Improving Grapheme-based speech recognition through P2G transliteration

W.D. Basson

A dissertation submitted in fulfilment of the requirements for the degree *Master of Science in Computer Science* at the Vaal Triangle Campus of the North-West University

Supervisor: Prof. M.H. Davel

May 2014

Acknowledgements

A special thanks to my supervisor, Marelie Davel, for providing me with the needed guidance, encouragement and support.

Thank you to the Multilingual Speech Technologies (MuST) research team.

I am also very grateful to family and friends for being supportive and understanding.

Sufficient processing resources in support of this work were provided by North-West University's high performance computing facility [1] and the Centre for High Performance Computing [2].

[1] http://www.nwu.ac.za/content/hpc-high-performance-computing
[2] http://www.chpc.ac.za

Disclaimer

Financial support was provided by the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore the NRF does not accept any liability in regard thereto.

This work was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Defense U.S. Army Research Laboratory contract number W911NF-12-C-0013. The U.S. government is authorised to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation on the work. Disclaimer: The views and conclusions contained herein are those of the author(s) and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of IARPA, DoD/ARL, or the U.S. government.

Opsomming

Grafeem-gebaseerde spraakherkenningstelsels is vinniger en eenvoudiger om te ontwikkel as foneem-gebaseerde stesels, maar lewer tipies swakker herkenningsresultate. Ons gebruik Afrikaanse spraakherkenning as gevallestudie, en analiseer die redes vir die verskil tussen grafeem-gebaseerde en foneem-gebaseerde herkenningsakkuraatheid. Daarna ontwikkel ons 'n nuwe tegniek om grafeem-gebaseerde herkenning te verbeter.

Tydens 'foneem-na-grafeemtransliterasie' transformeer ons die oorspronklike ortografie van woorde met onreëlmatige uitsprake na 'n geïdealiseerde ortografie. Ons vind dat deur 'n klein aantal onreëlmatige woorde te hanteer, ons die herkenningsakkuraatheid van grafeem-gebaseerde stelsels kan verbeter.

'n Woordkategorie-gebaseerde ontleding van spraakherkenningsakkuraatheid toon dat foneem-na-grafeemtransliterasie daarin slaag om onreëlmatige kategorieë, spesifiek dié waarin grafeem-gebaseerde stelsels gewoonlik swak vertoon, te verbeter, en dat hierdie kategorieë geïdentifiseer kan word alvorens die afrigting van herkenningstelsels.

Ons analiseer wanneer kategorie-gebaseerde foneem-na-grafeemtransliterasie voordelig is, evalueer vertoning in 'n tweede taal (Viëtnamees) en bespreek hoe die tegniek prakties geïmplementeer kan word.

Summary

Grapheme-based speech recognition systems are faster to develop, but typically do not reach the same level of performance as phoneme-based systems. Using Afrikaans speech recognition as a case study, we first analyse the reasons for the discrepancy in performance, before introducing a technique for improving the performance of standard grapheme-based systems.

It is found that by handling a relatively small number of irregular words through phoneme-to-grapheme (P2G) transliteration – transforming the original orthography of irregular words to an 'idealised' orthography – grapheme-based accuracy can be improved. An analysis of speech recognition accuracy based on word categories shows that P2G transliteration succeeds in improving certain word categories in which grapheme-based systems typically perform poorly, and that the problematic categories can be identified prior to system development.

An evaluation is offered of when category-based P2G transliteration is beneficial and methods to implement the technique in practice are discussed. Comparative results are obtained for a second language (Vietnamese) in order to determine whether the technique can be generalised.

Keywords: automatic speech recognition, phoneme-to-grapheme, transliteration, grapheme-based ASR, Afrikaans, Vietnamese.

Contents

List of Figures

List of Tables

Abbreviations

ASR	**A**utomatic **S**peech **R**ecognition
CV	**C**onsonant **V**owel
D&R	**D**efault and **R**efine
DFT	**D**iscrete **F**ourier **T**ransform
EM	**E**xpectation **M**aximisation
FFR	**F**oreign **F**alse **R**ecognitions
FRR	**F**alse **R**ecognition **R**ate
HMM	**H**idden **M**arkov **M**odel
HTK	**H**idden Markov Model **T**ool **K**it
JSM	**J**oint **S**equence **M**odel
KL-HMM	**K**ullbach-**L**eibler Divergence-based **H**idden **M**arkov **M**odel
LSM	**L**ongest **S**tring **M**atching
MER	**M**atching **E**rror **R**ate
MFCC	**M**el **F**requency **C**epstral **C**oefficient
P2G	**P**honeme-to-**G**rapheme
PbA	**P**ronunciation **b**y **A**nalogy
PDP	**P**hone-based **D**ynamic **P**rogramming
PER	**P**honeme **E**rror **R**ate
SLM	**S**tatistical **L**anguage **M**odel
WER	**W**ord **E**rror **R**ate

In memory of my grandfather, Willem D. Basson

Chapter 1

Introduction

Speech can be seen as one of the most fundamental forms of human communication. Through it, we are able to communicate complex ideas and exchange vast amounts of information fast and effortlessly. The ability to recognise speech automatically with a machine has manifold applications and the development of automatic speech recognition (ASR) systems is therefore of great importance both scientifically and economically[1, 2].

Traditionally, the development of ASR systems is highly dependent on the availability of various linguistic resources [3]. Grapheme-based ASR – an approach that does not require language-specific phonetic information – can reduce the complexity of required resources. This study focuses on improving grapheme-based ASR by identifying and addressing some of its shortcomings.

1.1 Contextualisation

In ASR systems, words are traditionally represented as a sequence of acoustic sub-word units such as phonemes [4]. The mapping from these sub-word units to words is usually contained in some form of pronunciation dictionary. Since the overall performance of ASR systems is strongly dependent on the accuracy of the pronunciation dictionary, best results are usually obtained with hand-crafted dictionaries [2]. Development of these dictionaries is a time-consuming, costly and labour-intensive process, often requiring expert knowledge [2, 5]. If expert knowledge is unobtainable, manually developed or statistical grapheme-to-phoneme (G2P) rules can be used to generalise from small data sets [4]; however, these methods are not available for many languages and typically produce less accurate results than manually developed phoneme-based dictionaries.

The development of pronunciation dictionaries is even further constrained in resource-scarce environments. In resource-scarce environments expert knowledge might not readily be available and even simple linguistic resources such as word lists and phoneme sets may be hard to come by [3].

Earlier work in grapheme-based systems has shown that for regular languages – languages that exhibit a fairly close relationship between graphemes and phonemes – phoneme-based dictionary development may be unnecessary, and that the letters of the word can be used directly as the acoustic sub-word units to model [4, 5, 6]. Using grapheme-based sub-word units eliminates the need for expert knowledge and saves time and cost, results in a significantly simplified lexicon definition and can result in relatively noise-free pronunciation models [7].

The regularity of a language can be measured based on G2P consistency: using the average accuracy that is obtained at a specific dictionary size when extracting G2P rules [8]. According to this measure, languages vary considerably, from highly irregular languages such as English, to highly regular languages such as Spanish or Vietnamese, with Afrikaans – the language used as a case study – being of medium regularity [9].

