific characteristics of our selection criteria, and the differences in selection behavior these entail. Based on these considerations, then, we are able to determine which of our decision rules and selection criteria are best suited for use in our further work.

5.3.1 Choosing the most suitable decision rule

One of the most striking observations to be made from the experimental results presented in this chapter is that there seems to be virtually no difference in performance between the discriminative posterior selection criteria and the NPP posterior selection criterion. To be fair, these three criteria have their *posterior* factor, viz. log likelihood scores from the recognition engine, in common, so we might expect that their performance is similar. Nevertheless, it seems surprising that the differences are so insignificantly small, and that the additional information contained in the MCE and ME *prior* probabilities does not amount to any gain in performance. In order to investigate to what extent the prior probabilities effectively overlap with the NPP's log likelihood information, we examined whether or not the prior approaches corrected different recognition errors than the posterior NPP approach. We therefore performed two detailed error analyses on the small vocabulary results, comparing the variants selected by each of the discriminative prior criteria on the one hand with the variants selected in the NPP posterior approach on the other. We considered all cases where the two approaches selected different pronunciation variants for name W_k and where at least one test utterance of this name was misrecognized by at least one of the two approaches.

In our first error analysis, we compared the variants selected by the ME prior approach to those selected by the NPP posterior approach in a controlled environment for $M = 1$. Strikingly, we observed that these two approaches selected different variants for no more than 5% of the 617 names in the lexicon, indicating a very strong correspondence between the ME approach and the NPP approach. These variants accounted for 16% of the errors made by the two approaches. Of these errors, 34% occurred exclusively in the ME approach, and 30% occurred exclusively in the NPP approach. The observed overlap between ME and NPP may be explained as follows. The ME criterion is designed in such a way that it selects the candidate variants with the highest number of correctly recognized training utterances. However, if two or more variants of a name share the highest number of correct recognitions, the ME criterion has no basis to choose between them. In those cases, the ME approach effectively selects the variant that has the highest log likelihood, which is of course the same selection criterion used in the NPP approach. In the cases where there is a single variant yielding the highest number of correctly recognized training utterances, that variant also tends to have the highest log likelihood, so it need perhaps not have surprised us that the ME criterion and the NPP criterion behave so similarly.

In our second error analysis, we compared the variants selected by the MCE prior approach to those selected by the NPP posterior approach, again in a controlled environment for $M = 1$. We found that these two approaches behaved somewhat more divergently, selecting different variants for 13% of the 617 names in the lexicon. These variants accounted for one fifth of the errors made by the two approaches. Of these errors, 23% occurred exclusively in the MCE approach, and 29% occurred exclusively in the NPP approach. A closer inspection of these errors revealed that the NPP approach generally seems to prefer variants where the number of competitors in the N-best list is higher compared to the variants selected by the MCE prior approach. This effect can be largely attributed to the η parameter in the misclassification measure of the MCE framework (Equation (3.6) of Section 3.6.4) which makes the MCE criterion less dependent on the number of hypotheses in the N-best list. There is no clear evidence, however, that this effect has a notable impact on the recognition performance. Nevertheless, in those cases where both approaches selected variants corresponding with long N-best lists, signifying high confusion, we generally observed that the variant selected by the MCE prior approach seemed to be the better variant.

It seems, then, that the errors occurring exclusively in the ME prior approach were caused by giving too much weight to the number of correctly recognized training utterances, while the errors occurring only in the NPP approach were caused by relying too heavily on the acoustic alignment. The MCE prior approach combines these two components, taking into account both log likelihood scores and the amount of correct recognitions. This should result in a fairer balance between errors made on account of either of these factors, although we must note that the results for the MCE approach presented in this chapter are not particularly better than the results for the other approaches.

When we consider the question of *prior* versus *posterior* decision rule, however, it may be clear that the latter will only upset the balance that might exist in the prior ME and MCE approaches, as it will give even more weight to the acoustic alignment than is already built into the prior decision rules. The upshot of this is that our proposed selection criteria come to behave almost identically to the NPP approach, where we use only log likelihood scores from the recognition engine. Indeed, we have seen that the prior approaches already overwhelmingly select the same variants as the NPP approach, so it may be clear that no further bias in favor of acoustic alignment is desirable. In our attempt to identify a method to optimize recognition lexicons, we will therefore abandon the posterior selection approaches in the remainder of this dissertation, and focus instead on improving our application of the prior decision rule.

5.3.2 Limitations of the single-pass selection approach

As we strive to refine our utilization of the prior decision rule, we might do well to take a step back and reflect on some of the weaknesses of the approaches proposed in this chapter. In this section, we highlight two main limitations, and attempt to formulate some potential areas of further improvement these entail.

One major weakness of the selection approaches proposed in this chapter is their tendency to select multiple equivalent variants. The evaluation of whether or not a pronunciation variant of a given name should be included in the recognition lexicon does not take into account which variants of that name might already be present in the lexicon. This makes it impossible to favor the selection of alternative variants that are capable of correcting recognition errors that were not previously handled by the variants already in the lexicon. Indeed, including variants correcting the same errors is exceedingly likely, since two similar pronunciation variants are likely to get similar scores, making them equally ranked for inclusion in the lexicon. For instance, if our data set contains utterances of a particular name that are pronounced more or less consistently, but with a small amount of outlier pronunciations, the optimal population of the lexicon would contain one single variant covering the uniform majority of the pronunciations and a number of additional variants covering the anomalous utterances. The selection approaches proposed in this chapter, however, would prioritize the selection of a number of functionally equivalent variants, all of which cover only those utterances that are pronounced in the standard way. The upshot of this would be a lexicon inflated with a considerable proportion of effectively redundant variants, which are so similar to each other as to be virtually interchangeable. Variants that are capable of correcting the outlier pronunciations might not be added until a much later iteration, when the lexicon has already become bogged down with superfluous variants.

Another limitation of the prior decision rule is that it does not consider whether or not a lexicon entry actually benefits from the availability of an additional variant in the lexicon. It simply adds one variant in every iteration until there are no variants left in the pool of pronunciation candidates, regardless of the quality of the individual variant that is being added. Given enough iterations, all variants in the candidate pool will eventually be added to the lexicon, even if a particular candidate might be entirely unsuitable. In the hypothetical case that our data set contains a particular name that has an absolutely uniform pronunciation across all utterances, and there is a candidate variant which matches that standard pronunciation perfectly, then it would be meaningless and potentially counterproductive to add more variants to the recognition lexicon. This problem will make itself more felt for every iteration, as the variants that are being added for that name presumably become less and less well-matched to the standard pronunciation.

As we have seen in Section 4.3.1, lexical confusability is influenced by the size of the lexicon and the quality of the variants in the lexicon. It seems, then, that our present decision rule can have a negative effect on both of these factors, potentially causing an oversized lexicon containing substandard variants. In order to clarify the effect of these two limitations, Figure 5.5 provides a visual representation of the development of the lexicons constructed using the P2P, ME and MCE prior approaches. For each iteration (M), the respective columns represent the number of variants selected for inclusion into the lexicon by each of the selection algorithms.

The division of the columns into red and blue sections should be interpreted as follows. After the 10th iteration, we ran a recognition pass on all the utterances in our test set, using the end lexicons obtained from all 10 iterations of the various selection algorithms. The sections marked in blue in Figure 5.5 correspond to the successful variants: those variants that were effectively used by the recognizer and which resulted in an utterance being recognized correctly. The red sections correspond to unsuccessful variants: either these variants resulted in misrecognitions, or they were not used by the recognizer at all during this recognition pass with the respective end lexicons. The “blue” variants, then, are particularly valuable for recognition success, and in an optimal lexicon, we would want these variants to be selected during the earliest iterations. The “red” variants are not only likely to contain a substantial proportion of redundancy, but since some of these variants result in recognition errors, they also cause an increase of

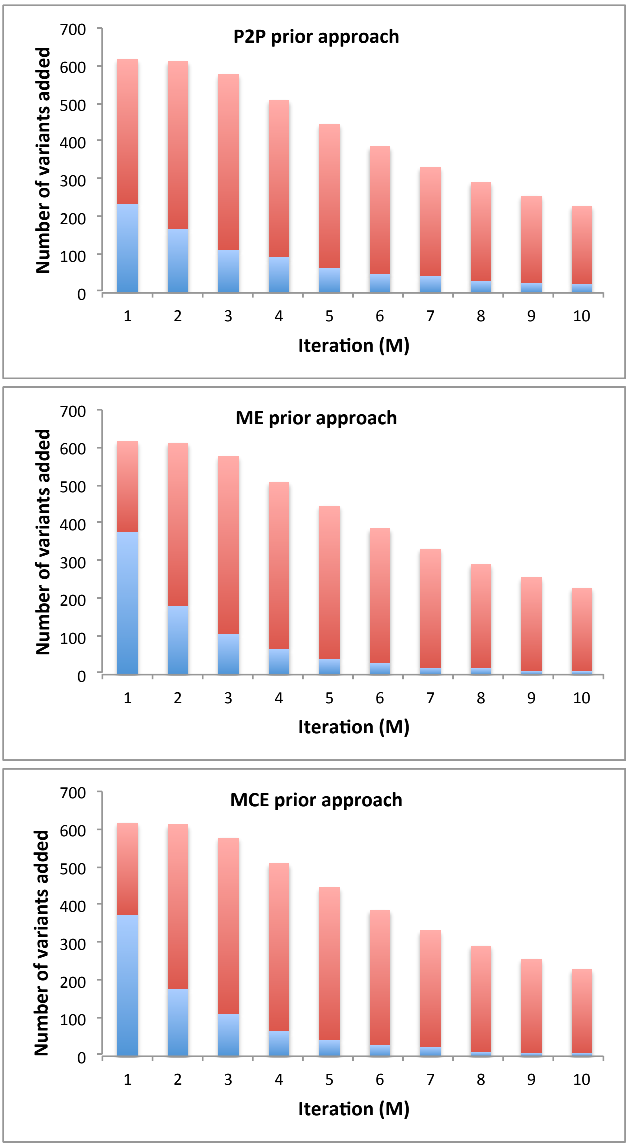

Figure 5.5: *The number of variants selected in each iteration (M) by the P2P, ME and MCE prior approaches.*

lexical confusion. It may be clear, then, that we would do well to limit the amount of red variants in our optimized lexicon. However, it should be noted that it would be wrong to assume that the blue variants are the only variants worthy of inclusion in the recognition lexicon: we must also be able to handle a certain amount of variation that may not be present in the particular set of utterances we have at our disposal while constructing the lexicon. Therefore, the lexicon should contain more variants than the ones used by any given recognition pass.

The first limitation described above is illustrated by the amount of blue variants selected in the later iterations of the selection algorithms. Presumably, these are variants used by the recognizer to correct outlier pronunciations for which no suitable variant was present in the earlier iterations. The second limitation is illustrated by the large proportions of red variants selected throughout the iterations: as long as there are candidate variables available, the selection algorithms select them. Although it is good to have a certain amount of variation in the recognition lexicon over and above what is observed in the available speech data, it may be clear that an optimized lexicon will need to minimize the amount of redundant variants.

One solution for these limitations is to design a decision rule that reflects the effect of a lexicon *change* rather than just the performance of the individual variants. Measuring the effect of adding a single variant to the recognition lexicon enables us to take the variants that are already in the lexicon into consideration when deciding whether or not to include a variant in the lexicon. It allows us to assess if a lexicon entry actually benefits from having this variant in the lexicon, and in this way it can prevent the inclusion of superfluous variants. Such a decision rule, however, requires an objective performance measure that is able to differentiate between the performance of a lexicon before and after the addition of a particular variant. In the following section, we will evaluate whether any of the variant selection criteria proposed in this chapter might be a suitable candidate for that role of performance measure.

5.3.3 Choosing the most suitable selection criterion

Not only did we find no substantial difference in performance between the prior and posterior decision rules, we also found that none of the proposed selection criteria significantly outperformed any other. We therefore considered the criteria more closely, inferring some main characteristics which might help us to decide on how to proceed in our further work.

The *No Prior Probability selection criterion (NPP)* selects the variants that on average align best acoustically with the name utterances in the training set. This selection criterion does not, however, consider whether or not the utterances were correctly recognized, nor does it consider the relative position of the correct hypothesis in the N-best list. As noted previously in this section, this approach also tends to select variants with a high number of competitors in the N-best list. These effects can in some cases result in poor variant selection, especially when the lexical confusion is high. A major disadvantage of the NPP selection criterion is that the log likelihood value is highly dependent on the specific utterances in the training set, making it unfeasible to compare lexicon performance for different data sets. For our purposes, however, the most crucial deficiency of an approach using log likelihood as a selection criterion is that it is not lexicon-dependent. By this we mean that the log likelihood score does not reflect the adverse effects that variants of other names that are already present in the lexicon may have on the recognition of a particular set of utterances. Comparing log likelihood scores before and after the addition of a new variant will therefore only tell us if the new variant aligns better acoustically with the training utterances than the variants already in the lexicon. Given that it is our objective to devise a procedure to optimize recognition lexicons by selecting a minimal amount of maximally effective variants, we require a selection criterion that takes the pre-existing context of the lexicon into account.

The *Maximum Entropy selection criterion (ME)* selects the variants that have the highest number of correctly recognized name utterances in the training set. If two variants result in an equal amount of correctly recognized utterances, the variant with the highest log likelihood score is chosen. In this way, the ME criterion incorporates information related to both the recognition performance and to the acoustic alignment of a variant with the training utterances. However, a selection based on the ME criterion is quite crude in the sense that it will always select variants proven to correct the highest number of recognition errors first, even if there might be other variants available that have a better overall acoustic match with the training data. This entails that a variant which has a perfect match with one single utterance will always be selected over variants that have a good (although not perfect) acoustic match with all utterances. The latter variant might well be more generally appropriate, and would in many cases be the better choice for lexical inclusion, even though it does not necessarily result in any utterances from a particular training set being recognized correctly. Incorporating additional features into the training of the ME model could reduce this effect to some extent and improve the performance of the ME criterion. However, the only features available to us during the development of this procedure were related to the recognizer's N-best list. Given that we make extensive use of the N-best list in the MCE criterion, such a redesign of the ME criterion would only make its selection behavior more similar to the MCE criterion. We therefore decided not to pursue this path further.

The *Minimum Classification Error criterion (MCE)* utilizes the N-best list proposed by the recognizer to select which pronunciation variants to include in the lexicon. The selection is done by choosing the variants that have, on average, the largest difference between the log likelihood score of the correct hypothesis and the average log likelihood of the competing hypotheses. In this way, the MCE criterion incorporates information about the recognition performance and the acoustic alignment of a certain variant, as well as the relative distance in performance between the correct hypothesis and all the other hypotheses in the N-best list. It therefore offers a more accurate prior probability than both the ME criterion and the acoustic likelihood. The MCE criterion is lexicon-dependent in the sense that it considers all the other variants in the lexicon. Additionally, as the MCE criterion constitutes a measure of recognition success on a zero-to-one continuum, it is more suitable for optimization than the ME criterion, which is a probability based on a discrete factor (viz. number of correctly recognized training utterances). For these reasons, the difference in MCE-score before and after a lexicon change can give an adequate reflection of the performance effects of the newly added variant.

All of these factors make us confident that the MCE criterion is the most suitable criterion to select the best performing pronunciation variants. In the remainder of this work, we will therefore abandon the NPP and ME approaches, and focus on investigating how we can exploit the potential of the MCE selection criterion more fully.

5.4 Conclusion

In this chapter we have evaluated two discriminative variant selection criteria using a prior and a posterior decision rule. The performance of each of the discriminative criteria was compared to that of a selection method using variant probabilities (P2P) and plain log likelihood scores (NPP) as a selection criterion. The experiments conducted in this chapter showed that both the discriminative selection criteria and the NPP criterion considerably outperformed the probability-based selection criterion when the number of variants in the lexicon was small. However, the experiments neither revealed which decision rule might be preferable, nor did they show any of the remaining selection criteria to perform better than the others.

Given that the experimental results were so inconclusive, we performed an in-depth analysis of the proposed decision rules and selection criteria. When investigating the decision rules, we found that the ME and MCE prior approaches overwhelmingly selected the same variants as the NPP approach. As the posterior decision rule only gives additional weight to the acoustic alignment, over and above the extent to which this is already built into the selection criteria, and the NPP approach selects variants purely on the basis of their acoustic alignment, it may be clear that applying the posterior decision rule to the ME and MCE criteria only serves to make their selection behavior resemble that of the NPP approach even more. This is illustrated well by the graphs of NPP, ME and MCE posterior in Figures 5.2 and 5.4, which are so similar as to be more or less indistinguishable: the selection behavior of the posterior selection approaches is all but identical.

In order to determine the most promising direction for our further research, we then reflected on the shortcomings of the proposed selection approaches, and found that lexical confusability was being augmented in the following two ways. Firstly, we observed that the selection approaches were unable to identify mutually compatible variants, and as a result they selected a large proportion of functionally equivalent and therefore effectively redundant variants. Secondly, there was no way to disqualify potentially harmful variants, and given a sufficient amount of iterations, all variants were included in the lexicon. The resulting lexicons were suboptimal both in terms of size and quality, in the sense that they contained an overly large amount of variants, some of which were of inferior quality.

We concluded that, in order to remedy these weaknesses, we might be better served with a method of measuring the effect of a lexicon change on recognition performance. By comparing the recognition success before and after the addition of a variant to the recognition lexicon, we could prevent the addition of redundant and inferior variants. We found the NPP and ME criterions to be less well-suited for this task than the MCE criterion.

In the next chapters, we aim to improve the prior selection rule. In this way, we hope to create a selection method that can generate a lexicon comprising variants that correct different recognition errors. Due to its ability to compare lexicons in a meaningful and optimizable way, the MCE score will be used to estimate the number of errors made by a lexicon before and after the addition of a particular variant.

Chapter 6

Selecting variants using an iterative approach

Although the MCE prior approach proposed in the previous chapter performed much better than the probability-based baseline approach, it was still not able to select variants correcting different types of recognition errors. This problem is caused by the fact that two similar pronunciation variants are likely to get similar MCE scores, making them equally ranked for lexical inclusion. To overcome this problem, we abandon the MCE prior approach and switch to a multi-pass iterative approach where variants are selected based on an estimate of the error rate *reduction* rather than simply using an estimate of the error rate. In this chapter we will describe the multi-pass iterative approach in detail. In the remainder of this dissertation the prior MCE approach will be referred to as the *single-pass* MCE approach. This work has also been presented in part in [3].

6.1 Decision rule and variant selection algorithm

In the iterative selection approach proposed in this chapter, a variant is only included in the recognition lexicon if it corrects errors that are left unhandled by the initial lexicon. To estimate the number of errors corrected by a particular variant, MCE scores calculated before and after the inclusion of the variant in the recognition lexicon, are compared. In the following section we define the variant selection criterion used in this selection algorithm. Section 6.1.2 clarifies some terminological conventions adhered to in this chapter. Section 6.1.3, then, describes the variant selection algorithm.

6.1.1 Decision rule

As in the previous chapter, we assume that we have a set of names, $\mathcal{W} = \{W_1, W_2, \ldots, W_K\ \}$, and that for some name $W_k \in \mathcal{W}$ a set of training utterances $\mathcal{X}_k = \{X_{k1}, X_{k2}, \ldots, X_{kN}\}$ and a set of candidate pronunciation variants $\mathcal{V}_k = \{V_{k1}, V_{k2}, \ldots, V_{kI}\}$ are available. Furthermore, we define an initial lexicon Λ_{init} where each name in $\mathcal{W}$ is represented by its single best performing variant. The goal, then, is to expand the lexicon with the variant $V_k^* \in \mathcal{V}_k$ that minimizes the risk of recognition errors. If $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{init})$ represents the *expected loss* of recognition accuracy for the training utterances in $\mathcal{X}_k$ when using the initial lexicon Λ_{init} and $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki})$ is the corresponding loss when using the initial lexicon extended with variant V_{ki}, then the variant selection criterion can be defined as

$$V_k^* = \underset{\mathcal{V}_k}{\operatorname{argmax}} \left(\mathcal{L}_k(\mathcal{X}_k; \Lambda_{init}) - \mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki}) \right). \tag{6.1}$$

The expected loss of an arbitrary model Λ is obtained as the accumulation of contributions $l_k(X_{kn}; \Lambda)$ emerging from the available training utterances $X_{kn} \in \mathcal{X}_k$ of name W_k:

$$\mathcal{L}_k(\mathcal{X}_k; \Lambda) = \frac{1}{N} \sum_{n=1}^{N} l_k(X_{kn}; \Lambda). \tag{6.2}$$

The loss function $l_k(X_{kn}; \Lambda)$ used in the experiments conducted in this chapter is the MCE loss function described in Equation (3.7) in Section 3.6.4.

6.1.2 Terminology for lexicon types

Before we describe the implementation of the selection algorithm, we must define a fixed terminology for the different types of lexicons that will be employed in this section. We will be using five different terms for our lexicons, dependent on which role they play and during which step of our selection procedure they are used. Two of these terms are preserved for lexicons that are essentially static, viz. they present the very begin and end state of our lexicon. For the former we will employ the term *start lexicon* while the latter will be called the *end lexicon.* To transform our start lexicon into our end lexicon, we use an iterative procedure, and in each iteration, we create three further lexicons, which we will call the *iteration-initial*, the *temporary* and the *iteration-final* lexicon respectively. These three lexicons are actively used throughout our procedure, and are therefore constantly changed.

With every iteration in this procedure we aim to augment our lexicon with the optimal available variants. The iteration-initial lexicon provides our starting point for an iteration. The temporary lexicon, then, is all but identical to the iteration-initial lexicon at any given point in the iteration The only exception is a single added pronunciation variant, namely variant V_{ki}. This temporary lexicon is used to evaluate the effect that this single lexicon change has on the recognition accuracy. In this way, we identify the optimal variants to be included in the iteration-final lexicon at the end of the iteration. Perhaps, unsurprisingly, this iteration-final lexicon will serve as the iteration-initial lexicon during the following iteration.

6.1.3 Variant selection algorithm

In this section we describe an iterative variant selection algorithm which in every iteration ($m = 1, \dots, M$) only selects variants which actually reduce the expected number of recognition errors. As a starting point we take a start lexicon Λ_s which comprises one English g2p transcription for each name. In the first iteration ($m = 1$) we then perform a procedure equivalent to that of the single-pass MCE approach:

1. create an iteration-initial lexicon lexicon by copying the variants contained in the start lexicon Λ_s, then, create an empty iteration-final lexicon

2. for each name $W_k \in \mathcal{W}$, extract the pronunciation candidate set $\mathcal{V}_k$ and perform the following steps for each candidate $V_{ki} \in \mathcal{V}_k$:

   (a) create a temporary lexicon, Λ_{ki}, by *replacing* the g2p transcription in the iteration-initial lexicon by the candidate pronunciation V_{ki}

   (b) perform a recognition pass on all the training utterances in $\mathcal{X}_k$ using this temporary lexicon and an isolated word grammar, and collect the most likely name hypotheses[1] proposed by the recognizer together with their likelihood scores

   (c) calculate the expected loss $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki})$ of the examined name according to Equation (6.2)

3. after performing the above procedure for all names in $\mathcal{W}$, select for each name the variant V_k^* yielding the lowest expected loss and add it to the iteration-final lexicon. If two or more variants have equal loss values, choose the variant with the highest average acoustic log likelihood score

[1] The maximum number of hypotheses in the N-best list was set to be 20.

The procedure in the subsequent iterations ($m = 2, ..., M$) is analogous, but with four differences: 1) we take as our iteration-initial lexicon the iteration-final lexicon emerging from the previous iteration; 2) when investigating variant V_{ki} of name W_k, a temporary lexicon is created by *adding* V_{ki} to the iteration-initial lexicon; 3) while in iteration $m = 1$ we selected variants with the lowest expected loss, in subsequent iterations we select variants causing the greatest *reduction* in expected loss, i.e. the variants maximizing Equation (6.1); 4) we no longer necessarily add a variant for *each* name: if none of the variants of name W_k cause a reduction in expected loss, we do not add any variants of W_k to the iteration-final lexicon.

This results in the following procedure performed for every iteration until the algorithm converges:

1. create an iteration-final and an iteration-initial lexicon $\Lambda_{iterinit}$, by collecting the variants contained in the iteration-final lexicon of the previous iteration

2. for each name $W_k \in \mathcal{W}$, extract the pronunciation candidate set $\mathcal{V}_k$ and perform the following steps for each candidate $V_{ki} \in \mathcal{V}_k$ not already contained in the iteration-initial lexicon:

   (a) create a temporary lexicon Λ_{ki} by *adding* the candidate pronunciation V_{ki} to the iteration-initial lexicon

   (b) perform a recognition pass on all the training utterances in $\mathcal{X}_k$, using this temporary lexicon and an isolated word grammar, and collect the most likely name hypotheses[2] proposed by the recognizer together with their likelihood scores

   (c) calculate the expected loss $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki})$ of the examined name according to Equation (6.2)[3]

3. after performing the above procedure for all names in $\mathcal{W}$, select for each name the variant V_k^* maximizing the decision rule in Equation (6.1) and add it to the iteration-final lexicon. If two or more variants have equal loss values, choose the variant with the highest average acoustic log likelihood score. If none of the variants can reduce the expected loss further, then no variant is added to the lexicon and no further attempts to add variants for that name are made

After a number of iterations, no more pronunciation variants that reduce the expected loss are available; the algorithm converges. In this way, the

[2] The maximum number of hypotheses in the N-best list was set to be 20.

[3] As of iteration $m = 2$, Λ_{init} is of course to be understood as $\Lambda_{iterinit}$.

iteration-final lexicon of the last iteration becomes our *end* lexicon.

6.2 Experiments and results

The multi-pass iterative approach was tested in two different environments: in a controlled setting of 617 names and in an open setting, using a much larger vocabulary of 16,045 names. As in the previous chapter, the three-fold cross-validation procedure described in Section 4.2.3, using data extracted from the NameDat database, was used in the two experiments. The results in this section are presented as the average NER of the three test sets and are given as a function of M, the maximum allowed number of variants per name. The average lexicon size, defined as the average number of pronunciations in the three final lexicons, is also presented for every M. The results are compared with the results of the probability-based baseline method (P2P) and the single-pass MCE approach given in Section 5.2. The multi-pass selection algorithm described in the previous section has also been evaluated on the Autonomata Spoken Name corpus [106], as was described in [3].

6.2.1 Testing in a controlled environment

First, the multi-pass iterative MCE approach was tested in a controlled environment, using an isolated-word grammar containing the 617 names of the NameDat corpus. The right-hand column of Table 6.1 shows the results of this experiment as a function of M. The two middle columns summarize the results for the P2P baseline approach and the MCE single-pass approach.

This table reveals several interesting properties of the multi-pass MCE approach. Firstly, it shows that the proposed approach significantly outperformed the baseline P2P method for the first three values of M. For higher values of M, the performance of the two approaches became more similar as the performance of the multi-pass approach converged (as did the size of the lexicons generated using this approach). Secondly, the table shows that the iterative approach needed to select no more than 2 variants per name to attain a performance equal to that of a lexicon comprising all available pronunciation candidates (12.5% NER). Finally, when comparing the results of the multi-pass MCE approach with that of the single-pass MCE approach, the table shows a slight performance increase in favor of the multi-pass approach. It should be noted however, that this performance increase is achieved using a recognition lexicon containing considerably fewer

Iteration	Baseline P2P		MCE Single-pass		MCE Multi-pass	
M	Size	NER	Size	NER	Size	NER
1	617	21.51%	617	13.53%	617	13.53%
2	1229	15.64%	1229	12.85%	1184	12.42%
3	1806	14,08%	1806	12.26%	1571	12.02%
4	2315	13.45%	2315	12.22%	1749	12.08%
5	2761	13.07%	2761	12.23%	1802	12.13%
6	3147	12.66%	3147	12.20%	1811	12.13%
7	3479	12.71%	3479	12.25%	1811	12.13%
8	3771	12.68%	3771	12.31%	1811	12.13%
9	4027	12.61%	4027	12.37%	1811	12.13%
10	4256	12.63%	4256	12.34%	1811	12.13%

Table 6.1: *Size and NER of lexicons created using different variant selection approaches in case of a small vocabulary.*

pronunciation variants. To demonstrate this effect, the performances of the three approaches are illustrated in Figure 6.1 as a function of the lexicon size on a logarithmic scale. In this figure, the results of the initial experiments conducted in Section 4.3.1 are also given for reference (marked with asterisks) in Figure 6.1. The figure illustrates that the multi-pass MCE approach generally seems to perform somewhat better than the single-pass MCE approach when using the same number of variants in the lexicon.

Including AV variants in the candidate set

Somewhat more disappointingly, the iterative MCE approach does not seem to perform as well as a lexicon comprising all auditorily verified transcriptions found in the training set (10.66% NER). One reason for this might be that the pronunciation candidates in our candidate set are automatically generated and of varying quality. To investigate this further, we repeated the experiment, now adding the auditorily verified transcriptions of the training set to the pool of pronunciation candidates. Since there is no accurate way of assigning probabilities to AV variants, the P2P baseline approach was not evaluated. Table 6.2 and Figure 6.2 illustrate the results of this experiment.

This table shows that the iterative MCE approach is able to achieve a slightly better performance than a lexicon containing all AV variants when using the same number of pronunciation variants in the lexicon (10.66% NER at $M = 3$). This means that, given a pool of pronunciation candidates containing some high quality and some lower quality variants, the

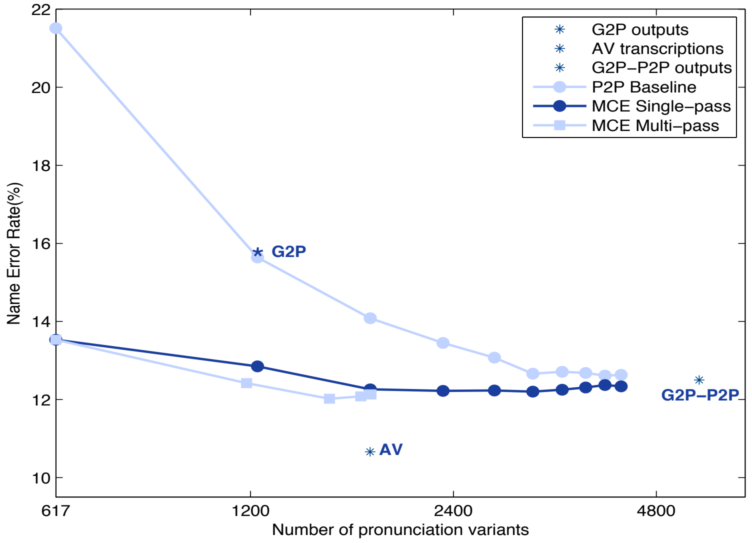

Figure 6.1: *NER of lexicons created using different variant selection approaches in case of a small vocabulary.*

Iteration	MCE Single-pass		MCE Multi-pass	
M	Size	NER	Size	NER
1	617	12.54%	617	12.54%
2	1234	11.58%	1203	10.80%
3	1851	11.15%	1615	10.07%
4	2452	10.65%	1841	9.97%
5	3029	10.37%	1979	9.90%
6	3578	10.34%	2026	9.87%
7	4084	10.27%	2031	9.87%
8	4528	10.23%	2031	9.87%
9	4918	10.19%	2032	9.87%
10	5270	10.18%	2032	9.87%

Table 6.2: *Size and NER in the small vocabulary case of lexicons created with different variant selection methods when having AV transcriptions in the candidate set.*

iterative MCE approach is in fact able to select some of the highest quality variants. Figure 6.2 illustrates that the performance gap between the single-pass approach and the multi-pass approach is effectively larger when having high quality variants in the candidate set. This implies that the multi-pass MCE approach potentially performs better when the quality of the variants in the candidate set is higher.

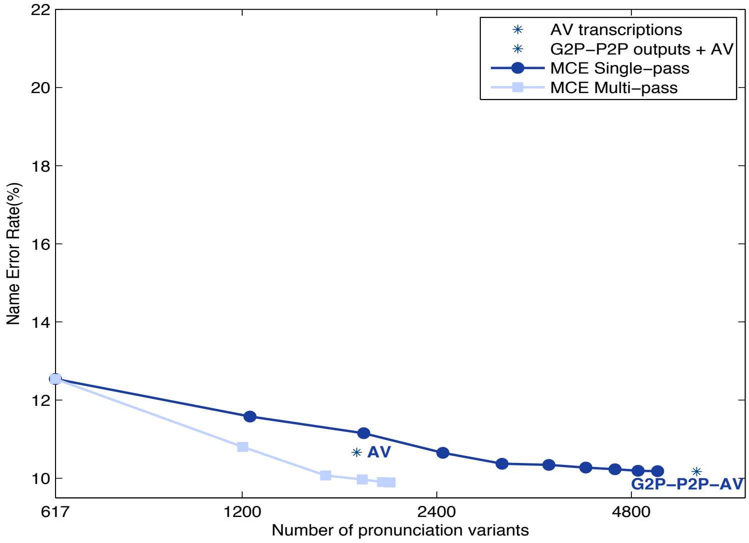

Figure 6.2: *NER in the small vocabulary case of lexicons created with different variant selection methods when having AV transcriptions in the candidate set.*

6.2.2 Testing in an open environment

The iterative MCE approach tested in a controlled setting gave a significant NER reduction compared to the baseline P2P method for the first three values of M. Nevertheless, the positive effect of reducing the lexicon size was not really apparent in this experiment. This is most likely an effect of the vocabulary size, since the vocabulary is too small to expose the increased lexical confusion caused by the inclusion of too many variants. Therefore, we repeated the recognition experiment, now using the extended vocabulary of 16,045 names. The procedure for this experiment was identical to that of the controlled experiment, but with one difference. In this experiment pronunciation variants for the filler names were added to the lexicon at the same rate as for the 617 names in our data set. By increasing the number of variants for all the names at the same rate, we aspired to gain some insight on the effect of the proposed selection algorithm when the vocabulary grew larger.

The procedure for adding variants to the filler names was as follows. In the first iteration, the recognition lexicon was augmented with a single variant for all the 617 names in the data set and a single variant for all the 15,428 filler names. Since no training data are available for the filler names, the most probable variant for each name was selected for lexicon inclusion. In the second iteration of the open experiment, the proposed algorithm selected 580 of the 617 names in the data set to be represented with a second variant in the lexicon. As a consequence, $\frac{580}{617} \cdot 15,428 = 14,503$ filler names "survived" the iteration and were also represented with a second variant. To decide which filler names should survive, we used the probability score of their second most probable variant. In other words, the 14,503 filler names which had the highest probability score for their second most probable variant survived the iteration. In the third iteration, 370 of the 617 names were selected by the proposed algorithm to be represented with a third variant. Hence, $\frac{370}{617} \cdot 15,428 = 9,252$ of the 14,503 filler names surviving the second iteration were represented with a third variant in the recognition lexicon. These were the 9,252 filler names surviving the second iteration with the highest probability score for their third most probable variant. This procedure was repeated until the proposed algorithm no longer could find any variants for the 617 names in the data set that could increase the recognition performance.

It may be clear that is an artificial setup, and we must bear this in mind when interpreting the results of this scheme. However, it seems a reasonable test of our hypothesis that a more careful variant selection could reduce the lexicon size considerably, which could be expected to lead to a decrease in lexical confusion. Table 6.3 shows the results for the iterative approach compared to the result of the probability-based baseline method and the single-pass MCE selection method as a function of M.

This table shows that the baseline (probability-based) selection method was again significantly outperformed by the multi-pass MCE-based selection method for all values of M. The proposed approach also performed significantly better than the MCE single-pass approach for $M = 2$. For M larger than 2, the iterative approach performed better than the single-pass approach as well, but with a somewhat smaller margin. Furthermore, the

Iteration	Baseline P2P		MCE Single-pass		MCE Multi-pass	
M	Size	NER	Size	NER	Size	NER
1	16045	34.32%	16045	22.78%	16045	22.78%
2	32067	26.13%	32067	21.78%	30755	20.49%
3	47929	24.11%	47929	21.06%	40672	20.59%
4	63250	23.30%	63250	21.09%	45258	20.70%
5	77563	22.86%	77563	21.25%	46567	20.73%
6	90574	22.47%	90574	21.43%	46828	20.73%
7	102157	22.30%	102151	21.57%	46838	20.73%
8	112208	22.23%	112208	21.79%	46839	20.73%
9	120930	22.40%	120930	22.10%	46839	20.73%
10	128284	22.48%	128284	22.25%	46839	20.73%

Table 6.3: *Size and NER of lexicons created by different variant selection approaches in case of a 16k vocabulary.*

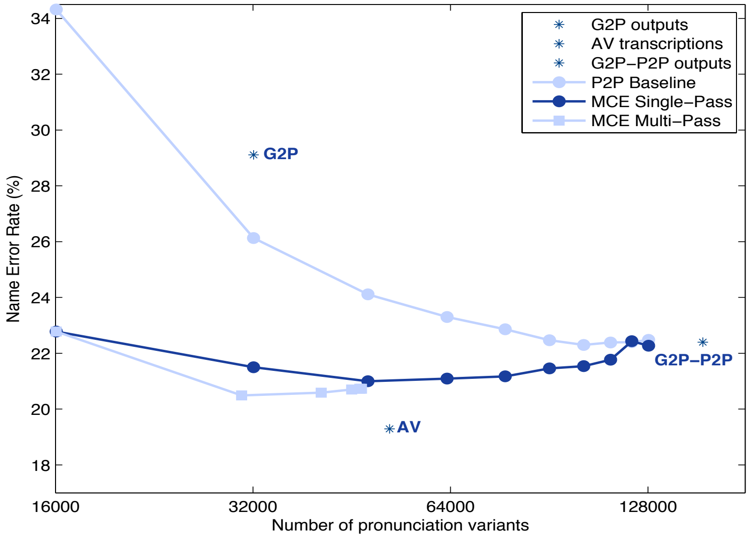

Figure 6.3: *NER as a function of lexicon size for lexicons created by different variant selection approaches in case of a 16k vocabulary.*

table shows that these performance gains were obtained using lexicons comprising considerably fewer variants compared to the corresponding lexicons generated with either the P2P approach or the MCE single-pass approach. This effect is illustrated in Figure 6.3 where the results of Table 6.3 are illustrated as a function of the lexicon size. In this figure, the AV lexicon contains the AV transcriptions for all the 617 names in the data set (on average 3 AV transcriptions per name) and the three most probable variants of the remaining 15,428 filler names. The results of the initial experiments conducted in Section 4.3.1 are also given for reference.