Some of the earliest work on grapheme-based speech recognition proposes using polygraphs, that is, letter-based units constructed from the orthographic word form with arbitrary length left and right contexts as sub-word units [6]. More recent work includes context-dependent grapheme-based recognisers [4]. In other grapheme-based approaches, acoustic data is leveraged for extra information during lexical development. [5] proposes a decision tree based on graphemic acoustic sub-word units together with either phonetic or automatically generated questions. A Kullback-Leibler divergence-based hidden Markov model (KL-HMM) probabilistic lexical modelling approach is used in [10]. (See Chapter 2 for more detail.)

However, grapheme-based ASR does not always reach the same level of performance as phoneme-based ASR [11]. This is expected in situations where grapheme-based ASR has difficulty modelling the orthographical relationship to acoustics, because of the irregular orthographical representation of words. Apart from trying to model a language that is itself irregular, irregular words can also originate from (1) irregular in-language words (such as abbreviations and spelled out words), and (2) code-switched words from a foreign language with a G2P relationship in conflict with that of the target language. (Proper names can form part of both the aforementioned categories.) In this study we explore the impact of words from specific word categories that exhibit a more irregular relationship between graphemes and phonemes.

1.2 Problem statement

Grapheme-based systems are faster to develop, but typically do not reach the same level of performance as a phoneme-based system that was developed using a manually verified dictionary. We hypothesise that this discrepancy in performance is due to very specific word categories (for example, spelled out words, acronyms, proper names and foreign words) that all tend to have highly irregular relationships between graphemes and phonemes, confusing both G2P-based and grapheme-based systems. This is compounded in multilingual environments (such as South Africa), where proper name and code-switched pronunciations tend to be irregular.

We are interested in developing appropriate techniques to deal with words from these irregular categories at the lexical level (that is, by changing the pronunciation dictionary rather than incorporating information from audio data).

1.3 Research questions

Given the above-mentioned problem statement, the following research questions are formulated:

- How do different word categories affect the performance of grapheme-based and phoneme-based ASR systems, respectively?
- Can the orthographic form of irregularly spelled words be adapted for better integration of these words in grapheme-based ASR systems?
- What effect do different spelling adaptation techniques have on ASR accuracy?
- What are the implications of these results for ASR development in resource-scarce environments?

1.4 Approach

This study follows a data-centric approach in which we aim to define, develop and evaluate a new technique for dealing with idiosyncratic pronunciations in grapheme-based ASR systems. We compare grapheme- and phoneme-based ASR performance and categorise the words responsible for recognition errors. A large percentage of these words are found to share a single characteristic: they all have an irregular G2P relationship.

Grapheme-based ASR performance is then improved by 'regularising' words from these problematic categories.

In the literature review, we first establish a foundation regarding ASR systems and their components, focusing on pronunciation modelling and statistical model training and adaptation. We also review current approaches to grapheme-based ASR and related work.

During the data preparation phase we extract and develop all the resources necessary to achieve our aim. First, a well-balanced subset is selected from an existing transcribed audio corpus. This becomes the dataset. We then hand-craft a pronunciation dictionary for all the words contained in the dataset and verify the quality of the pronunciations; at the same time assigning each of them to a category. For the same word list we also create a very simplistic grapheme-based dictionary and a state-of-the art G2P-based dictionary.

Using the newly selected dataset, we follow a standardised approach to develop three ASR systems: (1) a phoneme-based system, (2) a G2P-based system, and (3) a grapheme-based system. (The phoneme-based system uses the hand-crafted dictionary, the G2P-based system uses the G2P-based dictionary, and the grapheme-based system uses the grapheme-based dictionary.) These systems are used in initial experimentation where we compare them in terms of word error rate (WER) and through category-based analysis determine which word categories are responsible for grapheme-based performance degradation.

The first set of experiments confirms our intuition that a small set of word categories is responsible for most of the performance degradation observed. In order to analyse this phenomenon further, the pronunciation dictionary is first manually verified by two reviewers and word categories are double-checked. In addition, ASR system parameters are optimised. Using the newly reviewed resources and optimised parameter settings, baseline results for the rest of the study are established.

To improve grapheme-based performance, we use phoneme-to-grapheme (P2G) rules to transform the orthography of irregular words, and then use these 're-spelled' words when building the improved grapheme-based system. Simplistic P2G rules are trained using only words with a largely regular G2P relationship. These rules are not capable of capturing too much spelling detail and can be said to model the default G2P relationship of the language, in this case, regular Afrikaans words. When applied to the pronunciation (phoneme string) of an irregular word, these rules will transform the orthography of the word to a more regular spelling. Though pronunciations are still needed for these irregular words, typically their frequency of occurrence is lower than that of regular

words, and it is hoped that a much smaller investment in manual lexicon development will achieve the same level of accuracy as that of a system developed using a hand-crafted phonemic dictionary.

We then determine how these rules influence grapheme-based ASR performance and when (and when not) to apply them, among others taking word categories into account. Finally, we establish whether our results generalise to a different language.

1.5 Chapter overview

The rest of this study is organised as follows: Chapter 2 gives a detailed overview of relevant literature pertaining to grapheme-based ASR and related work. Chapter 3 describes the data preparation process, including data selection, development of pronunciation dictionaries and categorisation of word lists. A comparison between grapheme-based and phoneme-based ASR systems is performed and word categories analysed in Chapter 4. Chapter 5 details and evaluates the category-based P2G transliteration technique. Chapter 6 concludes with a summary of our findings and observations, alongside suggestions for future research.

Chapter 2

Literature review

In this chapter we establish a foundation for ASR systems and their components, focusing on pronunciation modelling and statistical model training. We also provide an overview of relevant literature pertaining to grapheme-based ASR.

Given the large number of different approaches and techniques being used and actively studied in ASR-related research, it is not the intention of this literature review to provide a broad overview of the field. Rather, we will focus only on the main concepts and techniques referred to in the remainder of this dissertation.

2.1 Introduction

ASR involves the processing of acoustic input via a machine and generating some form of output, usually a textual transcription. A typical ASR system is a combination of various probabilistic models, namely a language model, an acoustic model and a pronunciation model [12]. The following section describes the fundamental structure of ASR systems in terms of these probabilistic models, before describing the pronunciation model – the focus of this study – in more detail.

The rest of this chapter is organised as follows: a conceptual framework for ASR systems is described in Section 2.2. An overview of G2P conversion techniques is given in Section 2.3, followed by a brief overview of non-standard pronunciation modelling in Section 2.4. The basic concepts and motivations behind grapheme-based sub-word units are discussed in Section 2.5. Section 2.6 defines the concept of language regularity and the chapter concludes in Section 2.7 with a summary.

2.2 ASR systems

Following the description in [13], the speech recognition task can be summarised as follows:

For a language L, given an acoustic speech signal O, what is the most likely sentence to have been spoken?

If we convert a speech signal O into a sequence of speech vector observations:

$$O = o_1, o_2, o_3, ..., o_t, \tag{2.1}$$

where each o_i indicates a successive observation, the task of determining the most likely sequence of words $\hat{W}$ can be formulated as:

$$\hat{W} = \arg\max_{W \epsilon L} P(W|O), \tag{2.2}$$

that is, the word sequence $\hat{W}$ such that the probability of the word sequence $\hat{W}$, given a sequence of acoustic observations O, is largest.

We can simplify Equation 2.2 using Bayes' theorem:

$$\hat{W} = \arg\max_{W \epsilon L} \frac{P(O|W)P(W)}{P(O)}, \tag{2.3}$$

and since the observation sequence remains the same for each sentence that is evaluated, we can ignore the probability of the observation sequence *P(O)*, which leaves:

$$\hat{W} = \arg\max_{W \epsilon L} P(O|W)P(W). \tag{2.4}$$

Then the most likely sentence $\hat{W}$, given a sequence of observations O, is the sentence with the highest product of two probabilities, namely *P(W)*, the prior probability of word strings, and *P(O|W)*, the observation likelihood. The prior probability *P(W)* is typically estimated by a language model and the observation likelihood, *P(O|W)*, is calculated by the acoustic model. The pronunciation model provides the link between the language model and acoustic model by mapping each word to its sub-units (indirectly modelled by the acoustic model).

These models form the basic components of an ASR system.

2.2.1 ASR architecture

As introduced in Section 2.1, a typical ASR system can be viewed as a combination of three probabilistic models, namely a language model, an acoustic model and a pronunciation model. The language model estimates the probability of a sequence of words and the acoustic model estimates the likelihood of observing acoustic feature vectors, given a certain linguistic context, usually phonetic sub-word units. (Sub-word units refer to linguistic units smaller than the words they are representing, such as phonemes, subphones, or in the case of grapheme-based ASR systems, graphemes.) The pronunciation model indicates how words are split into sub-word units and is used to map the acoustic model to the language model.

To decode a spoken utterance, features from an input speech signal are matched against the acoustic models and the most likely sub-word units are then selected. By limiting possible matches to correspond to sequences of sub-word units, decoding is simplified. A language model then further simplifies the recognition process by constraining possible word sequences [14].

2.2.1.1 Feature extraction

Before a speech signal can be used by an ASR system, it must first be converted into a different representation that can be modelled more easily. The relevant information present in the signal (features) must first be extracted. Mel frequency cepstral coefficients (MFCCs) [15] and perceptual linear predictive features [16] are some of the most common feature representations in ASR. Following the description in [13], we describe feature extraction using MFCCs (as these features are used in the remainder of this study).

Firstly, to compensate for spectral tilt – the imbalance of energy in low and high frequency voiced segments – high frequencies in the speech signal are boosted. The boosted speech signal is converted into a sequence of equally spaced parameter vectors. Using a window of approximately 25 ms, the speech signal is split at equal intervals of approximately 10 ms to form frames.

Spectral features are extracted from these frames by separating the frequency components of a sound wave by performing a Discrete Fourier Transform (DFT). The results from the DFT – frequency band specific energy information – are used to calculate MFCCs by warping the frequencies onto the mel scale.

Next, the cepstrum is calculated using the inverse DFT. The cepstrum gives information about the position of the vocal tract, which is useful for detecting phonemes. From this

cepstrum, typically 12 cepstral coefficients combined with the energy from the frame are used. Finally, the delta and double deltas are usually calculated for the 12 cepstral coefficients and one energy coefficient. The 39 resulting MFCC features can then be used during acoustic modelling and decoding.

2.2.1.2 Acoustic models

The acoustic model *P(O|W)* computes the likelihood of observing spectral feature vectors, given a specific linguistic context. In the ASR systems we study, the acoustic model is built using an HMM. The basic theory behind HMMs was published between 1966 and 1972 and was first implemented in speech processing applications in the 1970s [14]. An HMM used in speech recognition can be parameterised by:

- a set of states corresponding to some linguistic sub-word units such as phonemes or subphones,
- a transition probability matrix representing the likelihood of moving from one state to another, and
- a set of observations likelihoods (emission probabilities) that express the probability of a feature vector being generated from a state.

A density estimation technique (often Gaussian mixture models) is then used to compute, for each HMM state, the likelihood of a given feature vector given this state. The acoustic model probabilities are combined with language model probabilities and used in a search algorithm to compute which sentence has the maximum probability, given a speech signal [13].

A word can be thought of as an HMM with the sub-parts of its pronunciation (typically consisting of phonemes) corresponding to states. (The mapping of how words are split into phonemes is provided by the pronunciation model.) For example, the word *state* can be represented by the four phonemes /s t @i t/ (using Speech Assessment Methods Phonetic Alphabet notation [17]).

Because of the large spectral variation of phonemes over time and the fact that phonemes are not of fixed length, they are typically modelled with more than one HMM state. The most common number of states is three: beginning, middle and end. In addition, a phoneme is not modelled in isolation, but multiple context-dependent models are created per phoneme. When a phoneme and its immediate left and right contexts are modelled together, this is referred to as a triphone. (Two contexts to the left and right would

create a quinphone.) Using a context-dependent triphone model, the word *state* can be represented by the four triphones /sil-s+t s-t+@i t-@i+t @i-t+sil/, with sil representing silence.

Modelling triphones instead of phonemes greatly increases the number of models and consequently causes a data scarcity problem. Context-dependent tying alleviates data scarcity by 'tying' together similar subphones and training only a single Gaussian model for each tied state [13].

2.2.1.3 Pronunciation models

As mentioned in the previous section, the pronunciation model provides the mapping for how words are split into their sub-word units, typically phonemes. A pronunciation model can be realised as a pronunciation dictionary consisting of a list of words together with their corresponding pronunciations, each pronunciation being represented by a string of linguistic sub-word units. Sub-word units need not be phonetic. They can be graphemic units such as polygraphs, or even single graphemes.

A pronunciation model is seldom a static component and often requires extension to be able to accommodate additional words. For phonemic dictionaries, this is done either manually, or, more typically, through G2P prediction, as described in more detail in Section 2.3. Grapheme-based pronunciation models are described in Section 2.5.

2.2.1.4 Language models

Statistical language models (SLMs) aim to capture the regularities of a given language in order to determine the most likely word, given a sequence of preceding words [18]. In other words, they capture the prior probability *P(W)* of word strings. By providing additional statistical knowledge during the recognition process they drastically improve ASR performance (lessening the total number of paths that need to be considered) [19]. The prior probability *P(W)* is the probability distribution over all possible sentences and can be estimated with an n-gram language model. An n-gram language model estimates the probability of a word w_N, given the preceding $N-1$ words $w_1, \ldots, w_{N-1}$.

How well a language model succeeds in modelling a specific language or domain can be measured in terms of perplexity [19], with lower perplexity being an indication of a better model. However, lower language model perplexity does not always result in lower WER.

2.3 G2P conversion

The expense of developing pronunciation dictionaries manually can be alleviated by utilising automatic G2P conversion, that is, using existing G2P rules to predict the phonemic transcriptions of a word based on its orthography. In other words, given a sequence of letters (graphemes) we seek the most likely phoneme sequence [20]. Though effective, G2P conversion based on manually developed rules still does not eliminate the need for expert knowledge and can be tedious and complex to develop. Various data-driven techniques exist for G2P conversion. Some of the techniques mentioned in literature include (1) decision trees [21, 22], (2) Bayesian (stochastic) techniques such as HMMs [23], (3) pronunciation by analogy (PbA) [24, 25], (4) neural networks [26], (5) instance-based learning [27, 28], (6) Default and refine (D&R) [9], and (7) joint sequence models (JSMs) [29].