As this figure illustrates, the MCE multi-pass approach needed to select fewer than two variants per name on average to achieve a lower NER than every other lexicon used in this experiment[4], including the lexicon comprising all g2p and p2p variants (G2P-P2P 22.40% NER). Moreover, comparing this figure with Figure 6.1 shows that the performance gain achieved by the proposed approach in a large vocabulary setting is relatively larger than in the case of a small vocabulary.

Including AV variants in the candidate set

As in the controlled experiment, we wanted to investigate the effect of having high quality transcriptions in the pool of pronunciation candidates. To examine this, we added auditorily verified transcriptions to the pool and repeated the experiment. The results of this experiment are illustrated in Table 6.4 and in Figure 6.4. The AV lexicon in Figure 6.4 contains AV transcriptions for all the 617 names in the data set as well as the three most probable filler variants for the remaining 15,428 filler names.

Figure 6.4 shows that the multi-pass MCE approach, when tested in a large vocabulary setting, was able to surpass the performance of a lexicon containing all AV variants (AV 19.26% NER) as well as the performance of a lexicon comprising all AV variants and all g2p and p2p variants (G2P-P2P-AV 18.40% NER). Moreover, the best performing lexicon contained on average 3.1 variants per name which is only one third of the variants comprised in the G2P-P2P-AV lexicon. Another interesting observation is that the performance gain obtained in each iteration (until the procedure converged) was relatively larger compared to when using a candidate pool comprising only g2p and p2p transcriptions. It is also worth noticing that the proposed approach no longer seems to add variants that introduce new errors to the same extent as when using the g2p-p2p candidate pool. Both of these observations suggest that the proposed approach is in fact able to select the higher quality variants from the pool of transcription candidates.

[4] With the exception of the “cheating” experiment which employ a lexicon comprising transcriptions verified by an human expert.

Iteration	MCE Single-pass		MCE Multi-pass	
M	Size	NER	Size	NER
1	16045	20.94%	16045	20.94%
2	32068	19.41%	31267	18.09%
3	47941	18.55%	42996	17.52%
4	63301	18.41%	49506	17.44%
5	77671	18.17%	52038	17.50%
6	90741	18.23%	52620	17.55%
7	102386	18.06%	53479	17.55%
8	112496	18.18%	52700	17.55%
9	121273	18.29%	52700	17.55%
10	128676	18.39%	52700	17.55%

Table 6.4: *Size and NER in case of a 16k vocabulary for lexicons created by different variant selection approaches after adding AV transcriptions to the candidate pool.*

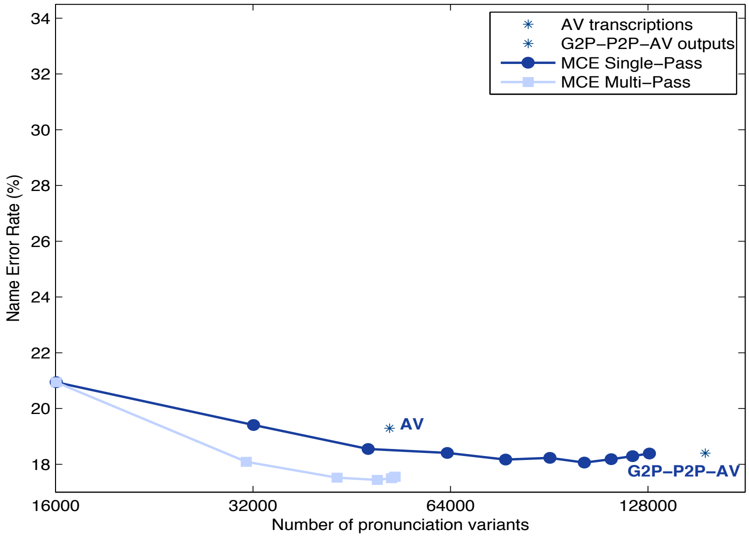

Figure 6.4: *NER in case of a 16k vocabulary for lexicons created by different variant selection approaches after adding AV transcriptions to the candidate pool.*

As in the small vocabulary experiment, the performance gap between the single-pass approach and the multi-pass approach effectively seems to be larger when having high quality variants in the candidate set. These observations confirm the results of the small vocabulary experiments and support our hypothesis that the multi-pass MCE approach is likely to perform better when the variants in the candidate set are of higher quality.

6.3 Discussion

6.3.1 Observations

The pronunciation variant selection algorithm proposed in this chapter aims to select only the variants that effectively reduce the error rate. The algorithm was implemented in such a way that the variants most likely to correct the most recognition errors were added to the lexicon first. By adding these variants early in the lexicon generation process, we could identify the point where no further gain could be achieved by adding more variants. In this way, we were able to prevent the addition of redundant variants to the lexicon. The upshot of this approach is a lexicon that is doubly optimized: we have at once attempted to minimize its size and maximize its performance. As discussed in Section 4.3.1, we have found that confusability can be counteracted to some degree by the presence of high quality transcription variants (which align better with the individual training utterances) in the lexicon. The algorithm presented in this chapter, then, aims to minimize confusability by selecting an optimally low number of strictly high quality pronunciation variants.

The multi-pass selection algorithm was evaluated in a controlled and in an open environment. In both environments, a significant performance increase was observed in favor of the proposed approach when compared to the P2P baseline approach, whereas a small performance increase was observed when the results were compared to that of the single-pass MCE approach. Adding auditorily verified transcriptions to the candidate pool produced some interesting results. Firstly, our multi-pass approach outperformed a lexicon consisting exclusively of the entire set of available AV transcriptions, which shows that the set of alternate variants generated by the p2p converter indeed contained some high quality variants, and that our approach was able to exploit these to some extent. Moreover, adding AV variants to the candidate pool caused the relatively small performance gap with the single-pass approach to widen, indicating that our proposed approach is in fact more effective when the candidate pool contains variants of high quality.

In the large vocabulary setting, the observed performance gain was relatively larger compared to the small vocabulary setting. This is most likely because the potential of the iterative approach to reduce lexical confusion is exploited more fully in a large vocabulary setting where confusability is naturally higher. The most interesting observation in the large vocabulary setting, however, was that the lexicon yielding the highest performance contained on average fewer than two variants per name. After this point, the proposed approach selected pronunciation variants that introduced more errors than they corrected, indicating some generalization problems in the large vocabulary case. Adding AV variants to the candidate pool, however, seemed to reduce these problems to some degree, as the best performing lexicon now contained 3.1 variants per name and the number of variants introducing more errors than they corrected was reduced. These effects can largely be attributed to the AV variants aligning better with the training utterances, increasing the distance between competitors in the N-best list, which again increases the selection criterion's ability to differentiate between variants. Looking back at the results of the same experiment in the small vocabulary case (Table 6.2) we observe that the same effect was in fact also present in the controlled setting. These observations indicate a correlation between the optimal number of variants in a lexicon and the quality of the variants within the lexicon in the proposed approach.

To illustrate the difference in behavior between the single-pass MCE approach and the multi-pass MCE approach, we made a visual representation of the development of the lexicons constructed using each of the selection approaches (Figure 6.5 and Figure 6.6). Each column in these figures represents the total number of variants selected for a specific value of M. The division of the columns into red and blue sections should be interpreted as in Section 5.3.2: the blue sections represent variants that triggered a correct recognition, the red sections represent variants that were either not used or used in misrecognitions. As in Figure 5.5, the numbers correspond to a recognition pass performed on all test utterances, using the end lexicon obtained after 10 iterations of the respective selection approaches.

When comparing Figure 6.5 with Figure 6.6, the first thing we notice is that the total number of variants selected in each iteration is considerably lower for the multi-pass MCE approach than for the single-pass approach.[5] Furthermore, the multi-pass approach selected no additional variants after the sixth iteration. As we have seen, this is not at the cost of a loss in recog-

[5] Of course with the exception of the first iteration, where the same initial variant per name was selected in both approaches.

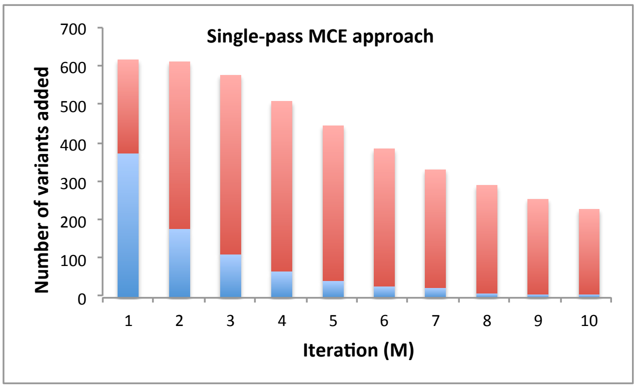

Figure 6.5: *Number of variants added in every iteration using the single-pass MCE approach.*

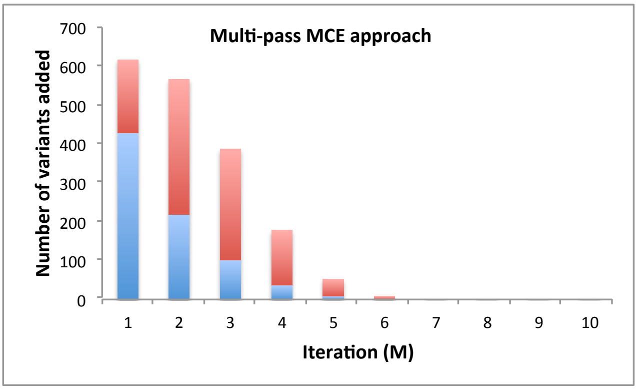

Figure 6.6: *Number of variants added in every iteration using the multi-pass MCE approach.*

nition performance: Table 6.1 shows that the multi-pass approach performs slightly better than the single-pass approach for all values of $M > 1$. This considerable reduction in lexicon size is obviously an effect of the stopping criterion introduced in the multi-pass approach, which prevents the addition of variants that fail to reduce the expected loss of recognition accuracy.

We also observe that the number of successful variants selected by the multi-pass algorithm in the early iterations is clearly higher than the corresponding number selected by the single-pass algorithm in the same iterations. This is particularly striking in the first iteration, where the number of successful variants is higher for the multi-pass approach, even though the lexicons employed in the two approaches are identical. This indicates that the stopping criterion introduced in the multi-pass approach does indeed prevent the inclusion of superfluous variants. In later iterations of the single-pass approach, a number of variants are selected which do result in recognition success, but it transpires that the same recognition success might have been achieved with variants that were already selected during the first iteration, and that these later variants are effectively redundant. In the multi-pass approach, these variants are not selected, and the recognizer successfully resorts to the more generic pronunciation variants added in the first iteration.

The same applies to the following iterations, although it is less straightforward to compare these, since the variants selected are only identical in the first iteration. We can conclude, however, that the variants selected by the multi-pass approach in later iterations are more complementary with the previously selected variants, whereas those selected in later iterations of the single-pass approach are more functionally equivalent to their predecessors. The upshot of this is a far more compact lexicon, where the ratio of successful variants is far higher: 44% of the variants selected by the multi-pass approach are successful variants, versus only 20% in the single-pass approach.

These observations show that the proposed approach succeeds to some extent in preventing the inclusion of redundant variants in the recognition lexicon, with a marginal gain in recognition performance. In larger vocabularies, where lexical confusability inevitably plays a more crucial part, this can be expected to lead to a more pronounced performance increase.

6.3.2 Limitations of the multi-pass iterative MCE approach

One limitation of the proposed method is that pronunciations that are not represented in the training set will not be well covered by the resulting lexicon. This is especially the case for names which have a uniform set of pronunciations in the training set. In these cases, a more relaxed selection and stopping criterion can be beneficial to represent a larger number of outlier pronunciations in the lexicon. At the same time, however, we observe from Figure 6.6 that a large number of the variants selected by the multi-pass algorithm are not actually being used during testing. As pointed out in Section 5.3.2, it is true that there must be some surplus in order to cover unseen variation, but the amount of unused variants still seems to be unnecessarily large. Paradoxically, this appears to indicate that we should impose a stricter selection and stopping criterion in order to obtain a more compact lexicon. The linear contradiction of these two observations suggests that our selection and stopping criterion may need to be revised altogether.

Another weakness of the proposed approach is its “breadth-first” strategy of adding pronunciation variants to the lexicon. This amounts to adding one or no pronunciation variant for every name before recalculating the MCE scores and continuing to the next iteration. Such a strategy has several limitations. First of all, it does not fully exploit the discriminative power of the MCE score, since the scores are recalculated only after adding several new variants to the lexicon. Ideally, new MCE scores should be calculated after every lexicon change. Secondly, this approach treats all variants selected in a particular iteration as equally important. Consider for example the names “Gloucestershire” and “London”. The former is obviously more prone to pronunciation variation than the latter, and will therefore need to be represented with more pronunciation variants in the lexicon. In most cases therefore, it would be more beneficial to add several pronunciation variants of the name “Gloucestershire” to the lexicon before adding one extra variant of the name “London”. The argument for this is the same as for the selection approach proposed in this section. Prioritizing the selection of variants with the highest potential to correct recognition errors would enable us to implement a more precise stopping criterion, which in its turn can prevent the addition of redundant variants to the lexicon. We might therefore do better to abandon the “breadth-first” selection strategy and adopt a “best-first” strategy, where at every step the best available variant is added to the lexicon, regardless of the name it represents and how many variants already represent this name in the lexicon. In this way, we might be able to further reduce the lexicon size by selecting the variants that yield the largest global reduction in error rate first. We will investigate this hypothesis further in the next chapter.

Another acute problem of the proposed approach is its high computational load. In each iteration, all training utterances of each name must be decoded as many times as there are available pronunciation candidates of that name. Therefore, further optimizations of the method are necessary in order to make it suitable for very large vocabularies.

6.4 Conclusion

In this chapter we have proposed a pronunciation variation modeling approach that proved effective for the recognition of English proper names spoken by native Norwegian speakers. The iterative nature of the proposed approach ensured that only the variants expected to decrease the error rate were included in the final lexicon. Testing this approach in a controlled environment confirmed that selecting variants in this manner substantially reduces both the error rate and the required number of variants per name compared to a probability-based baseline selection method. When comparing the proposed iterative approach to the single-pass approach described in the previous chapter, we also observed some performance gain, given a lexicon of considerably smaller size. The proposed approach proved to be particularly effective when the quality of the variants in the pronunciation candidate set was high. The positive effect of reducing the lexicon size was illustrated in an experiment conducted in an open environment, where lexical confusability was more prominent than in a small vocabulary setting. When analyzing the behavior of the proposed approach, it was observed that further improvements of the iterative MCE selection approach might be achieved through:

- improving the coverage for unseen variation,
- removing an even larger amount of redundant variants,
- adopting a “best-first” selection strategy,
- reducing the computational load.

In the next chapter, we will investigate these limitations further and propose new solutions for the problems identified in this chapter.

Chapter 7

Selecting variants using a tree search approach

In the previous chapter, we proposed an efficient variant selection approach based on the reduction in error rate observed after the addition of a particular variant. Although this approach constituted a significant improvement over the probability-based baseline, we argued that it was still suboptimal due to its inability to foresee which lexicon entries would benefit the most from having additional pronunciation variants in the lexicon. Moreover, since this approach employed a “breadth-first” selection strategy, adding one pronunciation variant to each lexicon entry in every iteration until a stopping criterion was met, it did not necessarily result in the most accurate and compact lexicon.

In this chapter, we aim to deal with these shortcomings by recasting the pronunciation variant selection task as a “best-first” tree search problem. In this approach, the optimal recognition lexicon corresponds with the optimal path through a search tree. To guide the search algorithm, we define a discriminative evaluation function which is based on estimates of the number of recognition errors before and after a lexicon change. As before, the error rate for a given lexicon is estimated using the Minimum Classification Error framework. This work was also presented in [4].

7.1 Evaluation function and selection algorithm

In this chapter we will consider the pronunciation variant selection problem as a tree search problem, where the goal is to find the optimal path through a predefined tree structure by using a discriminative evaluation function. In order to make the chapter as easily comprehensible as possible, the following subsection gives an intuitive clarification of the type of tree structure we aspire to construct. In Section 7.1.2, we define the evaluation function used by the tree search variant selection algorithm. In Section 7.1.3, then, we describe the variant selection algorithm itself.

7.1.1 Conceptualizing an optimal lexicon as a tree structure

As in our previous experiments, we assume that we have a set of names $\mathcal{W} = \{W_1, W_2, \ldots, W_K\}$ and that for each name $W_k \in \mathcal{W}$ we have a set of training utterances $\mathcal{X}_k = \{X_{k1}, X_{k2}, \ldots, X_{kN}\}$ and a set of candidate pronunciation variants $\mathcal{V}_k = \{V_{k1}, V_{k2}, \ldots, V_{kI}\}$. We now define a pool of all candidate variants $\mathcal{V}$ as the union of all sets of pronunciation variants: $\mathcal{V} = \{\mathcal{V}_1 \cup \mathcal{V}_2 \cup \ldots \cup \mathcal{V}_K\}$. From this candidate pool $\mathcal{V}$ we attempt to select those variants that together make up the optimized recognition lexicon Λ_{opt}.

An example of such an optimized lexicon Λ_{opt} might then be visualized as the tree structure in Figure 7.1. Each branch of this tree corresponds with one name $W_k \in \mathcal{W}$ and contains a set of nodes $\mathcal{N}_k = \{n_{k1}, n_{k2}, \ldots, n_{kL}\}$, where L is a predefined threshold denoting the maximum number of variants allowed per name. The filled nodes of a given branch represent the selected pronunciation variants for the name with which the branch corresponds. The branches do not have an equal amount of filled nodes, because not all names benefit equally from the inclusion of multiple pronunciation variants. Names that are pronounced in a fairly uniform way by different speakers should be represented by fewer variants than names that can be pronounced in a wide variety of ways, and consequently have fewer filled nodes.

The gray nodes, which are the lowest non-empty nodes on each branch, constitute the *goal nodes* for the corresponding names. A name W_k is said to have reached a goal node when at least one of the following four stopping criteria is met: (1) there are no more available pronunciation variants of W_k in the candidate pool $\mathcal{V}$; (2) there are L variants of W_k in the recognition lexicon, which means that W_k is represented by the maximum allowed number of variants; (3) all the utterances of W_k in the training set $\mathcal{X}_k$ are correctly recognized; (4) none of the remaining variants of W_k in the candidate pool $\mathcal{V}$ is able to correct any more recognition errors of the training utterances in $\mathcal{X}_k$.