The following sections describe the two G2P conversion methods used in this study, namely JSM and D&R.

2.3.1 Joint sequence models

Prior to training, G2P methods typically require alignments between letters and phonemes. JSMs infer these alignments implicitly by way of grapheme-phoneme joint-multigram models, or *graphones* [20]. A graphone is a paired sequence of graphemes and phonemes. The sequences need not be the same length.

To train JSMs, a unigram graphone model for a graphone set is inferred using the expectation maximisation (EM) algorithm. This context-independent unigram model is then used to segment a training corpus into graphones. Using this training corpus, maximum likelihood training of higher order M-gram models can be performed using EM. Models are smoothed using state-of-the-art Kneser-Ney smoothing [30].

The context size of graphones can be restricted to a specified maximum value. This value is typically the same for both graphemes and phonemes. Alternatively, grapheme and phoneme sequence limits can be set independently and a minimum value can be used in addition to the maximum value [29].

The range of a model is determined by (a) the context size of the graphones, and (b) the order M of the sequence model, that is, the context size of graphone sequences. In [29] it is observed that very low order M-gram models ($M \leq 2$) benefit from larger graphone sizes. Conversely, larger order M-gram models ($M \geq 4$) have higher accuracy with smaller graphone sizes.

To predict pronunciations for a word, the most likely graphone sequences, given the orthography of the word, is calculated.

The inverse task of G2P conversion is P2G conversion. Given a sequence of phonemes, we then want to know the most likely sequence of graphemes. JSMs use statistical models that are symmetric with regard to both conversion tasks, making them well suited for the purpose of this study.

2.3.2 Default and refine

D&R is built upon the notion of the 'default behaviour' between graphemes and phonemes observed in languages. The more regular the language, the stronger the concept of a 'default phoneme', that is, "a grapheme that is realised as a specific phoneme significantly more than any other phoneme" [9]. The G2P conversion task is modelled by a set of rules that captures the hierarchical default behaviour of a language. Each rule acts as a back-off to the next rule, starting with the most refined rule, and backing off to a less refined rule, if required.

Each rule consists of a grapheme with a left and right context. When predicting pronunciations for a word, individual graphemes together with their left and right contexts are compared against the grapheme-specific rules (starting at the most refined rule, only falling back to a more default rule if no applicable context is found) and the first matching rule is applied. Evaluated on English and Flemish [9], D&R performed better than algorithms with which it was compared (PbA, variations of dynamically expanding context and instance-based learning). D&R has the advantage that rules generated are humanly interpretable.

2.4 Modelling non-standard pronunciations

G2P rules trained on standard corpora are typically ill-suited to model words with non-standard pronunciations such as proper names, whose spelling can be archaic or of foreign origin, or whose pronunciations can be non-standard due to variation in speaker pronunciation [31].

Lexical modelling approaches (that do not take acoustic evidence into account) to address the complications in proper name pronunciation modelling can include: (1) foreign-language-specific G2P conversion [32, 33] and/or (2) phoneme-to-phoneme (P2P) conversion [31, 34]. If the language of origin of a proper name can be identified, its pronunciation can be modelled by a set of G2P rules from the same language. Using a

combination of orthographic transcriptions, initial pronunciations obtained by means of G2P conversion, as well as target (actual or expected) pronunciations, P2P rules can be trained to model the variation in non-standard speaker pronunciations by generating alternative pronunciations.

Acoustic modelling approaches that address non-standard pronunciation modelling are beyond the scope of this literature review.

2.5 Grapheme-based sub-word units

Grapheme-based sub-word units are sub-word units based on the orthography of words rather than their phonemic representation. For example, using the most simplistic form of grapheme-based sub-word units, the word *state* can be represented by the sequence of graphs that makes up its orthography /s t a t e/. These orthographically motivated sub-word units form the basic recognition units of grapheme-based ASR systems.

First proposed in 1993 [6], grapheme-based ASR has shown that, for regular languages, phoneme-based dictionary development might be unnecessary [4, 5]. Grapheme-based ASR has been applied to many different languages with varying degrees of success. With the ever increasing availability of training data, grapheme-based ASR yields competitive performance and has in some instances been shown to outperform phoneme-based systems [12].

Some of the earliest work on grapheme-based speech recognition proposes using polygraphs as sub-word units [6]. Polygraphs, similar to polyphones (only polyphones are phoneme-based units), are letter-based units constructed from the orthographic form of words with arbitrary length left and right contexts. Applied to English, they were found to yield higher WERs than both polyphones and triphones, but succeeded in outperforming context-independent phoneme-based recognition.

Context-dependent acoustic modelling using a decision tree based on graphemic-acoustic sub-word units together with either phonetic or automatically generated questions is explored in [5]. Decision tree state tying is applied directly to the orthographic word form. Question sets are obtained by either (a) relying on pre-existing hand-crafted phonetic questions, assigning graphemes to phonetic questions if the graphemes and phonemes are mapped, or (b) by learning them directly from the acoustic training data. This approach was evaluated on Dutch, German, Italian and English, yielding competitive performance in all languages except English.

In [4], context-dependent grapheme-based recognition is compared against phoneme-based recognition in three different languages: English, German and Spanish. Context is modelled through decision tree based clustering. It is shown that for regular languages (German and Spanish) grapheme-based recognition is comparable to phoneme-based recognition.

Some approaches, such as that of [35, 36], model both grapheme- and phoneme-based sub-word units simultaneously. Leveraging the additional information provided by grapheme-based sub-word units result in ASR systems able to outperform their singleton counterparts. Using a hybrid HMM/artificial neural network framework, grapheme- and phoneme-based systems are developed alongside each other and then used simultaneously or independently during recognition. These hybrid systems outperformed purely phoneme-based systems in two different English recognition tasks: isolated word recognition [36] and number recognition [35].

The most recent work on grapheme-based ASR includes a KL-HMM probabilistic lexical modelling approach [10]. Grapheme-based KL-HMM-based ASR models the probabilistic relationship between context-dependent graphemes and acoustic states. Evaluated on English – a highly irregular language – and without using any phonetic knowledge, the KL-HMM approach reaches close to state-of-the art performance.

2.6 Language regularity

The regularity of a language indicates how consistently the orthography of words from the language describes their pronunciation. In other words, the term describes the strength of the relationship between graphemes and phonemes (for example, a fully regular language will have a one-to-one relationship between graphemes and phonemes.)

As mentioned in Section 1.1, the regularity of a language can be quantified using G2P consistency measures [8]. According to this measure, languages vary considerably, from highly irregular languages such as English, to highly regular languages such as Spanish or Vietnamese, with Afrikaans – the language used as a case study for this paper – being of medium regularity [37].