The order in which we build up the lexicon is governed by the potential improvement in overall recognition performance offered by the variants that are still available in the candidate pool $\mathcal{V}$. This affects the population of the nodes in the lexicon in two ways. Firstly, and quite intuitively, it means that the consecutive nodes within a branch are ordered by their beneficial impact on recognition success. Secondly, and perhaps somewhat

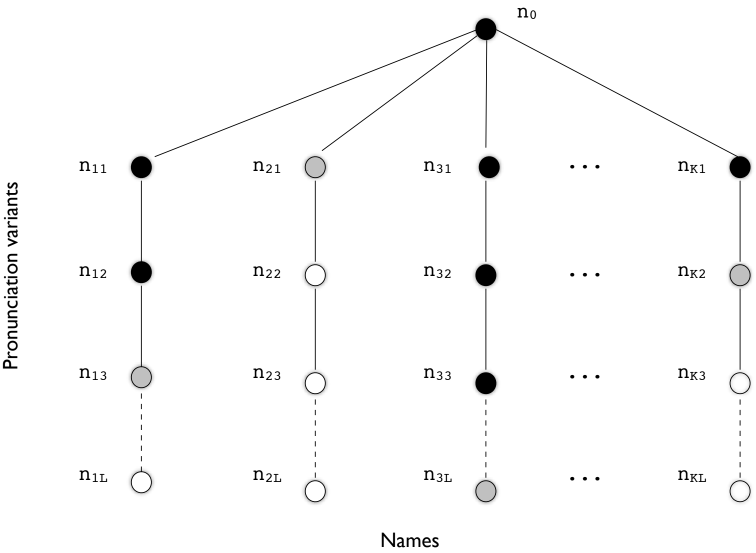

Figure 7.1: *An example of an optimized lexicon conceptualized as a tree structure.*

less self-evidently, it means that the order in which additions are made to the respective branches of the lexicon is not fixed. We should not add one pronunciation variant to each name in sequence, but rather prioritize the name which could have the greatest possible effect on overall recognition success. That is to say, at any given point during the construction of the lexicon, there may be a difference in the number of filled nodes per branch, as different names may be represented by a different number of variants. The upshot of this is that if Λ_m is an instance of such an optimized lexicon containing m populated nodes, Λ_m will consist of the optimal set of pronunciation variants for any given value of m. In practice, this means that if we require our lexicon to be of a specific size the lexicon generated by the tree search approach will inherently comprise the optimal set of variants for that lexicon size. Our objective in this chapter, then, is to design a variant selection approach that matches these expectations of a lexicon optimization method.

7.1.2 The evaluation function

Our main goal is to compile a recognition lexicon Λ_m in such a way that both accuracy (in terms of recognition performance) and compactness (in terms of lexicon size) are optimized. To achieve this, our search tree is incrementally populated with the *most promising* node n_{kl}. We define this node as the pronunciation variant that maximizes the evaluation function

$$\hat{f}_m(n_{kl}) = \alpha_l \cdot \hat{g}_m(n_{kl}) + \hat{h}_m(n_{kl}). \qquad (7.1)$$

This function consists of two elements: a *correction potential* factor $\hat{g}_m(n_{kl})$ and an *error potential* factor $\hat{h}_m(n_{kl})$. The correction potential factor gives an estimate of the number of errors corrected for name W_k after adding the variant of node n_{kl} to the recognition lexicon Λ_m.[1] The error potential factor gives an estimate of the number of errors likely to be corrected by the successors of node n_{kl}.

The scaling factor α_l, is a heuristic function designed to modify the relative weight between the error potential factor and the correction potential factor in the evaluation function. In this work, we aim to use this heuristic function to emphasize the importance of the correction potential factor at shallow tree depths in order to give the search more of a breadth-first character for the nodes higher in the search tree. The reasoning behind this is to prevent the search algorithm of overemphasizing branches with a high error potential factor high in the tree (i.e. names with poor recognition performance when represented with one or two variants in the lexicon). Obviously, there are a variety of functions which can be used to give more weight to the nodes higher in the tree. In the work described in this chapter, we opted for a simple function, $\alpha_l = (L - l)$, to serve as our scaling factor. It should be noted that no attempt has been made to optimize this function and it is therefore possible that there are functions better suited for this purpose.

Figure 7.2 illustrates the correction potential factor and the error potential factor within the search tree when node n_{K2} is being evaluated. As in the previous experiments, the number of errors corrected by an arbitrary lexicon Λ can be estimated by means of the expected loss of recognition accuracy. This expected loss is the normalized sum of the loss values calculated by the loss function $l_k(X_{kn}; \Lambda)$ obtained after a recognition pass of

[1] This makes the evaluation function dependent on the m nodes populated so far.

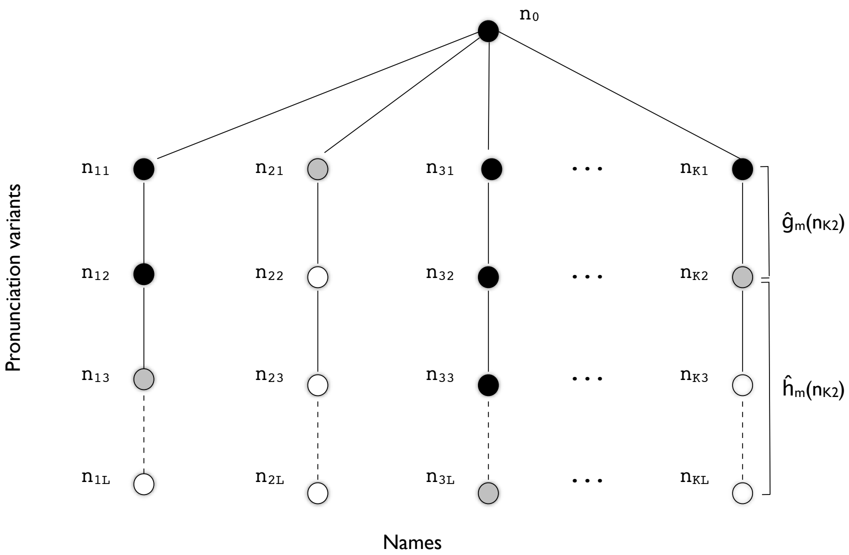

Figure 7.2: *The tree structure for lexicon expansion with $\hat{g}_m(n)$ and $\hat{h}_m(n)$.*

the training utterances $X_{kn} \in \mathcal{X}_k$ of name W_k:

$$\mathcal{L}_k(\mathcal{X}_k; \Lambda) = \frac{1}{N} \sum_{n=1}^{N} l_k(X_{kn}; \Lambda) \tag{7.2}$$

where $l_k(X_{kn}; \Lambda)$ is the MCE loss function described in Equation (3.7).

To define the correction potential factor of node n_{kl} we first need to create a temporary lexicon Λ_{kl} consisting of the variants comprised in Λ_m with the addition of a candidate variant V_{ki} of node n_{kl}. If $\mathcal{L}_k(\mathcal{X}_k; \Lambda_m)$ is the expected loss of recognition accuracy when using lexicon Λ_m and $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{kl})$ is the corresponding loss when using the temporary lexicon Λ_{kl}, we define the following correction potential factor:

$$\hat{g}_m(n_{kl}) = \mathcal{L}_k(\mathcal{X}_k; \Lambda_m) - \mathcal{L}_k(\mathcal{X}_k; \Lambda_{kl}). \tag{7.3}$$

The error potential factor for node n_{kl} is estimated using the same temporary lexicon Λ_{kl}. Since names with a high error potential factor are likely to benefit the most from having an additional variant in the recognition lexicon, we aim to select variants with a high error potential factor as well as a high correction potential factor. The error potential factor is estimated as the expected loss of recognition accuracy of lexicon Λ_{kl}

$$\hat{h}_m(n_{kl}) = \mathcal{L}_k(\mathcal{X}_k; \Lambda_{kl}). \tag{7.4}$$

When populating the search tree, the evaluation function $\hat{f}_m(n_{kl})$ is calculated for every non-populated node in the tree (i.e. for all variants in $\mathcal{V}$ not already in Λ_m) before the node with the highest $\hat{f}_m$-value is populated and its variant added to the recognition lexicon Λ_m.

Approximating the evaluation function

It may be clear that continuously calculating the evaluation function for all non-populated nodes in the manner described above is computationally very expensive. Due to practical constraints, we decided to approximate the calculation of the evaluation function $\hat{f}_m$ in order to reduce the computational load.

To approximate the calculation of the evaluation function, let us first assume that node n_{kl} has just been populated by some variant V_{ki} and that this variant has been added to the recognition lexicon Λ_m as the l-th variant of name W_k. To determine which node to populate next, the evaluation function should in principle be calculated for all variants in $\mathcal{V}$ not already in Λ_m, which entails performing a large number of relatively time-consuming computations. To reduce the number of calculations, we make use of the following observation: since the evaluation function is based on the expected loss of recognition accuracy, the only values of $\hat{f}_m$ which are likely to change considerably after the addition of variant V_{ki} to Λ_m are values calculated for other variants of name W_k. Thus, if we calculate new values only for the variants in $\mathcal{V}_k$, and use previously calculated values for the variants not in $\mathcal{V}_k$, we can reduce the number of calculations considerably.

When calculating the evaluation factor using this approximation, we can employ the same correction potential factor and error potential factor as described in Equation (7.3) and Equation (7.4). The only difference is that we replace the recognition lexicon Λ_m with a lexicon $\Lambda_{k(l-1)}$, comprising the variants contained in the recognition lexicon when populating the *previous node of name* W_k (node $n_{k(l-1)}$). Then, if Λ_{kl} is a temporary lexicon containing the variant V_{ki} and the variants in $\Lambda_{k(l-1)}$ we can define the following approximation to the correction potential factor and the error potential factor:

$$\hat{g}_{kl}(n_{kl}) \approx \tilde{g}_{kl}(n_{kl}) = \mathcal{L}_k(\mathcal{X}_k; \Lambda_{k(l-1)}) - \mathcal{L}_k(\mathcal{X}_k; \Lambda_{kl}) \tag{7.5}$$

$$\hat{h}_{kl}(n_{kl}) \approx \tilde{h}_{kl}(n_{kl}) = \mathcal{L}_k(\mathcal{X}_k; \Lambda_{kl}). \tag{7.6}$$

This results in the following approximation of the evaluation function:

$$\hat{f}_{kl}(n_{kl}) \approx \tilde{f}_{kl}(n_{kl}) = \alpha_l \cdot \tilde{g}_{kl}(n_{kl}) + \tilde{h}_{kl}(n_{kl}). \tag{7.7}$$

7.1.3 The variant selection algorithm

The discriminative tree search algorithm was implemented as follows:

1. calculate the start lexicon Λ_m using the same procedure as in the first iteration of the MCE approaches (described in Section 5.1.2). For each name $W_k \in \mathcal{W}$, populate the first node n_{k1} of the corresponding branch with the variant that minimizes Equation 7.2.

2. initialize the selection algorithm by performing the following steps for all names $W_k \in \mathcal{W}$:

   (a) if n_{k1} is a goal node, skip to the next name. A node is defined as a goal node if one of the following criteria is met: $\tilde{h}_{k1}(n_{k1}) = 0$ or $L = 1$ or $\mathcal{V}_k = \{V_{k1}\}$

   (b) if n_{k1} is not a goal node, proceed to the following node in the branch for W_k by incrementing l

   (c) perform the following steps for every pronunciation candidate $V_{ki} \in \mathcal{V}_k$ not already in Λ_m:

      i. create a temporary lexicon Λ_{ki} by adding the candidate pronunciation to the recognition lexicon Λ_m

      ii. calculate the expected loss $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki})$ for every pronunciation candidate by performing a recognition pass on all the training utterances $\mathcal{X}_k$ of name W_k, using Λ_{ki} and an isolated word grammar

   (d) find the pronunciation variant in $\mathcal{V}_k$ with the highest $\tilde{g}_{k2}$ value and put the node in the stack

3. if the stack is empty, exit the algorithm

4. remove the node with the highest $\tilde{f}_{kl}$ value from the stack, populate node n_{kl} with its variant and add the variant to Λ_m

5. perform the following steps for name W_k:

(a) if n_{kl} is a goal node, go to step 3. A node is defined as a goal node if one of the following criteria is met: $\tilde{h}_{kl}(n_{kl}) = 0$ or $\tilde{g}_{kl}(n_{kl}) = 0$ or $L = l$ or Λ_m contains all variants $V_{ki} \in \mathcal{V}_k$

(b) if n_{kl} is not a goal node, proceed to the following node in the branch for W_k by incrementing l

(c) perform the following steps for every pronunciation candidate $V_{ki} \in \mathcal{V}_k$ not already in Λ_m:

   i. create a temporary lexicon Λ_{ki} by adding the candidate pronunciation to the recognition lexicon Λ_m

   ii. calculate the expected loss $\mathcal{L}_k(\mathcal{X}_k; \Lambda_{ki})$ for every pronunciation candidate by performing a recognition pass on all the training utterances $\mathcal{X}_k$ of name W_k, using this temporary lexicon and an isolated word grammar

(d) find the pronunciation variant in $\mathcal{V}_k$ with the highest $\tilde{g}_{kl}$ value and put the node in the stack

6. go to step 3

It may be evident that step 2 and step 5 in this algorithm are largely identical. There are, however, two crucial differences. Firstly, since the goal of step 2 is to initialize the algorithm by filling the stack, this step is repeated for all names $W_k \in \mathcal{W}$. Step 5, on the other hand, is only performed for the particular name for which an additional node was populated in step 4. As a consequence, the value of l becomes name-dependent as of step 5b, since it is incremented for that name exclusively. Secondly, it should be noted that step 2 does not include $\tilde{g}_{kl}(n_{kl}) = 0$ as a stopping criterion. The reason for this is that $\tilde{g}_{k1}(n_{k1})$ cannot be meaningfully calculated. As defined in Equation 7.3, the correction potential factor compares the performance of a temporarily expanded recognition lexicon with the previous state of the lexicon, but given that the lexicon was empty previous to the population of the first layer of nodes, this comparison is impossible.

Step-by-step illustration of a hypothetical algorithm run

In order to further elucidate the procedure, Figure 7.3 and Figure 7.4 present a step-by-step illustration of the subsequent states of our recognition lexicon during a hypothetical run of the selection algorithm. Figure 7.3 shows the initial states of the recognition lexicon, before the iterative procedure described in step 2 above is initialized. Figure 7.4 shows the first series of changes in a hypothetical example run of the actual tree search procedure, where one variant is added to the lexicon at a time. For every change in state of the lexicon, we have marked the individual elements that have undergone an alteration in red.

State 0 in Figure 7.3 shows the recognition lexicon *before* the selection procedure is started; that is before step 1 of our selection algorithm. As the recognition lexicon is empty at this point, it can be visualized as a tree structure consisting exclusively of empty nodes. There is one branch for each name $W_k \in \mathcal{W}$ containing L virtual nodes. Our objective, then, is to populate these nodes in an optimally efficient way. State 1 corresponds with the start lexicon compiled in step 1 of our selection algorithm. In this state, the recognition lexicon is identical to the lexicons for $M = 1$ in the MCE single-pass and multi-pass approaches: for each name $W_k \in \mathcal{W}$, we select the variant with the lowest MCE loss value. These variants provide the base layer of our start lexicon that is needed for our evaluation function to work upon. The stack is marked as empty during the two initial states illustrated in Figure 7.3, as it does not come into play before the iterative phase of the algorithm is initialized.

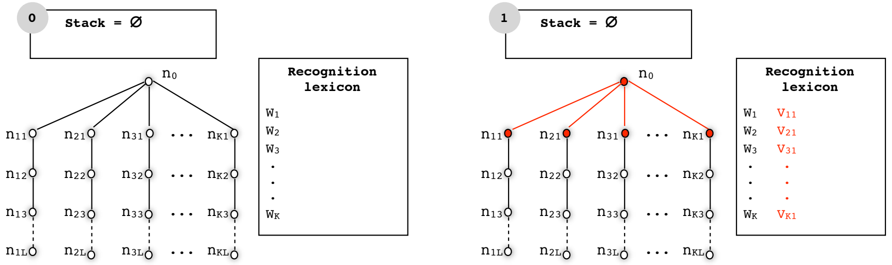

Figure 7.3: *The initial states of the tree search algorithm, prior to the iterative phase.*

From state 2 onwards, we see the iterative tree search algorithm proper at work.[2] This is visualized in Figure 7.4. State 2 illustrates the first use of the stack [step 2]. For all names that have not yet reached a goal node, the variant with the highest $\tilde{f}_{kl}$ value is added to the stack. The stack is sorted by the $\tilde{f}_{kl}$ values of its members, which means that in our hypothetical example, variant V_{13} of name W_1 is the most promising variant we have available in our candidate pool. In state 3 this variant is used to populate its corresponding node n_{12}: variant V_{13} is now removed from the stack and

[2] Where appropriate, the corresponding step in the selection algorithm described in Section 7.1.3 is henceforth noted between brackets.

added to the recognition lexicon [step 4].

State 4 shows the next change performed by the tree search selection algorithm: a new variant V_{16} of name W_1 is added to the stack, which has a lower $\tilde{f}_{kl}$ value than variant V_{34} of name W_3 and variant V_{22} of name W_2 [step 5]. In state 5, node n_{32} is populated with variant V_{34} [step 4], and in state 6, a new variant V_{37} of name W_3 is added to the stack [step 5]. As this is the variant with the highest $\tilde{f}_{kl}$ value, it is immediately added to node n_{33} in state 7 [step 4].

In state 8, no new pronunciation variant of name W_3 is added to the stack, which indicates that name W_3 has now reached a goal node [step 5a]. Given that we have not yet reached L, the maximum number of variants allowed per name, and that there are still pronunciation variants for name W_3 available in the candidate pool, this must be either because $\tilde{h}_{kl}(n_{33}) = 0$ or because $\tilde{g}_{kl}(n_{33}) = 0$. In the case that $\tilde{h}_{kl}(n_{33}) = 0$, we may conclude that all recognition errors for name W_3 have already been corrected by the variants in the recognition lexicon. If $\tilde{g}_{kl}(n_{33}) = 0$, this indicates that none of the remaining variants of name W_3 in the candidate pool can be expected to correct any more recognition errors. To emphasize that W_3 has reached a goal node, we have marked node n_{33} in gray in state 9 of Figure 7.4. It may be clear that the algorithm does not stop here, since there are still variants of other names in the stack, but this should suffice to illustrate the procedure.

7.2 Experiments and results

In this section, we will compare the performance of the discriminative tree search algorithm with the performance of the discriminative single-pass and multi-pass selection approaches described in Chapter 5 and in Chapter 6 respectively. The experiments described in Section 7.2.1 were conducted in a controlled environment, using a vocabulary of 617 names. Section 7.2.2 gives the results of our experiments in an open environment, using a vocabulary of 16,045 names. As elsewhere, the three-fold cross validation procedure described in Section 4.2.3 was used in both experiments. The results of these experiments are presented as the average name error rate (NER) of the three test sets and are given as a function of the recognition lexicon size. After a systematic trial and error procedure in both a controlled and open environment, the maximum number of variants per name, L, was set to 4.