2.7 Conclusion

This literature review introduced the key terms and concepts necessary to contextualise the rest of this study. After defining the problem of speech recognition, we described

ASR systems as HMM-based recognisers consisting of three probabilistic models, namely (1) an acoustic model, (2) a language model, and (3) a pronunciation model. We briefly mentioned the various G2P conversion techniques and described JSMs and D&R – the G2P techniques used in this study – followed by a quick introduction to pronunciation modelling of non-standard words. After an overview of grapheme-based ASR and grapheme-based sub-word units, we introduced the concept of language regularity.

Chapter 3

Data preparation

To develop ASR systems, large datasets of transcribed audio are typically required. These datasets are used to train the acoustic models, and their transcriptions can also be used in language modelling and pronunciation dictionary development. In the next chapter, we intend to compare different (grapheme- and phoneme-based) ASR systems and establish the effect that different word categories have on ASR performance. This requires various resources, such as a trustworthy pronunciation dictionary and categorised word-lists. This chapter is dedicated to the data preparation necessary to obtain these resources, which are required to facilitate ASR system development and evaluation.

3.1 Introduction

During the data preparation phase, we set out to achieve the following outcomes: (1) ensure a well-balanced dataset, (2) produce a gold-standard pronunciation dictionary, and (3) categorise all the words in our dataset:

- A *well-balanced dataset* is selected from an existing transcribed audio corpus. A development set is held out and the dataset is prepared for four-fold cross-validation.
- To ensure a fair comparison between grapheme- and phoneme-based systems, a considerable amount of time is spent on pronunciation dictionary development and verification. We employ various strategies and implement language-specific methods to lessen the total effort of developing and verifying an Afrikaans pronunciation dictionary.
- In order to determine the effect that certain *word categories* have on both grapheme- and phoneme-based ASR, each word contained in the dataset is assigned to a category.

The rest of this chapter describes the data selection process in Section 3.2, pronunciation dictionary development in Section 3.3 and categorisation of word lists in Section 3.4. Section 3.5 concludes the chapter with a summary of our findings as well as a list of resources.

3.2 Data selection: Afrikaans corpus

Afrikaans was selected as the experimental language both because of its G2P regularity (fairly regular without being fully regular) and the author's inherent familiarity with the language, making it easier to identify problems, trace errors and evaluate results while comparing grapheme- and phoneme-based ASR systems.

The dataset used is a subset of the National Centre for Human Language Technologies corpus [38] and has a total length of approximately $64\frac{1}{2}$ hours, consisting of 75 150 utterances from 167 speakers with a male-to-female ratio of 48.5/51.5. Every utterance in this dataset passed basic quality control checks, namely clipping detection, volume detection and speech cutting detection [39].

We ensure that every speaker contributes exactly 450 utterances, ignoring speakers who contribute fewer. From this speaker-balanced dataset, a development set of approximately two hours and 45 minutes was held out. The remaining utterances were split into four folds with four mutually exclusive test sets. Each fold's train set is roughly 46 hours long and contains 54 000 utterances from 120 different gender-balanced speakers. Table 3.1 shows the number of utterances, hours of audio data and number of speakers in each fold's train and test set.

TABLE 3.1: *Data selection: Number of utterances (utt), hours (hr) of audio data and number of speakers (spkr) in train (trn) and test (tst) sets across folds (F).*

F	utt trn	hr trn	spkr trn	utt tst	hr tst	spkr tst
1	54000	46:18:56	120	18000	15:25:09	40
2	54000	46:51:34	120	18000	14:52:31	40
3	54000	45:51:57	120	18000	15:52:08	40
4	54000	46:09:50	120	18000	15:34:15	40

All four train sets were then individually subdivided into 46 totally random, non-sequential incremental segments. In effect, each segment contained approximately one hour more data than the previous one. We then selected segments 5, 10, 20 and 40 for training. The division of these train and test sets for each fold is shown in Table 3.2.

TABLE 3.2: *Training segments: Hours of audio data and number of utterances (utt) per segment (seg) across folds (F).*

F	seg 5	seg 10	seg 20	seg 40
1	05:05:24	10:05:53	20:07:59	40:14:12
2	05:06:05	10:11:15	20:24:25	40:45:14
3	05:02:28	10:00:23	19:55:38	39:53:24
4	05:02:50	10:03:34	20:05:03	40:07:01
utt	5870	11740	23479	46957

3.3 Pronunciation dictionaries

We developed three different pronunciation dictionaries: (1) a hand-crafted phonemic dictionary, (2) a state-of-the-art G2P-based dictionary, and (3) a minimal effort grapheme-based dictionary.

Firstly, we developed and manually verified a phonemic pronunciation dictionary, which served as a gold standard. To lessen the total effort of classifying, predicting pronunciations for and verifying all the words in our dataset, we employed various strategies (detailed in Section 3.3.1.1). It should be noted that this dictionary contained pronunciation variants where appropriate. Also, since Afrikaans contains many compound words, we focused our effort on identifying known compounds from existing dictionaries, using both a form of longest string matching (LSM) and automated morphological decomposition to achieve this aim. All automated methods used to produce pronunciations were manually verified, which allowed us to report on the success rates of each of these methods.

Secondly, to illustrate the level of performance achievable by a rule-based method, the best possible rule set available to date – rules extracted from the *rcrl_apd* pronunciation dictionary [40] – was used to create an automated (state-of-the-art G2P) pronunciation dictionary.

Finally, a minimal effort grapheme-based dictionary was developed by simply splitting the orthographical form of words into space-separated single letters.

Each of these pronunciation dictionaries will be used to train a separate ASR system. For a fair comparison between systems, it is important to ensure that these dictionaries (specifically the gold-standard dictionary), are not only as accurate as possible but also that each word contained in them is correctly categorised. The following sections detail the development of these dictionaries.

3.3.1 Phoneme-based dictionary

The most comprehensive Afrikaans dictionary currently available is the *Resources for Closely Related Languages Afrikaans Pronunciation Dictionary (rcrl_apd)* [40]. This dictionary, however, does not include all of the 9 375 unique words present in the dataset we are modelling. We therefore have to obtain and verify pronunciations for each of the additional words.

3.3.1.1 Pronunciation verification

The total effort in verifying all the phonetic sub-word units is lessened by using methods such as:

- known word extraction: accepting known pronunciations from existing dictionaries;
- decompounding unknown words and matching these to known components in existing dictionaries;
- short word extraction: analysing short words – which are often non-standard words such as abbreviations or acronyms – separately; and
- the classification of word types to be preprocessed by appropriate G2P methods.

Initially, all known words from existing dictionaries were extracted: this comprised nearly two thirds of the dictionary. Remaining words were then checked against known word lists and classified as either valid Afrikaans words, valid English words or unknown words.

All valid English words were then removed, their pronunciations predicted with English G2P rules and manually verified. The remaining words were then processed concurrently by the two different decompounding methods described in Section 3.3.2.

Short word extraction was performed on the remaining words by extracting all words with a length of one to four characters. The vast majority of these words fell into the category of spelled out Afrikaans words. Spelled out words are words with pronunciations that are spelled out as if they consist of single letters in the written alphabet. For example, the word *RSA* is phonemically spelled out in Afrikaans as /{ r E s A:/. High numbers of partials, abbreviations and acronyms were also present. Partials constitute incomplete words caused by fragmented utterances encountered during the transcription of the audio corpus. For example, the partial word *kontan-* can be the partial form of

the word *kontant*, with the missing part of the word indicated with a hyphen. Unfortunately, such fragments are not specifically marked in the corpus used, and the word list would then (in this example) include the partial word *kontan* as an entry.