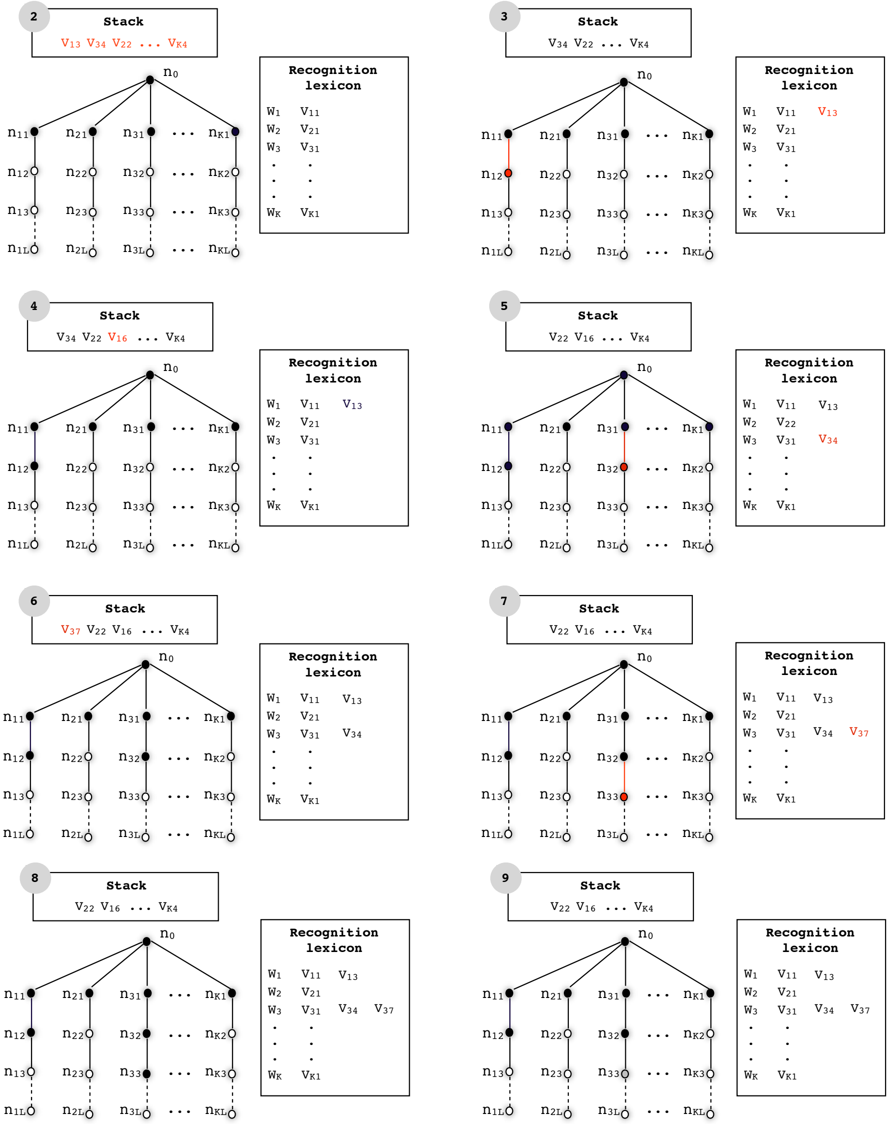

Figure 7.4: *The first sequence of states of the tree search algorithm during a hypothetical example of the iterative phase.*

7.2.1 Testing in a controlled environment

The proposed tree search algorithm was first tested in a controlled environment, using a single-word grammar containing only the 617 names of the NameDat corpus. The results of these experiments are given in Table 7.1. It may be noted that the format in which these results are presented is somewhat different from previous chapters. As the previously proposed selection algorithms assess the addition of a variant to the lexicon for each name in succession, it is natural to evaluate the recognition performance of the resulting lexicons for the maximum number of variants per name M. The tree search approach, on the other hand, simply adds the most promising variant from the candidate pool, regardless of the corresponding name. Consequently, the recognition lexicon emerging from this selection algorithm must be evaluated per added variant, rather than per iteration over all names in the vocabulary. In the controlled experiments described in this section, the performance of the tree search selection approach is evaluated per 100 additions to the recognition lexicon.

Tree-search approach	
Size	NER
617	13.53%
717	12.58%
817	12.04%
917	11.88%
1017	11.80%
1117	12.07%
1217	11.96%
1317	12.00%
1417	12.05%

Table 7.1: *Size and NER of a lexicon created with the tree-search variant selection approach evaluated in a controlled environment.*

To compare these results with those of previous experiments, they are illustrated in Figure 7.5 together with the results of the probability-based baseline approach (P2P Baseline), the single-pass MCE approach (MCE Single-pass) and the multi-pass MCE approach (MCE Multi-pass). For reference, the results of the threshold experiments conducted in Section 4.3 are marked with asterisks. The figure shows the NER as a function of the lexicon size on a logarithmic scale.

Figure 7.5 clearly shows that the tree search selection approach pre-

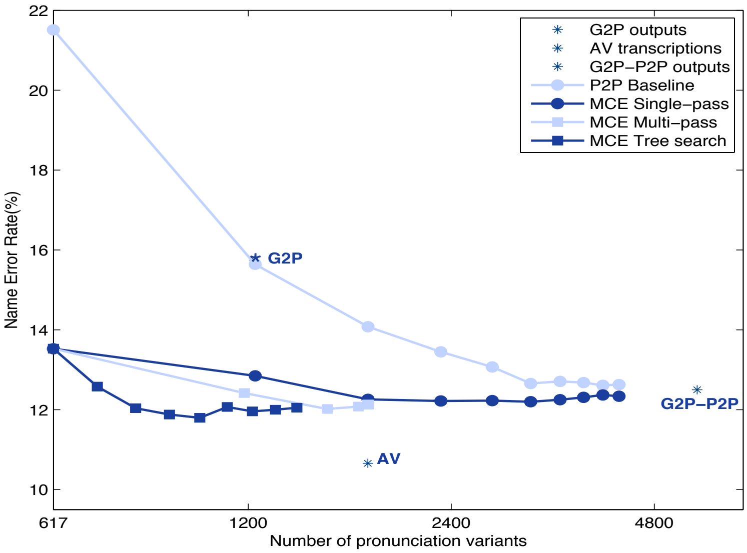

Figure 7.5: *The results for the single-pass MCE, multi-pass MCE and tree-search approach in a controlled environment.*

sented in this chapter has considerable success in constructing a lexicon yielding strong recognition performance for a low number of variants per name. Comparing the performance of the tree search variant selection approach with the full G2P-P2P lexicon containing all available variants in the candidate pool, we observe a decrease in Name Error Rate while using fewer than one fifth of the variants. Remarkably, the lexicon yielding the highest performance contains an average of no more than 1.6 variants per name. When we compare the tree search approach with the other selection methods presented in this dissertation, we see that it is capable of producing the best performance altogether. There is a significant improvement over our baseline probability-based selection approach for all lexicon sizes, and the discriminative single-pass MCE selection approach is significantly outperformed for lexicons containing between approximately 900 and 1200 variants. For other lexicon sizes, and when comparing the tree search approach with the multi-pass MCE approach, the differences are not quite big enough to confirm their statistical significance, but the numbers do show in favor of the tree search algorithm.

A noteworthy element in this figure is the distinctive shape of the tree search curve. While the other curves show a steady decrease in NER before either leveling out or tipping only slightly upwards, the dark blue curve of the tree search approach clearly exhibits a more rapid decrease in NER. This seems to indicate that the tree search algorithm is indeed capable of selecting the strongest variants early on in the lexicon construction process, as it is designed to do. It seems likely that it does this better than the other algorithms presented here because it has a non-uniform distribution of variants across the name space: while the other algorithms select one variant per name in sequence, the tree search approach adds variants in order of expected performance gain, irrespective of the amount of variants already representing a particular name. This allows for better variant selection and more flexible stopping criteria, which results in a steeper initial decline in NER and more compact lexicons.

Experimental analysis of the evaluation function

In Section 6.3.2, we observed that our multi-pass MCE approach might be improved by adopting a best-first approach, where new MCE scores are calculated for every lexicon change, rather than for every round of lexicon changes. That is of course exactly what we aim to achieve with the tree search method proposed in this chapter. It may be argued, however, that it is far from obvious whether the introduction of the error potential factor $\tilde{h}_{kl}(n_{kl})$ provides any additional value. It might have been sufficient to simply apply the same decision rule used in the previous chapter; that is to say to recalculate MCE scores after every addition to the recognition lexicon and to select the variant that maximizes the decrease in overall expected loss in recognition performance. On the other hand, the idea of expressly prioritizing names that show the greatest room for improvement does seem to have intuitive merit. In order to investigate the effective contribution of the error potential factor, we have therefore conducted two additional experiments, the results of which are shown in Table 7.2.

The left-hand column in this table reproduces the results of our original tree search experiment, as described in the previous section and previously presented in Table 7.1. The middle column shows the results of an experiment where the correction potential factor $\tilde{g}_{kl}(n_{kl})$ is used as the evaluation function, without its scaling factor α_l. The upshot of this is an approach that is effectively identical to the multi-pass method used in Chapter 6, except for the crucial difference that additions to the lexicon are done individually rather rather than collectively. The right-hand column, then, shows the results of the obverse experiment, in which the error potential factor $\tilde{h}_{kl}(n_{kl})$ is used as the evaluation function. In this experiment, names with a high

$\tilde{f}_{kl}(n_{kl})$		$\tilde{g}_{kl}(n_{kl})$		$\tilde{h}_{kl}(n_{kl})$	
Size	NER	Size	NER	Size	NER
617	13.53%	617	13.53%	617	13.53%
717	12.58%	717	12.71%	717	13.08%
817	12.04%	817	12.38%	817	12.70%
917	11.88%	917	12.39%	917	12.54%
1017	11.80%	1017	12.15%	1017	12.18%
1117	12.07%	1117	12.04%	1117	12.14%
1217	11.96%	1217	11.96%	1217	12.25%
1317	12.00%	1317	12.13%	1317	12.00%
1417	12.05%	1417	12.02%	1417	12.04%
1517	-	1517	12.02%	1517	12.00%
1617	-	1617	12.02%	1617	-

Table 7.2: *Performance of a tree-search variant selection approach when using $\tilde{f}_{kl}$, $\tilde{g}_{kl}$ and $\tilde{h}_{kl}$ as evaluation functions.*

number of misrecognized training utterances are prioritized entirely.

The main observation to be made from Table 7.2 is that the combination of the error potential factor with the correction potential factor into the evaluation function $\tilde{f}_{kl}(n_{kl})$ does produce the overall best result (11.80% NER) and the best results for the smallest lexicon sizes. For lexicon sizes of less than 1000 variants, $\tilde{g}_{kl}$ performs appreciably better than $\tilde{h}_{kl}$, but is in its turn outperformed by $\tilde{f}_{kl}$. This would seem to bear out our intuition that there is some performance gain to be obtained from the *combination* of both factors. Interestingly, however, after this initial benefit, recognition performance in the three experiments converges completely.

Another observation that we want to make from these experiments, is how they compare with our multi-pass MCE approach. As noted previously, doing such a comparison is somewhat problematic, since the multi-pass approach has a different iteration loop, and it can therefore not be evaluated for any given lexicon size: we are bound to the discrete evaluation points defined as a function of M, the maximum number of allowed variants per name. Referring back to Table 6.1, the best possible approximation is to compare the multi-pass MCE Name Error Rate for $M = 2$ (lexicon size 1184) with the closest tree search lexicon size of 1217. This shows a performance gain of 0.46% absolute in favor of the tree search approach. Interestingly, at this particular lexicon size, the results of the experiments using $\tilde{f}_{kl}$ and $\tilde{g}_{kl}$ have already begun to converge. This means that, for this lexicon size, no noticeable gain is derived from $\tilde{h}_{kl}$ or from the scaling factor α_l: the initial benefits that our selection approach reaps from applying these as factors in the evaluation function have at this point already been leveled out. The performance gain over the multi-pass MCE approach, then, is to be ascribed completely to the improved order in which variants are added to the lexicon. This results in a more optimal distribution of variants across the name space, properly prioritizing the names that benefit most from the inclusion of additional pronunciation variants.

Finally, the fact that the three columns differ in length may deserve some clarification. The differences are to be ascribed to the respective stopping criteria. The experiment using only $\tilde{g}_{kl}$ as its evaluation function selects the greatest number of variants, due to the fact that $\tilde{h}_{kl}(n_{kl}) = 0$ does not count as a stopping criterion: this experiment is intended as the closest possible approximation to the multi-pass approach, where $\tilde{h}_{kl}$ does not affect the selection algorithm in any way. Perhaps somewhat surprisingly, on the other hand, $\tilde{g}_{kl}(n_{kl}) = 0$ does play a role as a stopping criterion in the experiment using $\tilde{h}_{kl}$ as its evaluation function. The reason for this is that, in the case where none of the available variants affects recognition performance, the stopping criterion $\tilde{h}_{kl}(n_{kl}) = 0$ would never be met, and all available variants would therefore be selected, regardless of their poor suitability. We therefore elected to allow $\tilde{g}_{kl}$ some impact on the $\tilde{h}_{kl}$ experiment. The upshot of this is that this experiment applies the exact same range of stopping criteria as the original experiment using $\tilde{f}_{kl}$, described in Section 7.1.1. The reason that the right-hand column is nevertheless slightly longer than the left-hand column, then, is that due to the more efficient selection of the most suitable variants during the earlier iterations in the $\tilde{f}_{kl}$ experiment, the stopping criteria $\tilde{g}_{kl}(n_{kl}) = 0$ and $\tilde{h}_{kl}(n_{kl}) = 0$ are reached faster.

Including AV variants in the candidate set

Although the tree search selection approach performed better than the previously proposed variant selection approaches, it was still unable to outperform the AV lexicon comprising only auditorily verified transcriptions of the name utterances in the training set. In order to investigate the behavior of the selection algorithm given the availability of these higher quality transcriptions, we repeated the experiment with an extended candidate pool, now containing the auditorily verified transcriptions as well as the candidates generated by the g2p and p2p converters. Figure 7.6 compares the results of this experiment with the performance of the single-pass and the multi-pass selection approaches in the corresponding experiments.

This figure shows that the tree search variant selection approach offers a

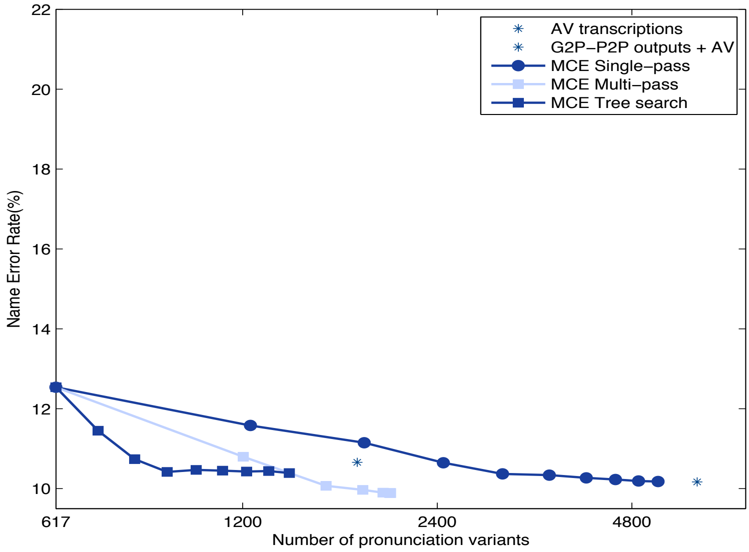

Figure 7.6: *The results for the single-pass MCE, multi-pass MCE and tree-search approaches using a candidate pool containing p2p, g2p and AV transcriptions.*

rapid decrease in name error rate (2.12% absolute) after adding on average only 0.5 variants per name (from 617 variants to 917 variants). Interestingly, however, this decrease in NER completely flattens out after this point and ends in a performance equal to that of the MCE multi-pass selection approach. Moreover, after the tree search algorithm has converged, the MCE multi-pass approach continues to select additional promising pronunciation variants, yielding a final performance surpassing that of the tree search selection approach.

The more rapid initial drop in NER observed in the tree search results is most likely an effect of the tree search algorithm's ability to select the variants yielding the highest performance gain at the earliest stages. In fact, the selection algorithm is able to construct a lexicon yielding the same performance as the AV lexicon using only half the number of transcription variants. However, once the variants covering most of the pronunciation variation observed in the training material have been added to the recognition lexicon, the tree search algorithm is no longer able to differentiate the promising variants from the redundant. One hypothesis which might explain this is that since the selected variants now are of a much higher quality, the variants which are selected first cover the majority of the variation observed in the training set; in other words: they have a high correction potential factor. The variants selected after this point will therefore have a very small, and mutually similar, correction potential factor. This can explain why these variants do not correct any additional errors and also why the tree search algorithm does not converge. A possible reason why the multi-pass approach is able to correct more errors in this experiment is that this approach employs a more imprecise stopping criterion, adding several variants which cover some of the variation not seen in the training set.

To investigate whether we could remedy this effect by relaxing some of the stopping criteria in the tree search approach, we performed three additional experiments the results of which are shown in Table 7.3.[3]

$\tilde{f}_{kl}(n_{kl})$ $L = 4$		$\tilde{f}_{kl}(n_{kl})$ $L = 6$		$\tilde{g}_{kl}(n_{kl})$ $L = 4$		$\tilde{g}_{kl}(n_{kl})$ $L = 6$	
Size	NER	Size	NER	Size	NER	Size	NER
617	12.54%	617	12.54%	617	12.54%	617	12.54%
717	11.45%	717	11.67%	717	11.31%	717	11.31%
817	10.74%	817	10.76%	817	11.01%	817	11.01%
917	10.42%	917	10.40%	917	10.63%	917	10.63%
1017	10.47%	1017	10.28%	1017	10.56%	1017	10.56%
1117	10.45%	1117	10.20%	1117	10.46%	1117	10.46%
1217	10.43%	1217	10.39%	1217	10.31%	1217	10.31%
1317	10.44%	1317	10.43%	1317	10.26%	1317	10.26%
1417	10.37%	1417	10.43%	1417	10.23%	1417	10.23%
1517	-	1517	-	1517	10.20%	1517	10.20%
1617	-	1617	-	1617	10.20%	1617	10.20%

Table 7.3: *Performance of a tree-search variant selection approach when using different evaluation functions and L-values in the case when the candidate set contain auditorily verified transcriptions.*

Again, the left hand column of this table reproduces the results of the original AV experiment. The middle-left column shows the results of the exact same experiment except that the value of L has been increased from 4 to 6.[4] In the two columns to the right, we repeated the $\tilde{g}_{kl}(n_{kl})$ experi-

[3] A candidate pool containing auditorily verified transcriptions was used in all three experiments.

[4] This particular value of L was chosen because the maximum number of variants per

ment described in the previous section, using only the correction potential factor $\tilde{g}_{kl}(n_{kl})$ as the evaluation function. The first experiment (the results of which are shown in the middle-right column) was performed using an L-value of 4, while in the second experiment (results shown in the rightmost column) this value was again set to 6.

The results of these experiments showed that increasing the maximum number of variants per name from 4 to 6 had no noteworthy effect on the recognition performance. In other words, the restriction on L is not the main reason why the tree search approach is unable to reach the same performance as the multi-pass approach. When repeating the $\tilde{g}_{kl}(n_{kl})$ experiment, we observed that the results were virtually identical to those of the original tree search approach when tested using a candidate pool comprising auditorily verified transcriptions. These results seem to indicate that the error potential factor in Equation (7.7) contributes less to the overall recognition performance when the quality of the variants in the candidate set is high.

Interestingly, none of these experiments resulted in lexicons containing more than 1617 variants. Even the two $\tilde{g}_{kl}(n_{kl})$ experiments did not generate lexicons containing as many variants as the multi-pass approach, even though these experiments employed no additional stopping criteria other than those incorporated in the multi-pass approach. One hypothesis that might explain this effect is that the order in which variants are added to the lexicon alters the values of the expected loss (the MCE loss value) to such a degree that the tree search algorithm converges earlier than the multi-pass approach. This would mean that there are still beneficial variants left in the candidate pool which are not exploited by the tree search approach. However, since it is hard to analyze this effect directly, we can only speculate as to its impact on the recognition result.

7.2.2 Testing in an open environment

The tree search selection approach gave a considerable performance increase compared to the probability-based baseline when tested in a controlled environment. To compare the performance of the different selection approaches in a large vocabulary system, we repeated the recognition experiment using a vocabulary of 16,045 names, namely the 617 names in the data set and 15,428 filler names.

As in the multi-pass approach, we wanted to add pronunciation variants for the filler names in a way that reflected the behavior of the selection al-

name in the multi-pass approach was 6.

gorithm. To achieve this, a simulation algorithm was implemented to select variants for the filler names. This simulation algorithm selects filler variants with properties similar to that of the original variants selected by the tree search selection algorithm. The variant properties deemed important in this regard are the name the variant represents and the number of variants already representing this name in the lexicon.

In practice, this means that when the tree search selection algorithm has selected a pronunciation variant to represent one of the original 617 names, the simulation algorithm selects $15,428 \cdot \frac{1}{617} = 25$ variants from the pool of filler pronunciation variants. To decide which filler variants to select, the simulation algorithm uses the properties of the original variant selected by the tree search algorithm. For example, if the variant selected by the tree search selection algorithm represents a name which already has two variants in the recognition lexicon, the simulation selection algorithm retrieves a subset of all the filler names which also are represented with two variants in the recognition lexicon. The simulation algorithm then sorts this subset according to the p2p probability of the third most probable variant and selects the 25 names with variants on top of this list. These variants are then added to the recognition lexicon and the process is repeated.

To compare the performance of the tree search selection approach with that of the previously proposed selection methods, the results of this experiment are illustrated in Figure 7.7 in juxtaposition with the results of the single-pass and multi-pass MCE approaches. When interpreting these results, however, we should keep in mind that the large vocabulary experiments performed in this dissertation have a somewhat artificial set-up, and that it is therefore difficult to draw definite conclusions from these data.

This figure shows that the performance of the tree search selection approach is very similar to that of the multi-pass MCE approach. Contrasting the two approaches, we observe that the initial increase in performance achieved by the tree search approach is not as marked as in the small vocabulary experiment. A possible explanation for this is that the performance gain observed after adding one variant to the recognition lexicon is smaller in a large vocabulary setting, due to the increased lexical confusion in the large vocabulary lexicons. This means that the values of the correction potential factor $\tilde{g}_{kl}$ and the values of the multi-pass decision rule are relatively smaller in large vocabulary systems than in small vocabulary systems. In other words, there is less clear evidence on which to base the selection, resulting in a measure of randomness in the selection procedure. Of course, this affects the performance of both the tree search and multi-pass approach, but the effect is somewhat less prominent in the multi-pass

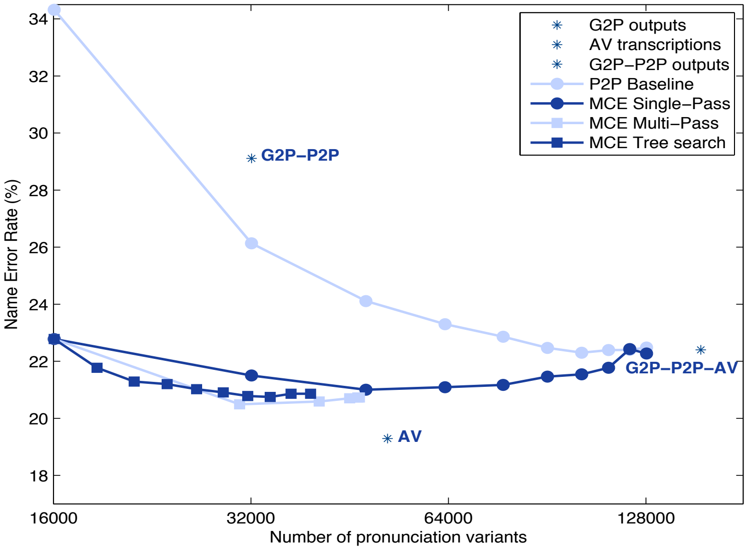

Figure 7.7: *The results for the baseline P2P, single-pass MCE, multi-pass MCE and tree-search MCE approaches as a function of the lexicon size in the case of a 16k vocabulary.*

approach, as variants are added collectively in each iteration. The order in which variants are added to the lexicon is therefore not considerably affected by this. In the tree search approach, however, the variants are added individually on the basis of their $\tilde{g}_{kl}$ and $\tilde{h}_{kl}$ values. Changes in these values will therefore affect the order in which variants are added to the lexicon to a much higher degree. A reduction in $\tilde{g}_{kl}$ values and $\tilde{h}_{kl}$ values will give the tree search more of a random character, as a large number of nodes in the stack have nearly identical $\tilde{f}_{kl}$ values.

Including AV variants in the candidate set

As in the previous experiments, we added the auditorily verified transcriptions of the training set to the pool of pronunciation candidates and repeated the large vocabulary experiment. The results of this experiment are shown in Figure 7.8. In this figure, the AV lexicon contains auditorily verified transcriptions for all the 617 names in the data set and the three most probable variants for the filler names, while the G2P-P2P-AV lexicon contains all the variants in the extended variant pool.

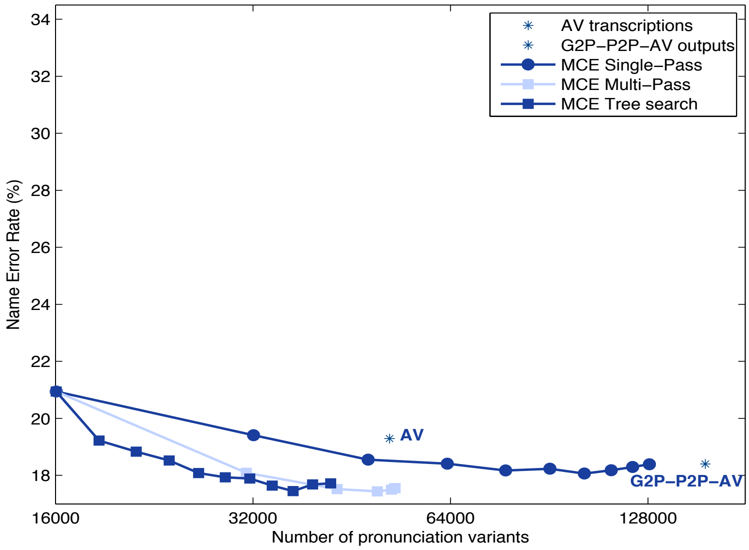

Figure 7.8: *The results for the single-pass MCE, multi-pass MCE and tree-search approaches tested in an open environment using a candidate pool containing p2p, g2p and AV transcriptions.*

This figure shows that both the multi-pass and the tree search approach significantly outperformed the AV lexicon with a 2% absolute reduction in NER. The G2P-P2P-AV lexicon was also outperformed by both approaches with a 1.14% absolute reduction in NER (using only one fourth of the pronunciation variants). Furthermore, the strict stopping criterion implemented in the tree search approach proved to be quite effective in this experiment, as the best performing tree search lexicon achieved the same NER as the best performing multi-pass lexicon, using only 2.5 variants per name compared to 3.1 variants per name for the multi-pass approach.

Figure 7.8 further illustrates that the tree search approach seems to benefit more from having high quality variants in the variant pool compared to the multi-pass approach, particularly when the lexicon contains less than 32,000 variants. After this point, the two approaches achieve equivalent recognition performances. This can most likely be attributed to the fact that adding high quality variants to the recognition lexicon generally results in a larger performance gain, hence providing the tree search algorithm with stronger evidence of which is the better performing variant. This reduces the random factor discussed in the previous subsection and enables the tree search approach to make a better informed selection at the individual variant level.

7.3 Discussion

7.3.1 Observations

The variant selection approach proposed in this chapter aims to add pronunciation variants to the lexicon entries which benefit the most from having an additional variant, regardless of the number of variants already representing this entry in the lexicon. To achieve this, the recognition lexicon was represented as a tree structure with a set of branches (names), each containing a set of nodes (variants). This search tree was then incrementally populated with the most promising node n_{kl} in a best-first manner. By compiling the recognition lexicon in this way, both accuracy (in terms of recognition performance) and compactness (in terms of lexicon size) were optimized. The upshot of this variant selection approach is that the recognition lexicon always contains the optimal set of pronunciation variants for any given lexicon size. This means that if we require our lexicon to be of a specific size, we can simply use the tree search approach to generate a lexicon of the required size and this lexicon will inherently comprise the optimal set of variants for that lexicon size. This contrasts sharply with the previously proposed variant selection approaches where the generated lexicons are always constrained by the maximum number of allowed variants per name (M).

The tree search approach was evaluated in a controlled and in an open environment. In both environments, a significant performance increase was observed in favor of the proposed approach when compared to the baseline P2P approach, whereas a small performance increase was observed for small lexicon sizes when the results were compared to those of the multi-pass approach. Additionally, in all experiments conducted in this chapter, a rapid initial decrease in name error rate was observed for the proposed approach after adding only a small number of variants. This is most likely an effect of the ability of the tree search algorithm to identify the names that benefit the most from the inclusion of an additional variant. By selecting variants in this manner, the tree search approach is able to generate lexicons with high coverage for names that are likely to be pronounced in a variety of ways, even for small lexicon sizes. This effect was also observed after adding auditorily verified variants to the candidate pool, but in this case the performance stabilized after the initial increase.

The experiments conducted in an open environment showed that the performance of the tree search approach is very similar to that of the multi-pass approach in large vocabulary systems. We hypothesized that this can be ascribed to the fact that the impact on overall performance of a single added variant is generally smaller when the vocabulary is large compared to in a small vocabulary setting. In practice, this means that the variants selected by the tree search approach in an open environment will have very small, and mutually similar, correction potential factors. This makes it more difficult for the selection algorithm to decide which lexicon entry benefits most from having an additional variant, and the variant selection is subject to random effects. This hypothesis is supported by the open environment experiment using auditorily verified variants in the candidate pool. The results of this experiment show that the performance gap between the tree search approach and the multi-pass approach is relatively larger compared to when using only p2p-g2p variants in the candidate pool. This performance increase can most likely be attributed to the fact that the AV variants tend to have a relatively high correction potential, and therefore reduce the degree of randomness affecting the tree search approach.

There are three main differences between the multi-pass approach and the tree search selection approach. The first and most important difference is the flexibility of the order in which the variants are added to the lexicon. This property enables the tree search approach to generate a lexicon with a non-uniform distribution of variants across the name space. The second difference lies in the introduction of an extra stopping criterion which prevents the addition of variants to names which have already reached their error potential (i.e. names for which the lexicon already covers all the pronunciation variation observed in the training utterances). This enables the tree search approach to avoid adding potentially redundant variants to the lexicon, resulting in a more compact lexicon. The last difference is the inclusion of the error correction factor $\tilde{h}_{kl}$ in the evaluation function $\tilde{f}_{kl}$. When analyzing the evaluation function in Section 7.2.1 we observed that the combination of the error potential factor with the correction potential factor into the evaluation function $\tilde{f}_{kl}$ does produce the best results, at least for the smallest lexicon sizes. Interestingly, this is not necessarily true for candidate sets of different quality, as we observed when we added auditorily verified variants to the candidate set. In this case, the error potential factor neither increased nor decreased the recognition performance compared to using an evaluation function consisting only of the correction potential factor. A possible explanation is that the correction potential factor $\tilde{g}_{kl}$ tends to be considerably higher for high quality variants. The contribution of the correction potential factor to the evaluation function tends to be very heavy in these cases, rendering the contribution of the error correction factor negligible.

To illustrate the difference in behavior between the variant selection approaches proposed in this dissertation, we made a visual representation of the development of the lexicons constructed using the single-pass MCE approach (Figure 7.9), the multi-pass MCE approach (Figure 7.10) and the tree search approach (Figure 7.11). Each column in these figures represents the total number of variants selected for a specific lexicon size. The division of the columns into red and blue sections is to be interpreted as in Section 5.3 and in Section 6.3: the blue sections represent variants that triggered a correct recognition, the red sections represent variants that were either not used or used in misrecognitions. As in the previous two chapters, the numbers correspond to a recognition pass performed on all test utterances, using the end lexicon, obtained after 10 iterations in the case of the single-pass MCE approach and after the convergence of the respective selection algorithms for the two other approaches.

Strikingly, the number of blue variants in every set of 100 variants added by the tree search selection algorithm is nearly constant, whereas the number of blue variants added in the single-pass and multi-pass MCE approaches decreases steadily with each iteration. We also observe that the number of blue variants in the first column is higher for the tree search approach than for either of the two other approaches, even though the three corresponding lexicons are completely identical. This phenomenon, which we encountered previously in Section 6.3.1 when comparing the behavior of the single-pass and the multi-pass MCE approach is most likely an effect of the stricter stopping criteria which prevent the inclusion of superfluous variants, forcing the recognizer to resort to the more generic variants added in the first iteration. It is reasonable to assume that this effect also applies to the variants selected subsequently, although it is less straightforward to compare these. The upshot is a far more compact lexicon, where the ratio of successful variants is markedly higher: 51% of the variants selected by the tree search approach are successful variants, versus 41% in the multi-pass approach and only 20% in the single-pass approach.

Figure 7.12 and Figure 7.13 illustrate the selection behavior of the multi-pass and the tree search approach in a different way. The figures show for different lexicon sizes how the lexicon entries are distributed with regard to the number of variants by which they are represented. The blue curves in

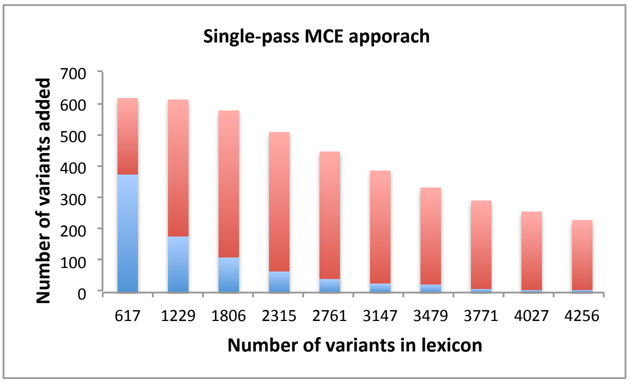

Figure 7.9: *Number of variants added per iteration using the single-pass method.*

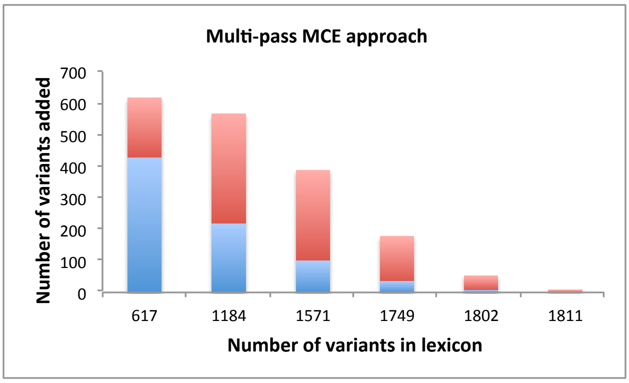

Figure 7.10: *Number of variants added per iteration using the multi-pass method.*

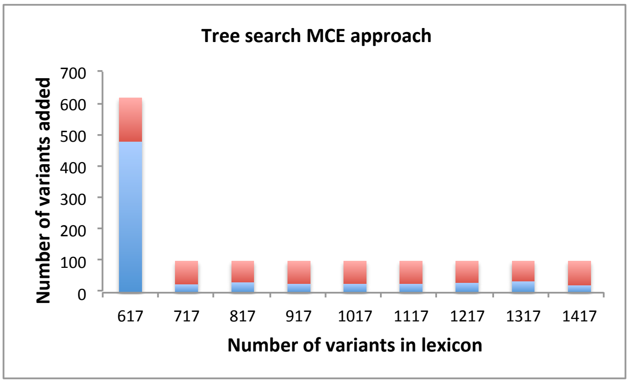

Figure 7.11: *Number of variants added per iteration using the tree search method.*

these figures illustrate the NER attained using the corresponding lexicons.[5]

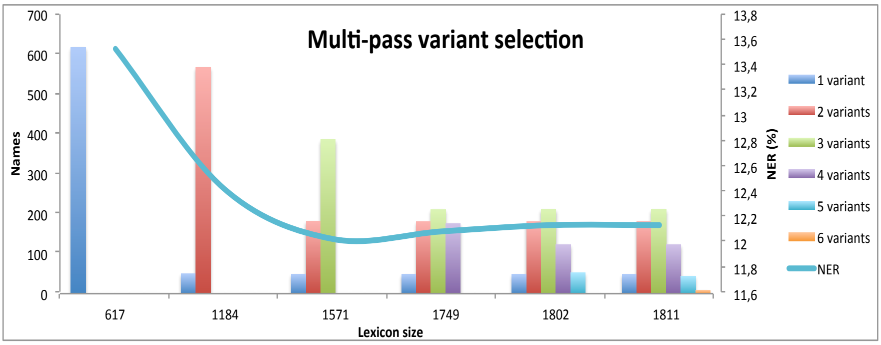

Figure 7.12: *Number of variants per lexicon entry for lexicons generated using the multi-pass selection approach.*

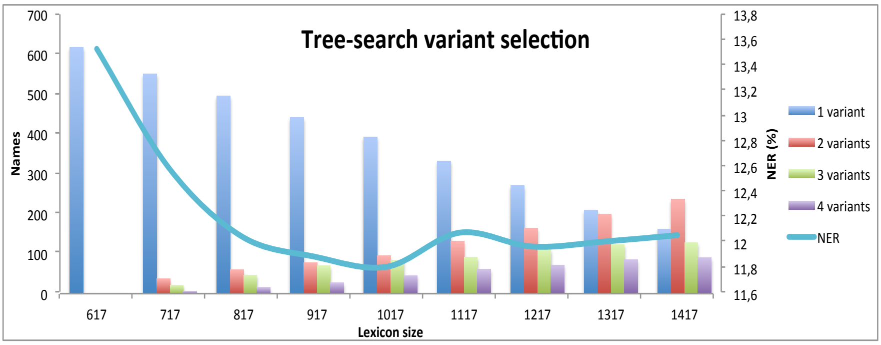

Figure 7.13: *Number of variants per lexicon entry for lexicons generated using the tree search selection approach.*

One of the most striking observations to be made from Figure 7.12 is that over 90% of the names in the lexicon receive an additional pronunciation variant during the second iteration of the multi-pass selection algorithm. This stands in sharp contrast to the tree search approach, where more than a quarter of the names are represented by only one single pronunciation variant in the final recognition lexicon. The contrast is even more emphatic when we compare the multi-pass lexicon for $M = 2$ with the tree search lexicon that most closely corresponds to it in size: in the tree search lexicon

[5] It should be noted that the differences in appearance between Figures 7.12 and 7.13 should not be taken at face value, as the scales on the corresponding X-axes are not identical.

containing 1217 variants, 44% of the names are represented by a single pronunciation variant. Given that the tree search lexicons consistently yield slightly better recognition performance, it may be clear that this provides further proof that they contain fewer redundant variants than the multi-pass lexicons.

In the same vein, Figure 7.13 illustrates that there is nearly a two-thirds majority (64%) of names that are represented by only one variant in the lexicon that yields the best recognition performance, viz. the tree search lexicon containing 1017 variants. The 7% most problematic names, on the other hand, are represented by the maximum of 4 pronunciation variants, while the remaining names are more or less equally distributed between the intermediate levels. This shows that our working assumption for designing the tree search approach was correct: there is clearly recognition gain to be obtained from adding variants in a non-linear, best-first manner.