Words were first categorised and pronunciations were generated with appropriate G2P methods, after which all these words were reviewed manually. Pronunciations for single character words were created manually. For the remaining 1 351 words, pronunciations were predicted and manually verified. All manual verification was performed by two different verifiers. Results for each step in this process are given in Table 3.3.

TABLE 3.3: *Per step of the dictionary development process: the number of words correctly identified and the number of valid pronunciations prior to manual correction.*

Process	Words identified	Valid categories	Valid pron
extract known Afr words	5 925	5 925	5 925
G2P valid Eng words	225	189	163
ID compounds (Morfessor)	1 419	1 313	1 265
extract short words	253	196	-
ID compounds (LSM)	203	179	151
review remaining	1 351	-	-

3.3.2 Identifying known constituents in compounds

As discussed in Section 3.3, we experimented with two different approaches to decompounding. Since Afrikaans contains many compounds, many words in a word list would be flagged as unknown when measured against existing dictionaries, while the constituents are actually known and pronounced in an identical manner. Note that the primary purpose was to lessen the total effort in creating a pronunciation dictionary: not to find linguistic compounds as such, but only to find known constituents from existing dictionaries (that is, where pronunciations are known.)

In the remainder of this section we describe the two approaches used (variants of Morfessor-based decompounding and LSM), the post-processing that is required (which is similar for both approaches) and the results achieved.

3.3.2.1 Morfessor

Morphological decomposition was performed using a modified version of Morfessor 1.0 [41], a popular language-independent tool for performing unsupervised morphological decomposition. We changed the tool to use only existing words as 'morphemes' and not to create smaller linguistic components, in effect changing it into a decompounding tool. All other settings were left at their default values.

Given as input a combination of unique words from an existing dictionary and all words with unknown pronunciations, Morfessor then suggests segmentations for all words, based on identified segments that exist as individual words in an existing dictionary. Words that can be segmented are flagged as candidate compounds, new pronunciations are generated based on the pronunciations of the individual words and prepared for review.

3.3.2.2 LSM

A restricted version of the LSM algorithm similar to that of [42] was used. In our LSM algorithm the longest left-hand match is performed at the same time as the longest right-hand match, possibly causing overlap and missing some compounds. A limited valence morpheme list is used containing only two valence morphemes, namely *s* and *en*. Using a lexicon of known words as a reference, the largest left- and right-hand matching strings of each candidate compound is determined. Words are then flagged as possible compounds if: (a) after subtraction of the left and right match, there is no remainder and the length of the compound is equal to the combined length of the largest left and right match, or (b) the remainder of the compound is either a valid word from the lexicon, or (c) the remainder is a valid valence morpheme from the list of morphemes.

3.3.2.3 Post-processing

After each decompounding method, the pronunciations of compound constituents were extracted from existing dictionaries, residual consonant doubling caused by constituent concatenation was removed, and finally, flagged compounds and their accompanying phoneme strings were manually verified.

3.3.2.4 Results

After verification, we found 1 492 compounds in the data set (containing 3 225 unique words) of which 1 416 had correct pronunciations. A breakdown of our results is shown in Table 3.4. Morfessor decomposition was applied first, then LSM-based decomposition. Note that LSM-based decomposition was only performed on words that Morfessor was not able to decompound, resulting in 179 additional compounds. Since we are not interested in finding linguistically accurate compound boundaries, some of the words identified are not actual compounds, yet they still produce correct phoneme strings. Table 3.5 summarises the effect of decomposition on pronunciation. Most pronunciation

errors relate to a few small morphemes ('ver', 'end', 'bes') that were incorrectly predicted as containing the /E/ vowel, rather than the /@/ vowel.

TABLE 3.4: *Breakdown of LSM and Morfessor-based decomposition showing the number of correctly identified and incorrectly identified compounds.*

	Total flagged	Correctly identified	Incorrectly identified
LSM	203	179	24
Morfessor	1419	1313	106

TABLE 3.5: *Effect of decomposition on pronunciations.*

	Pronunciation		
	correct	error	% correct
correctly decomposed	1 416	76	94.6
incorrectly decomposed	130	119	8.5

3.3.3 G2P-based dictionary

Using rules from the *rcrl_apd* pronunciation dictionary [40], we created a G2P-based pronunciation dictionary. We assumed zero knowledge about the words for which we were predicting pronunciations and blindly applied the rules. The G2P-based dictionary is included in this study to compare its effectiveness to that of the hand-crafted phoneme-based and grapheme-based dictionaries.

3.3.4 Grapheme-based dictionary

A minimal effort grapheme-based dictionary was developed by simply splitting the orthographical form of words into space-separated single letters. We again assumed zero knowledge about the words and blindly processed each one in the dataset.

3.3.5 Analysis

To establish a measure of how similar the phoneme-, grapheme- and G2P-based dictionaries are, we compared them in terms of phoneme accuracy. Phoneme accuracy is measured as:

$$\frac{H - I}{H + S + D} \times 100 \tag{3.1}$$

where S, D, I and H denote the number of substitutions, deletions, insertions and correct phonemes ('hits'), respectively. Later in the study we use phoneme error rate (PER) as defined in Chapter 4.

Using the gold-standard dictionary as a reference, the phoneme accuracy of the G2P-based dictionary measured 96.31% with 85.33% of words being identical. This indicates that there is strong similarity between the two dictionaries. Note that for words in the phoneme-based dictionary with more than one pronunciation, only the first pronunciation in a sorted list was chosen. (Since the list is sorted, the shortest pronunciation variants are always selected because they appear at the top of the list.) No further analysis was performed, as we only required an indication of the extent to which the dictionaries differed.

To compare the grapheme-based dictionary to the phoneme-based dictionary, the graphemic sub-word units – the space-separated single letters – are first converted to their 'default' corresponding phonemes. (A 'default' phoneme is the phoneme that a specific grapheme will map to, given zero context.) A relative phoneme accuracy of 63.27% was obtained by comparing the grapheme dictionary to the gold-standard dictionary. Our findings are presented in Table 3.6.

TABLE 3.6: *Relative phoneme accuracy and percentage of correct words for the G2P dictionary and grapheme dictionary using the gold-standard dictionary as reference.*

Dictionary	Total words	Total phonemes	% Words correct	% Phoneme accuracy
phone	9 374	78 621	-	-
graph	9 374	86 883	6.37	63.27
g2p	9 374	78 063	85.33	96.31

3.4 Word categorisation

In order to be able to test our hypothesis – that the discrepancy in grapheme-based performance is caused by words from more irregular categories – in the next chapter, every word in the dataset is carefully categorised.

For our initial experimentation, categories included (1) abbreviations, (2) acronyms, (3) foreign words, (4) generic Afrikaans words, (5) partial words, (6) proper names, (7) concatenated words, (8) spelling errors, (9) spelled out words, (10) spelled out single characters and (11) unknown words. Words that belonged to more than one category (because of pronunciation variants or context) were classified as multi-category words. Pronunciation variation caused all but one abbreviation to be classified as multi-category words. (For example, *kzn* - the abbreviated form of the proper name *KwaZulu-Natal* - can be pronounced in its entirety as /k w a z u l u n a t a l/ or its individual constituent letters can be spelled out as /k A: z E d E n/.) Word category type frequencies are listed in Table 3.7.