7.3.2 Limitations of the tree search approach

The tree search selection algorithm aims to generate a lexicon that covers the pronunciation variation observed in the training material using as few pronunciation variants as possible. One acute limitation of this approach is that pronunciation variation not observed in the training material will not be well represented using this selection algorithm. This is further exacerbated by the efficiency of the tree search approach: we have seen that the selection algorithm is particularly successful in the earliest iterations, which in many cases results in the algorithm's strict stopping criteria being met quickly. The upshot of this is extremely compact lexicons yielding good recognition performance, at the cost of increased data dependency. It may be clear, however, that this effect is strongly dependent on the size of the training set: a larger training set is likely to contain more variation and can therefore be expected to result in a lexicon with greater coverage. In this way, this shortcoming can be counteracted to some extent.

Two further weaknesses of the tree search selection approach follow from the fact that the error potential factor and correction potential factor are ultimately based on the same source of information, namely the expected loss of recognition accuracy. Firstly, while our experiments in Section 7.2.1 showed that both factors do contribute individually to the performance increase observed in favor of the tree search approach, it seems likely that utilizing some other measure for the error potential factor might be beneficial. We may speculate that this might be especially advantageous in environments where the error potential factor proved to be less effective, as was the case when we added auditorily verified variants to the candidate pool. This may well be a promising direction for future work aiming to improve the performance of the tree search selection approach. Secondly, once the differences in expected loss of recognition accuracy become negligibly small, the evaluation function used by the tree search approach loses much of its discriminative potential, and the approach becomes susceptible to random effects.

Finally, although the computational load is noticeably lower for the tree search selection approach than for the multi-pass selection approach, it is still relatively high when optimizing large vocabularies. However, given that the optimization process need be performed only once, and that it can be performed off-line, this issue does not seem critical.

7.4 Conclusion

In this chapter, the pronunciation variant selection task was recast as a best-first tree search problem, where the final recognition lexicon corresponds with the optimal path through a tree structure. To guide the search algorithm, an evaluation function consisting of a correction potential factor and an error potential factor was defined.

We have shown that optimizing the pronunciation variant selection by means of a best-first discriminative tree search algorithm is beneficial in terms of reduced lexicon size and increased recognition performance. The performance gain observed for small lexicon sizes demonstrated that the order in which variants are added to the recognition lexicon is an important contributor to recognition success. The introduction of the error potential factor as a factor in the evaluation function also proved to have some impact on the recognition accuracy when using a candidate set comprising only p2p-g2p variants. Finally, the strict stopping criteria of the proposed selection approach were shown to prevent the addition of redundant variants, which reduced the lexical confusion between the lexicon entries.

The main limitation of the tree search selection algorithm is its data dependency: the tree search approach is ill-suited to cover pronunciation variation that is not observed in the training data. This is especially so given the restrictive nature of the selection algorithm. While the efficiency of the selection approach during the earliest iterations and the use of strict stopping criteria result in highly compact recognition lexicons, it appears that more leniency might be beneficial in order to address unseen pronunciation variation.

Chapter 8

Conclusions

In this dissertation, we have investigated different approaches to pronunciation variation modeling of non-native proper names. Traditionally, lexical pronunciation modeling has been dominated by heuristics and various subjective optimization measures. The main goal of this work has therefore been to model the recognition lexicon in a data-driven manner, using an objective variant selection criterion directly connected with the actual recognition performance. The following section summarizes the main results of this research and Section 8.2 suggests some directions for future work.

8.1 Contributions of this dissertation

Three different variant selection approaches have been proposed in this dissertation and several experiments have been conducted to assess the behavior and performance of these approaches. In this section we will describe the main results drawn from these experiments. It should be noted that these experiments have been limited to the task of non-native name recognition and the NameDat corpus,[1] and that the variant selection approaches proposed in this dissertation have only been evaluated using a candidate pool consisting of transcriptions generated by a P2P converter and auditorily verified transcriptions. Therefore, the conclusions drawn here do not necessarily generalize to other recognition tasks or name databases, or to variant transcriptions of different quality.

[1] With the exception of the breadth-first variant selection approach described in Chapter 6, which was also tested using the Autonomata Spoken Name corpus [106]. These experiments were presented in [3].

An initial study of pronunciation variation of non-native proper names

In an initial study on the nature of non-native names in automatic speech recognition, we aimed to get a better understanding of the properties we might expect in a lexicon that allows for good recognition performance. The result of this study showed that there are two main properties which are important in this regard: the *quality of the variants* in the lexicon and the *coverage* of the lexicon (i.e. its ability to cover different pronunciation phenomena)[2]. The first property is quite simple, an accurate phonological transcription of a speech utterance will always perform better than an inaccurate transcription. The second property is somewhat more ambiguous. Ideally, the lexicon should contain an accurate transcription of every possible pronunciation. Unfortunately, this tends to introduce unwanted confusion between lexicon entries which may decrease the recognition performance. In our initial study we identified two key factors which may be relevant with respect to lexicon coverage. Firstly, we found that some proper names are likely to have a variety of different pronunciations, whereas other names can be expected to have a more uniform set of pronunciations. The lexicon should therefore contain a variable number of transcriptions for each entry, depending on the anticipated amount of pronunciation variation for that entry. Secondly, we found that large vocabulary lexicons seemed to benefit more from having additional transcription variants than small vocabulary lexicons, even though the lexical confusion is higher in these lexicons. This indicates that having variants of a higher quality in the lexicon is even more important when lexical confusion is higher.

Based on our findings from this initial study we formulated as a working hypothesis that a lexicon expected to yield good recognition performance should, for each entry, contain:

1. transcription variants of high quality;
2. transcription variants covering different pronunciation phenomena;
3. transcription variants correcting more errors than they introduce;
4. a number of transcription variants reflecting the anticipated pronunciation variation of the entry.

[2] We assume that the vocabulary size is given: in a different sense, the coverage of a lexicon can also be said to be dependent on the number of entries for which it contains variants, but our research is specifically concerned with pronunciation variation, and we therefore focus on this meaning of the concept of coverage.

A comparative study of different variant selection criteria

In a detailed comparative analysis of four different variant selection criteria, we found that variant selection criteria based on evidence extracted from the recognition engine significantly outperformed our baseline probability-based selection criterion. The analysis further showed that the variant selection criterion yielding the most promising results was based on the Minimum Classification Error (MCE) framework. We also found that this framework is particularly well-suited as a basis for new variant selection algorithms due to its ability to evaluate the difference in recognition performance when making use of different lexicons, for instance before and after a lexicon change. Optimizing the recognition lexicon by adding the variants according to their MCE score was found to perform considerably better than the probability-based baseline approach and to be computationally inexpensive. This approach, however, was also prone to selecting variants covering the same pronunciation phenomena and to include many redundant variants in the lexicon.

A breadth-first variant selection approach

In our "breadth-first" variant selection algorithm, variants were only added to the lexicon on the condition that they correct recognition errors left unhandled by the initial lexicon. Optimizing the recognition lexicon using this approach showed to reduce the error rate compared to the single-pass approach and result in a compact lexicon covering different pronunciation phenomena. The breadth-first approach selected different numbers of transcription variants for each lexicon entry, but additional experiments showed that the lexicon still contained some redundant variants. Selecting variants in this manner was particularly effective when the quality of the transcription variants were high.

A best-first variant selection approach

At the end of each iteration loop, our "breadth-first" approach evaluates for each name in the lexicon whether it is beneficial to add its top-ranked pronunciation variant. However, since the level of observed pronunciation variation strongly differs from name to name, we might do better with a "best-first" variant selection algorithm that prioritizes inclusion of variants for names where most variation is expected. Optimizing the lexicon by selecting variants in a "best-first" manner showed to be beneficial in terms of reduced computational load, lexicon size and error rate. Selecting variants in this manner generated lexicons containing variants correcting different types of recognition errors and very few of the variants added to the lexicon were redundant. As this approach greatly reduces the lexicon size, the approach may suffer from generalization problems as a result of outlier pronunciations not seen in the training data.

8.2 Future work

In this section we will give some general directions to future research.

Expanding the Maximum Entropy model

In our comparative study of different variant selection criteria, the Maximum Entropy (ME) model satisfied one single constraint, namely if an utterance was correctly recognized or not. As the Maximum Entropy framework is designed to use several constraints simultaneously, it would be interesting to put additional constraints on the ME-model to see if we can utilize various kinds of information rather than just the recognition result. To determine the kind of information that may be helpful to the variant selection task, a more in-depth analysis must be performed. One direction that may be fruitful is to simulate the length of a transcription by means of acoustic models and then compare this length with that of the utterance. Another direction that can be used in iterative selection schemes, such as the breadth-first selection approach or the best-first selection approach, is to constrain the ME-model to give lower weight to variants which are very similar to the variants already in the lexicon. To determine the similarity of two variants string metrics such as the Levenshtein distance can be employed.

Repeating large vocabulary experiments using large data set

The large vocabulary experiments performed in this dissertation had a somewhat artificial setup due to the limited size of our data set. In these experiments, the effect of the proposed variant selection approaches was mimicked by a simulation algorithm and the resulting lexicon was only evaluated on a subset of the names in the vocabulary. To investigate the real effect of the proposed selection algorithms on a large vocabulary task, it would be very interesting to repeat these experiments using a large data set extracted from a real-life application such as a call center application or a car navigation application.

Handling data dependency

The pronunciation variation modeling approaches proposed in this dissertation all rely on having relevant training data available in the optimization process. This data dependency can in many cases pose some challenges to the designer of the variant selection approach and to the selection approach itself. Firstly, having to collect and process a large amount of training data can be impractical as well as quite costly. Secondly, it makes the selection approach highly dependent on the pronunciation variation captured in the training data, which gives rise to generalization problems when encountering pronunciations that differ significantly from the pronunciations seen in the training data. Finally, it makes the proposed methods unable to select good pronunciation variants for unseen names. In this section we will give some suggestions for directions of future research in this area which may help to circumvent some of these issues.

Formulating the proposed variant selection methods in terms of more generic mechanisms, e.g. the ones modeled by the p2p converter, would enable the generation of more accurate transcriptions for both seen and unseen names. Investigating this further would therefore be of great interest. One technique that is often used in conjunction with lexical pronunciation variation modeling, is to use the variants chosen by the selection algorithm to iteratively re-train the mechanism that generates the transcription variants and produce new and more accurate variants. It would be interesting, therefore, to use a similar scheme to re-train the phoneme-to-phoneme converter so as to give more weight to pronunciation phenomena proven to have error correcting capabilities.

An interesting approach that might prove useful to reduce the generalization problem is that of McAllaster *et al.* described in [46]. In this approach, acoustic speech data is simulated using a set of acoustic models and a recognition lexicon. Using such an approach to simulate acoustic data for the transcription variants proposed by the p2p converter can enable the proposed selection approaches to make better informed decisions as to which, and how many, variants to include in the recognition lexicon. A similar approach that might be investigated further is to simulate the recognition errors without having to use actual acoustic data. An interesting approach in this respect is that of Jyothi and Fosler-Lussier [72]. This approach used a predictive WFST framework, composed of a confusion matrix that used acoustic and pronunciation information from the recognizer to model possible phone confusions. These confusions were then used to simulate probable word errors. A study investigating whether these simulated word errors can be used independently, or in conjunction with some other objective performance measure, to select good performing pronunciation variants would be of great interest.

Appendix A

Statistical Significance Testing

When comparing the performance of two different speech recognition algorithms, it is important to be able to assess whether an observed performance difference is related to one algorithm actually performing better then the other, or if the difference is merely due to chance effects. In this appendix a set of confidence intervals is calculated for a predefined selection of name error rates using the total number of test utterances in our three test sets. These confidence intervals enable us to test whether or not an observed difference in NER is statistically significant.

A.1 Confidence intervals

In the experiments conducted in this dissertation n isolated name utterances are tested in each experiment. For each of these experiments we observe whether a name utterance was correctly recognized or not and count the total number of errors n_e made by the recognition algorithm. Using the observed number of errors the name error rate is normally estimated as

$$\hat{p} = \frac{n_e}{n}.$$

Since this is only an estimate of the true error rate p it necessary to find an interval in which the true error rate is most likely to be found. This interval is called the confidence interval and is often defined as

$$P(c_1 < n_e < c_2) = 1 - \alpha$$

where c_1 and c_2 are chosen in such way that the probability, $P(c_1 < n_e < c_2)$, equals a predefined confidence level $1 - \alpha$. This confidence level is very often set to 95%, which is the same as will be used in this dissertation. By calculating such a confidence interval we know with a 95% certainty that the true value of p can be found within the interval (c_1, c_2). If an error rate produced by another recognition algorithm is outside this interval the algorithm is said to perform significantly better or worse than the baseline.

Assuming that the recognition errors are independently distributed according to a binomial distribution, the probability of n_e can be expressed as

$$P(n_e = x) = b(x; n, p) = \binom{n}{p} p^x (1-p)^{(n-x)}.$$

The mean value and the variance of this distribution are then given by

$$\begin{aligned} \mu &= np \\ \sigma^2 &= np(1-p). \end{aligned}$$

If the values n and p are sufficiently large (a general rule of thumb is $np > 5$ and $n(p-1) > 5$), this distribution can be approximated by a normal distribution $\mathcal{N}(\mu, \sigma)$. Using this approximation the general normal random variable Z is defined as

$$Z = \frac{n_e - \mu}{\sigma}.$$

It is then possible to find the numbers $-z$ and z between which Z lies with a probability of $1 - \alpha$

$$P(-z < Z < z) = (1 - \alpha) = 0.95.$$

The number z can then be found from the inverse of the cumulative normal distribution function $\Phi(z)$

$$\begin{aligned} \Phi(z) = P(Z \leq z) = 1 - \frac{\alpha}{2} = 0.975 \\ z_{0.975} = \Phi^{-1}(0.975) = 1.96. \end{aligned}$$

Including this z-value in the confidence interval and inserting np for μ and $\sqrt{np(1-p)}$ for σ we get

$$\begin{aligned} P(-1.96 < \frac{n_e - np}{\sqrt{np(1-p)}} < 1.96) = (1 - \alpha) \\ P(np - 1.96\sqrt{np(1-p)} < n_e < np + 1.96\sqrt{np(1-p)}) = (1 - \alpha). \end{aligned}$$

Substituting the lower and upper bounds, c_1 and c_2, with $c_1-0.5$ and $c_2+0.5$ for continuity correction we get

$$P(np-0.5-1.96\sqrt{np(1-p)} < n_e < np+0.5+1.96\sqrt{np(1-p)}) = (1-\alpha).$$

It can now be shown [107] that the confidence interval $P(a_1(n_e) < p < a_2(n_e))$ for the true error probability p can be found by calculating the value of the two functions a_1 and a_2 as follows

$$\begin{aligned} a_1(n_e) &= \frac{n_e - 0.5 + 0.5z_{0.975}^2 - z_{0.975}\sqrt{0.25z_{0.975}^2 + \frac{(n_e-0.5)(n-n_e+0.5)}{n}}}{(n+z_{0.975}^2)} \\ a_2(n_e) &= \frac{n_e + 0.5 + 0.5z_{0.975}^2 + z_{0.975}\sqrt{0.25z_{0.975}^2 + \frac{(n_e+0.5)(n-n_e-0.5)}{n}}}{(n+z_{0.975}^2)} \end{aligned} \tag{A.1}$$

Assuming that the recognition results of the three test sets described in Chapter 4.2.3 are independent, we can use the total number of utterances in the three test sets $n = 3875$ and Equation (A.1) to calculate the confidence interval of the experiments conducted in this dissertation. The 95% confidence interval is illustrated in Figure A.1 and in Table A.1 for selected error rates.

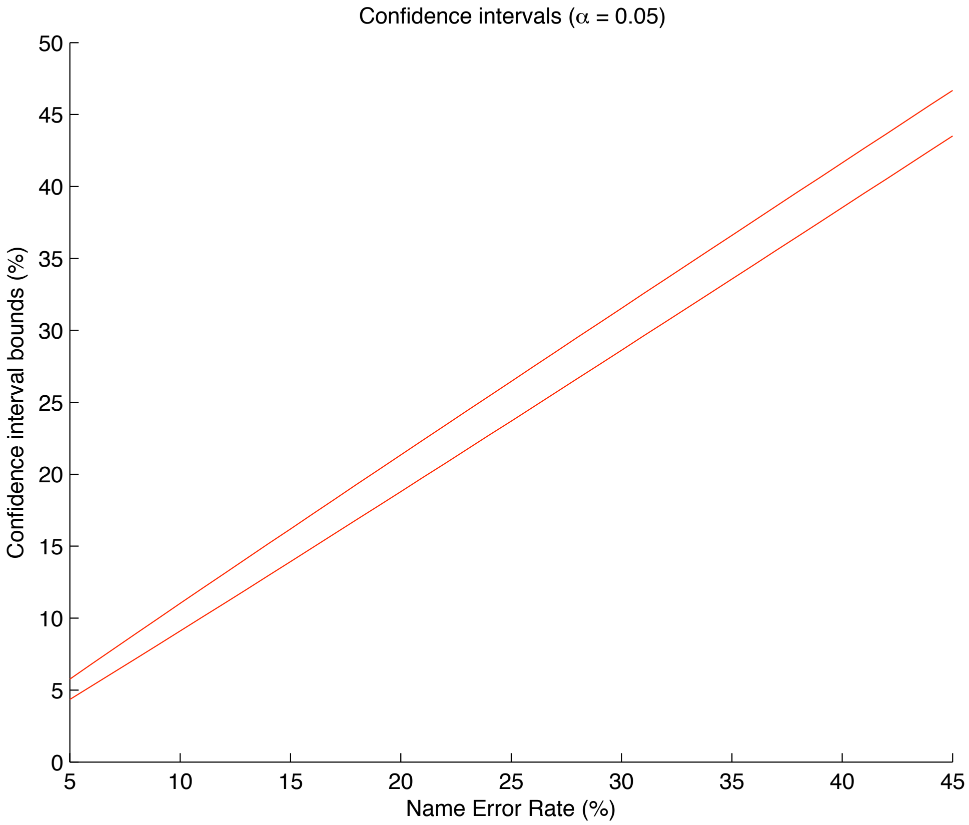

Figure A.1: *Confidence bounds for* $\alpha = 0.05$ *and* $n = 3875$.

NER	Interval	Low (a_1)	High (a_2)
5%	1.40	4.35%	5.76%
6%	1.53	5.29%	6.82%
7%	1.64	6.24%	7.88%
8%	1.74	7.19%	8.93%
9%	1.83	8.14%	9.98%
10%	1.92	9.10%	11.02%
11%	2.00	10.06%	12.06%
12%	2.08	11.02%	13.10%
13%	2.15	11.99%	14.14%
14%	2.22	12.96%	15.17%
15%	2.28	13.93%	16.20%
16%	2.34	14.90%	17.24%
17%	2.40	15.87%	18.26%
18%	2.45	16.84%	19.29%
19%	2.50	17.82%	20.32%
20%	2.55	18.80%	21.34%
21%	2.60	19.77%	22.37%
22%	2.64	20.75%	23.39%
23%	2.68	21.73%	24.41%
24%	2.72	22.71%	25.43%
25%	2.76	23.70%	26.45%
26%	2.79	24.68%	27.47%
27%	2.83	25.66%	28.49%
28%	2.86	26.35%	29.51%
29%	2.89	27.63%	30.52%
30%	2.92	28.62%	31.54%
31%	2.94	29.61%	32.55%
32%	2.97	30.60%	33.57%
33%	2.99	31.59%	34.58%
34%	3.01	32.58%	35.59%
35%	3.03	33.57%	36.60%
36%	3.05	34.56%	37.61%
37%	3.07	35.55%	38.62%
38%	3.09	36.54%	39.63%
39%	3.10	37.54%	40.64%
40%	3.12	38.53%	41.65%
41%	3.13	39.53%	42.65%
42%	3.14	40.52%	43.66%
43%	3.15	41.52%	44.67%
44%	3.16	42.52%	45.67%
45%	3.16	43.51%	46.68%

Table A.1: *Confidence intervals for $\alpha = 0.05$ and for $n = 3875$.*

Bibliography

[1] Line Adde and Torbjørn Svendsen, "NameDat: A Database of English Proper Names Spoken by Native Norwegians," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[2] Line Adde and Torbjørn Svendsen, "On the Use of Discriminative and Non-Discriminative Pronunciation Priors in Pronunciation Variation Modeling of Non-Native Proper Names," in *Proceedings of the IEEE Workshop on Spoken Language Technology*, Berkeley, California, USA, December 2010.

[3] Line Adde, Bert Réveil, Jean-Pierre Martens, and Torbjørn Svendsen, "A Minimum Classification Error approach to pronunciation variation modeling of non-native proper names," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, Makuhari, Japan, September 2010, pp. 2282–2285.

[4] Line Adde and Torbjørn Svendsen, "Pronunciation Variation Modeling of Non-Native Proper Names by Discriminative Tree Search," in *In the Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, 2011.

[5] Stephen B. Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357–366, 1980.

[6] John Makhoul, "Spectral linear prediction: Properties and applications.," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, pp. 283–296, 1975.

[7] Hynek Hermansky, “Perceptual linear predictive (PLP) analysis of speech.,” *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990.

[8] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing*, Prentice Hall, 2001.

[9] Lawrence R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *In Proceedings of the IEEE*, 1989.

[10] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, Cambridge University Engineering Department, 2006.

[11] Wu Chou, “Minimum Classification Error Approach in pattern recogntion,” in *Pattern Recognition in Speech and Language Processing (W. Chou and H.-H. Juang eds.).* CRC Press, 2003.

[12] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, Wiley-Interscience, 2001.

[13] E.T. Jaynes, “Information Theory and Statistical Mechanics,” *Physical Review*, vol. 106, pp. 620–630, 1957.

[14] Ronald Rosenfeld, “A maximum entropy approach to adaptive statistical language modelling,” *Computer Speech & Language*, vol. 10, pp. 187–228, 1996.

[15] Stephen Della Pietra, Vincent Della Pietra, Robert Mercer, and Salim Roukos, “Adaptive Language Modeling Using Minimum Discriminant Estimation,” in *In proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1992)*, 1992.

[16] Oliver Bender, Klaus Macherey, Franz Josef Och, and Hermann Ney, “Comparison of Alignment Templates and Maximum Entropy Models for Natural Language Understanding,” in *In proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, 2003.

[17] Wolfgang Macherey and Hermann Ney, “A Comparative Study on Maximum Entropy and Discriminative Training for Acoustic Modeling in Automatic Speech Recognition,” in *In Proceedings of the*

European Conference on Speech Communication and Technology (Eurospeech 2003, 2003.

[18] Hong-Kwang Jeff Kuo and Yuqing Gao, "Maximum entropy direct models for speech recognition," in *In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2003)*, 2003.

[19] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, vol. 22, pp. 39–71, March 1996.

[20] J. N. Darroch and D. Ratcliff, "Generalized Iterative Scaling for Log-Linear Models," *The Annals of Mathematical Statistics*, vol. 43, pp. 1470–1480, 1972.

[21] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty, "Inducing Features of Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 380–393, 1997.

[22] J. Benson, Steven and J. Moré, Jorge, "A Limited Memory Variable Metric Method in Subspaces and Bound Constrained Optimization Problems," Tech. Rep., Preprint ANL/ACSP909–0901 Argonne National Laboratory, 2002.

[23] Robert Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *In Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, 2002.

[24] Biing-Hwang Juang Juang and Shigeru Katagiri, "Discriminative learning for minimum error classification [pattern recognition]," *IEEE Transactions on Signal Processing*, vol. 40, pp. 3043–3054, 1992.

[25] Hong-Kwang Jeff Kuo, Eric Fosler-Lussier, Hui Jiang, and Chin-Hui Lee, "Discriminative training of language models for speech recognition," in *Proceedings of the International Conferance on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, 2002.

[26] Vladimir Magdin and Hui Jiang, "Discriminative training of n-gram language models for speech recognition via linear programming," in *In Proceedings of the IEEE Workshop on Automatic Speech Recognition Understanding (ASRU 2009)*, 2009.

[27] Oriol Vinyals, Li Deng, Dong Yu, and Alex Acero, "Discriminative Pronunciation Learning Using Phonetic Decoder and Minimum-Classification-Error Criterion," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'09)*, Taipei, Taiwan, 19-24 April 2009.

[28] ChiShi Liu, ChinHui Lee, Wu Chou, BiingHwang Juang, and Aaron E. Rosenberg, "A study on minimum error discriminative training for speaker recognition," *Journal of the Acoustical Society of America* , vol. 97, pp. 637–648, 1997.

[29] Filipp Korkmazskiy and Biing-Hwang Juang, "Discriminative adaptation for speaker verification," in *In Proceedings of The Fourth International Conference on Spoken Language Processing (ICSLP 1996)*, 1996.

[30] Biing-Hwang Juang, Wu Chou, and Chin-Hui Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, May 1997.

[31] Robert Eklund and Anders Lindström, "Xenophones: An investigation of phone set expansion in Swedish and implications for speech recognition and speech synthesis," *Speech Communication*, vol. 35, no. 1-2, pp. 81–102, 2001.

[32] Susan Fitt, "The Pronunciation Of Unfamiliar Native And Non-Native Town Names," in *Proceedings of the Fourth European Conference on Speech Communication and Technology (EUROSPEECH '95)*, Madrid, Spain, 1995, pp. 2227–2230.

[33] Isabel Trancoso, C'eu Viana, Isabel Mascarenhas, and Carlos Teixeira, "On deriving rules for nativised pronunciation in navigation queries," in *Proceedings of the Sixth European Conference on Speech Communication and Technology (EUROSPEECH'99)*, 1999.

[34] C.A. Kamm, K.-M. Yang, C.R. Shamieh, and S. Singhal, "Speech recognition issues for directory assistance applications," in *Proceedings of the IEEE Workshop on Interactive Voice Technology for Telecommunications Applications.*, 1994.

[35] Yuqing Gao, Bhuvana Ramabhadran, Julian Chen, Hakan Erdo, and Michael Picheny, "Innovative Approaches for Large Vocabulary Name

Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, 2001.

[36] Michael Meyer and Hermann Hild, "Recognition of spoken and spelled proper names," in *Proceedings of the Fifth European Conference on Speech Communication and Technology (EUROSPEECH 1997)*, 1997.

[37] Hermann Hild and Alex Waibel, "Recognition of spelled names over the telephone," in *Proceedings of the Fourth International Conference on Spoken Language (ICSLP 1996)*, 1996.

[38] Andreas Kellner, Bernd Rueber, and Hauke Schramm, "Strategies for name recognition in automatic directory assistance systems," in *Proceedings of IEEE 4th Workshop Interactive Voice Technology for Telecommunications Applications (IVTTA 1998)*, 1998.

[39] Sameer R. Maskey, Michiel Bacchiani, Brian Roark, and Richard Sproat, "Improved name recognition with meta-data dependent name networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04).*, 2004.

[40] Bhuvana Ramabhadran, Olivier Siohan, and Geoffrey Zweig, "Use of Metadata to Improve Recognition of Spontaneous Speech and Named Entities," in *Proceedings of the 8th International Conference on Spoken Language Processing (INTERSPEECH 2004)*, 2004.

[41] Frédéric Béchet, Renato de Mori, and Gérard Subsol, "Very large vocabulary proper name recogntion for directory assistance," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2001)*, 2001.

[42] Frédéric Béchet, Renato de Mori, and Gérard Subsol, "Dynamic generation of proper name pronunciations for directory assistance," in *Proceedings of the IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP 2002)*, 2002.

[43] Michael Harris Cohen, *Phonological Structures for Speech Recognition*, Ph.D. thesis, University of California, Berkeley, 1989.

[44] Qian Yang and Jean-Pierre Martens, "On the importance of exception and cross-word rules for the data-driven creation of lexica for ASR," in *Proceedings IEEE ProRisk (Veldhoven)*, 2000.

[45] Judith M. Kessens, Catia Cucchiarini, and Helmer Strik, "A data-driven method for modeling pronunciation variation," *Speech Communication*, vol. 40, pp. 517–534, 2003.

[46] Don McAllaster, Larry Gillick, Francesco Scattone, and Mike Newman, "Fabricating conversational speech data with acoustic models: A program to examine model-data mismatch," in *Proceedings of the fifth International Conference on Spoken Language Processing (ICSLP 1998)*, 1998.

[47] Murat Saraçlar, Harriet Nock, and Sanjeev Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech & Language*, vol. 14, pp. 137–160, 2000.

[48] Helmer Strik, "Pronunciation adaptation at the lexical level," in *Proceedings of ISCA ITRW on Adaptation Methods for Speech Recognition*, 2001.

[49] Ingunn Amdal, *Learning pronunciation variation. A data-driven approach to rule-based lexicon adaptation for automatic speech recognition*, Ph.D. thesis, Norwegian University of Science and Technology (NTNU), 2002.

[50] Eric Fosler-Lussier and Gethin Williams, "Not just what, but also when: Guided automatic pronunciation modeling for Broadcast News," in *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[51] Eric Fosler-Lussier, "Multi-Level Decision Trees for Static and Dynamic Pronunciation Models," in *Proceedings of the Sixth European Conference on Speech Communication and Technology (EUROSPEECH 1999)*, 1999.

[52] Eric Fosler-Lussier, "Contextual Word and Syllable Pronunciation Models," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 1999)*, 1999.

[53] Sam Bowman and Karen Livescu, "Modeling Pronunciation Variation with Context-Dependent Articulatory Feature Decision Trees," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)'*, 2010.

[54] Karen Livescu and James Glass, "Feature-based Pronunciation Modeling with Trainable Asynchrony Probabilities," in *Proceedings of*

the 8th International Conference on Spoken Language Processing (INTERSPEECH - ICSLP 2004), 2004.

[55] Michael Riley, William Byrne, Michael Finke, Sanjeev Khudanpur, Andrej Ljolje, John McDonough, Harriet Nock, Murat Saraclar, Charles Wooters, and George Zavaliagkos, "Stochastic pronunciation modelling from hand-labelled phonetic corpora," *Speec*, vol. 29, pp. 209–224, 1999.

[56] Silke Goronzy, Ralf Kompe, and Stefan Rapp, "Generating Non-Native Pronunciation Variants for Lexicon Adaptation," *Speech*, vol. 42, pp. 109–123, 2004.

[57] Yi-Ning Chen, Peng Liu, Jia-Li You, and Frank K. Soong, "Discriminative training for improving letter-to-sound conversion performance," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, 2008.

[58] Xiao Li, Asela Gunawardana, and Alex Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding (ASRU 2007)*, 2007.

[59] Ibrahim Badr, Ian McGraw, and James Glass, "Learning New Word Pronunciations from Spoken Examples ," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, 2010.

[60] I. Lee Hetherington, "An Efficient Implementation of Phonological Rules using Finite-State Transducers," in *7th European Conference on Speech Communication and Technology (EUROSPEECH 2001*, 2001.

[61] Timothy J. Hazen, I. Lee Hetherington, Han Shu, and Karen Livescu, "Pronunciation modeling using finite-state transducer representation," *Speech Communication*, vol. 46, pp. 189–203, 2005.

[62] I. Trancoso, D. Caserio, C. Viana, F. Silva, and I. Mascarenhas, "Pronunciation modeling using finite state transducers," in *Proceedings og the 15th International Congress of Phonetic Sciences (ICPhS2003)*, 2003.

[63] Karen Livescu and James Glass, "Lexical modeling of non-native speech for automatic speech recognition," in *Proceedings of the IEEE*

International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000), 2000.

[64] Nick Cremelie and Jean-Pierre Martens, “In search of better pronunciation models for speech recognition,” *Speech Communication*, vol. 29, pp. 115–136, 1999.

[65] Qian Yang and Jean-Pierre Martens, “Data-driven lexical modeling for pronunciation variations for ASR,” in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP 2000)*, 2000.

[66] Tilo Slobada and Alex Waibel, “Dictionary learning for spontaneous speech recognition,” in *Proceedings of the fourth International Conference on Spoken Language Processing (ICSLP 96)*, 1996.

[67] D. Torre, L. Villarrubia, L. Hernandez, and J.M. Elvira, “Automatic alternative transcription generation and vocabulary selection for flexible word recognizers,” in *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1997)*, 1997.

[68] Gethin Williams and Steve Renals, “Confidence Measures for Evaluating Pronunciation Models,” in *Proceedings of the ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, 1998.

[69] Trym Holter and Torbjørn Svendsen, “Maximum likelihood modelling of pronunciation variation,” *Speech Communication*, vol. 29, no. 2–4, pp. 77–191, November 1999.

[70] Mirjam Wester and Eric Fosler-Lussier, “A comparison of data-derived and knowledge-based modeling of pronunciation variation,” in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP 2000)*, 2000.

[71] Eric Fosler-Lussier, Ingunn Amdal, and Hong-Kwang Je Kuo, “A framework for predicting speech recognition errors,” *Speech Communication*, vol. 46, pp. 153–170, 2005.

[72] Preethi Jyothi and Eric Fosler-Lussier, “Discriminative Language Modeling Using Simulated ASR Errors,” in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, 2010.

[73] Eric Fosler-Lussier and Nelson Morgan, "Effects of speaking rate and word frequency on pronunciations in convertional speech," *Speech C*, vol. 29, pp. 137–158, 1999.

[74] Michael Finke and Alex Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH 1997*, 1997.

[75] Eric Fosler-Lussier, *Dynamic pronunciation models for automatic speech recognition*, Ph.D. thesis, University of California, Berkeley, 1999.

[76] Stefan Schaden, "Generating Non-Native Pronunciation Lexicons by Phonological Rules," in *Proceedings of the 15th International Conferance of Phonetic Sciences (ICPhS 2003)*, 2003.

[77] Stefan Schaden, "Rule-based lexical modeling of foreign accented pronunciation variants.," in *Proceedings 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, 2003.

[78] B. Ramabhadran, L.R. Bahl, P.V. deSouza, and M. Padmanabhan, "Acoustics-only based automatic phonetic baseform generation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.

[79] Françoise Beaufays, Ananth Sankar, Shaun Williams, and Mitch Weintraub, "Learning Name Pronunciations in Automatic Speech Recognition Systems," in *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2003)*, 2003.

[80] The Onomastica Consortium, "The Onomastica Interlanguage Pronunciation Lexicon," in *Proceedings of the Fourth European Conference on Speech Communication and Technology (EUROSPEECH '95)*, Madrid, Spain, 1995, pp. 829–832.

[81] Nick Cremelie and Louis ten Bosch, "Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters," in *Proceedings of the ISCA ITRW on Adaptation Methods for Speech Recognition*, 2001.

[82] Bert Réveil, Jean-Pierre Martens, and Bart D'hoore, "How speaker tongue and name source language affect the automatic recognition of

spoken names," in *Proceedings of Interspeech 2009 the 10th Annual Conference of the International Speech Communication Association*, 2009.

[83] Benoît Maison, F. Chen, Stanley, and S. Cohen, Paul, "Pronunciation variation modeling for names of foreign origin," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2003)*, 2003.

[84] Qian Yang, Jean-Pierre Martens, Nanneke Konings, and Henk van den Heuvel, "Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, 2006, pp. 287–292.

[85] Henk van den Heuvel, Bert Réveil, and Jean-Pierre Martens, "Pronunciation-based ASR for names," in *Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech 2009)*, Brighton, UK, 2009.

[86] Helmer Strik and Catia Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," *Speech Communication*, vol. 29, pp. 225–246, 1999.

[87] Judith M. Kessens, Mirjam Wester, and Helmer Strik, "Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation," *Speech Communication*, vol. 29, no. 2–4, pp. 193–207, November 1999.

[88] Hauke Schramm and Peter Beyerlein, "Discriminative Optimization of the Lexical Model," in *ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA'02)*, 2002, pp. 105–110.

[89] Thomas Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, pp. 171–188, 2005.

[90] Abhinav Sethy, Shrikanth Narayanan, and S. Parthasarthy, "A split lexicon approach for improved recognition of spoken names," *Speech Communication*, vol. 48, pp. 1126–1136, 2006.

[91] Taeyoon Kim, Sunmee Kang, and Hanseok Ko, "An effective acoustic modeling of names based on model induction ," in *Proceedings of*

the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000), 2000.

[92] Rebecca Bates, Mari Ostendorf, and Richard Wright, “Symbolic phonetic features for modeling of pronunciation variation,” *Speech Communication*, vol. 49, pp. 83–97, 2007.

[93] Preethi Jyothi, Karen Livescu, and Eric Fosler-Lussier, “Lexical access experiments with context-dependent articulatory feature-based models,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, 2011.

[94] Mari Ostendorf, “Moving Beyond the ‘Beads-On-A-String’ Model of Speech,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 1999)*, 1999.

[95] Murat Saraçlar, *Pronunciation Modeling for Conversational Speech Recognition*, Ph.D. thesis, John Hopkins University, 2000.

[96] M. Bacchiani and M. Ostendorf, “Joint lexicon, acoustic unit inventory and model design,” *Speech Communication*, vol. 29, pp. 99–114, 1999.

[97] K. Beulen, S. Ortmanns, A. Eiden, S. Martin, L. Welling, J. Overmann, and H. Ney, “Pronunciation modelling in the RWTH large vocabulary speech recognizer,” in *Proceedings of the ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, 1998.

[98] Trym Holter, *Maximum likelihood modelling of pronunciation in automatic speech recognition*, Ph.D. thesis, Norwegian University of Science and Technology, 1997.

[99] Zhirong Wang, Tanja Schultz, and Alex Waibel, “Comparison of acoustic model adaptation techniques on non-native speech,” in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’03)*, 2003.

[100] Lui Wai Kat and Pascale Fung, “MLLR-based accent model adaptation without accented data,” in *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, 2000.

[101] Tanja Schultz and Alex Waibel, "Polyphone decision tree specialization for language adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000)*, 2000.

[102] Georg Stemmer, Elmar Nöth, and Heinrich Niemann, "Acoustic Modeling of Foreign Words in a German Speech Recognition System," in *Proceedings Eurospeech (EUROSPEECH 2001)*, 2001.

[103] Frederik Stouten and Jean-Pierre Martens, "Recognition of foreign names spoken by native speakers," in *Proceedings of the 8th annual conference of the international speech communication association (INTERSPEECH 2007)*, 2007.

[104] Frederik Stouten and Jean-Pierre Martens, "Dealing with cross-lingual aspects in spoken name recognition," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition Understanding, (ASRU 2007)* , 2007.

[105] Qian Yang, *Data-driven approaches to pronunciation variation modeling for automatic speech recognition*, Ph.D. thesis, University of Ghent, 2005.

[106] Henk van den Heuvel, Jean-Pierre Martens, Bart D'hoore, Kristof D'hanens, and Nanneke Konings, "The AUTONOMATA Spoken Names Corpus," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08)*, 2008.

[107] Erik Harborg, *Hidden Markov Models applied to automatic speech recognition.*, Ph.D. thesis, NTH (Norwegian Institute of Technology), 1990.