TABLE 3.7: *Frequency of word categories.*

Category	Types
abbreviations	1
acronyms	22
concatenated words	114
foreign words	204
generic Afr words	8185
multi-category words	137
partial words	135
proper names	223
spelling errors	112
spelled out char	20
spelled words	115
unknown words	37

During system optimisation (see Section 4.5), word categories were revised to better match the types of variation observed. Specifically, erroneous words were combined into a single category containing: (a) partial words, (b) concatenations and (c) spelling errors. Spelled out single characters were grouped together with spelled out words. Categories containing a limited number of words (abbreviations and acronyms), together with unknown words and all unclassifiable words, were classified as other.

3.5 Conclusion

In this chapter we described the data preparation necessary to facilitate our experimentation. A dataset was selected from an existing transcribed audio corpus and prepared for four-fold cross-validation. We developed three pronunciation dictionaries, namely (1) a hand-crafted phoneme-based dictionary that served as a gold standard, (2) a state-of-the-art G2P based dictionary, and (3) a minimal effort grapheme-based dictionary.

We also presented language-specific techniques that alleviated the amount of effort required to develop and verify a hand-crafted phoneme-based dictionary. We found that by identifying known constituents in compound words we could accurately predict the pronunciations of those compound words. Using a version of Morfessor modified to split words into constituent parts that exist as entries in a dictionary, we combined these constituent pronunciations to form new pronunciations.

We compared pronunciation dictionaries in terms of phoneme accuracy and found that phoneme- and G2P-based dictionary accuracies were very similar. Finally, the word categories used in this study were introduced. All relevant resources generated during the data preparation phase are listed in Table 3.8.

Having achieved our defined outcomes, namely (a) a gold-standard pronunciation dictionary, (b) categorised word lists, and (c) a well-balanced dataset, we set out to compare grapheme-, phoneme- and G2P-based ASR systems in the next chapter.

TABLE 3.8: *Resources generated during the data preparation phase.*

Resource name	Description
phone_dict_v1	hand-crafted, gold-standard phoneme-based dictionary
graph_dict	space separated single letter grapheme-based dictionary
G2P_dict	*rcrl_apd* rule generated G2P-based dictionary
word_cats_v1	preliminary categorised word list

Chapter 4

Comparing grapheme-based and phoneme-based systems

Conceptually the only difference between grapheme- and phoneme-based ASR systems is their respective sub-word units. In this chapter we develop a grapheme-based ASR system alongside a phoneme-based ASR system, compare their performance and perform an analysis of the word categories responsible for recognition errors. After initial experimentation, we optimise system parameters, review word categories and improve the phoneme-based dictionary. The resulting grapheme- and phoneme-based systems are used to establish baseline results that we aim to improve on in Chapter 5. Our findings point to an approach to address grapheme-based inaccuracies at the lexical level.

4.1 Introduction

In order to test our hypothesis – that grapheme-based performance degradation is primarily caused by irregular word categories – we develop a grapheme-based ASR system alongside a phoneme-based system using the same standardised approach in both, in the one case using tied-state triphones and in the other, tied-state trigrams. The only variable between the systems is their respective pronunciation dictionaries, which allows for a fairly direct comparison of strengths and weaknesses. We also include a G2P-based ASR system built using an automatically generated rule-based dictionary (*G2P_dict*). For each system the word categories of recognition errors are classified and compared.

The remainder of this chapter is structured as follows: Section 4.2 details the experimental design we follow to compare grapheme-based and phoneme-based performance.

We establish comparative system accuracy and present results in Section 4.3. Category-specific recognition errors are analysed in Section 4.4. System optimisation is performed and baseline results are established in Section 4.5. Section 4.6 discusses the difference between optimised and initial results. Finally, this chapter concludes with a summary of our main observations in Section 4.7.

4.1.1 Evaluation metrics

For this study, the following evaluation metrics are defined:

$$WER = \frac{S + D + I}{H + S + D} \times 100 \tag{4.1}$$

where S, D, I and H denote the number of substitutions, deletions, insertions and correct words ('word hits'), respectively.

WER is a standard metric in ASR literature and in this work it is used to report on most results where different systems are being compared. (Another measure often used when evaluating pronunciation dictionaries, PER, is calculated exactly like WER, except that deletions, substitutions, insertions and hits correspond to phonemes and not words.)

Since WER does not have a constant upper bound, we find matching error rate (MER) [43] to be a more intuitive metric when performing category-based analysis:

$$MER = \frac{S + D + I}{H + S + D + I} \times 100 \tag{4.2}$$

WER can then be seen as the ratio between the number of errors and number of words predicted in the reference transcription, and MER as the probability that a reference word is incorrectly predicted.

During the initial word category analysis in Section 4.4 we report on the category-specific false recognition rate (FRR):

$$FRR = \frac{S}{N} \times 100 \tag{4.3}$$

where N denotes the total number of words in a category, and S denotes substitutions, as above.

Category-specific FRR is the percentage of how many times words from a specific category are misrecognised as other words (either from the same category or another) out of the total number of words from that category in the test set. This measure only

shows how easily words are substituted, that is, how many substitution errors they are responsible for (it does not indicate the likelihood of an insertion, or deletion).

(Note that both WER and MER can either be measured for all words in a test set, or for a specific category of words. FRR is always category-specific.)

4.2 Experimental design

We develop comparable grapheme-based and phoneme-based ASR systems for different training data sizes ranging from five to 40 hours. The grapheme-based system is developed using *graph_dict* and the phoneme-based system using *phone_dict_v1*. We compare these using independent test sets obtained via four-fold cross-validation (see Section 3.2) and report on mean WER across folds. A G2P-based system developed using *G2P_dict* is included as a reference to show the performance achievable by state-of-the-art G2P-based pronunciation dictionaries.

All test sets are recognised using the same flat language model containing all the words in the entire data set (including the entire test set vocabulary). While better recognition accuracy can be obtained using a statistical language model, we specifically want to evaluate the effect of the acoustic models without recognition being guided by a language model, or constrained by out-of-vocabulary words. This means that the systems are evaluated and compared in terms of WER with the only difference between systems being their pronunciation dictionaries. For the later category-based analysis in Chapter 5, it is particularly important that categories are not influenced by the language model used, as some categories occur more rarely than others. (The effect of the language model is reconsidered later in the study, in Section 5.3.4.)

Substitution errors are then classified according to word category and compared across systems. Substitutions can indicate that either: (a) a word's pronunciation cannot be modelled properly and the system just selects the next most probable word, or (b) a word's pronunciation is easily confusable with the pronunciation of another word.

After the initial word category analysis we optimise system parameters, review word categories and improve the phoneme-based dictionary. After system optimisation, we establish baseline results. (Note that sections 4.3 and 4.4 form part of the initial word category analysis and from Section 4.5 onwards, all results are reported using optimised parameters, reviewed word categories and the improved phoneme-based dictionary.)

To evaluate the effect of the dictionaries, we developed all ASR systems using the same relatively standard approach. We used the hidden Markov model toolkit (HTK) [44]

and developed context-dependent tied-state acoustic models. Feature extraction on the speech audio data produced 13 MFCCs with their first- and second-order derivatives as 39 dimensional feature vectors. The MFCC window size was set at 25 ms with a frame rate of 10 ms. Cepstral mean normalisation was applied at speaker level. With regard to modelling structure, each triphone or trigraph had three emitting states with eight Gaussian mixtures per state and a diagonal covariance matrix. Where parameters (beam width and insertion penalty) were optimised, the development set was used.

4.3 Comparative accuracy

Figure 4.1 shows the effect of different dictionaries on WER at four different training sizes of five, 10, 20 and 40 hours. At all data points evaluated, the hand-crafted phoneme-based dictionary outperformed the other approaches, with the G2P-based system also outperforming the grapheme-based system. At the largest data set size (40 hours) the grapheme-based system had a WER of 41.13%, the G2P-based system a WER of 39.82% and the phoneme-based system a WER of 38.03%. As is evident in the convergence of WER between the phoneme-based and grapheme-based ASR systems, the more training data that is available, the less grapheme-based performance degradation becomes.

Figure 4.2 shows the absolute difference in WER between (1) grapheme-based and G2P-based ASR, (2) grapheme-based and phoneme-based ASR and (3) G2P-based and phoneme-based ASR systems. The highest absolute inter-system difference measured 8.25% between grapheme-based and phoneme-based ASR at five hours of training data.

As training hours increase, G2P-based ASR consistently performs approximately 1.93% absolutely worse than phoneme-based ASR. This indicates that even with an increase in training size, G2P-based ASR is unlikely to outperform phoneme-based ASR. The absolute inter-system difference at 40 hours measured 1.31% between G2P-based ASR and grapheme-based ASR.

4.4 Word category analysis

With the difference in WER being the most pronounced at five hours, we analysed FRR (substitution errors) at this data point according to word category. As mentioned in Section 3.4, the abbreviation category contained only one word, namely *mej*, and since it did not occur in every fold's test set, the abbreviation category was ignored during error analysis, leaving a total of 11 categories. Table 4.1 gives a detailed view of category-specific FRR, for *word_cats_v1*, observed at five hours of training data.

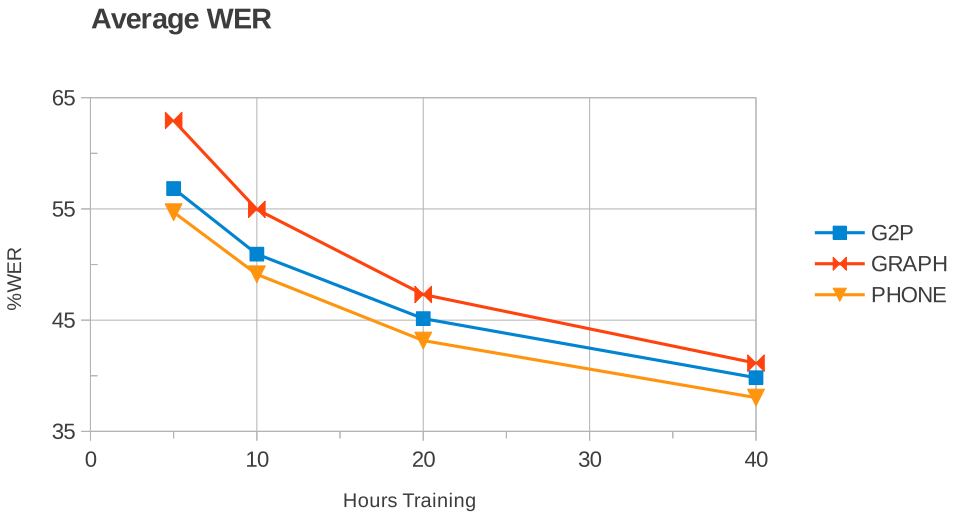

FIGURE 4.1: *Average WER of grapheme-based, G2P-based and phoneme-based ASR for training sizes of five, 10, 20 and 40 hours across four folds.*

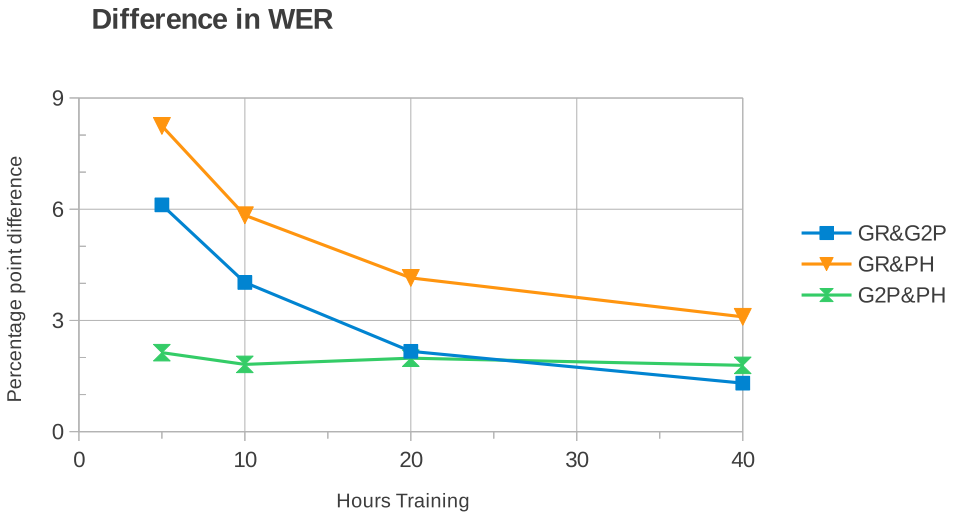

FIGURE 4.2: *Average absolute difference in WER between grapheme-based and G2P-based ASR, grapheme-based and phoneme-based ASR, and G2P-based and phoneme-based ASR for training sizes of five, 10, 20 and 40 hours across four folds.*

Each cell is coloured green, orange or red to indicate whether the relevant system performed best, second-best or worst, respectively.

Not surprisingly, grapheme-based ASR performed worse than phoneme-based ASR in 10 of the 11 categories. It did however outperform G2P-based ASR in five categories, namely (1) spelled out words, (2) proper names, (3) spelling errors, (4) partial words and (5) multi-category words. The high FRR of spelled out single characters can be attributed to the language model used: with a flat language model the insertion penalty

(the cost of adding an extra word during decoding) must be very high in order to produce sensible results. This causes short words to be misrecognised very frequently.

TABLE 4.1: *Category-specific FRR observed at five hours of training data.*

	% FRR		
Category	g-based	g2p	gold-dict
spelled out char	73.73	68.31	63.65
multi-category	38.53	40.54	29.36
acronyms	32.03	28.91	26.95
unknown words	28.65	25.15	28.65
spelled out words	27.96	30.53	15.27
foreign words	16.04	14.92	13.84
proper names	10.44	11.00	9.48
spelling errors	10.40	11.42	9.68
concatenation	7.48	5.79	5.67
partial words	6.62	7.31	6.13
generic Afr words	2.81	2.49	2.68

Similarly, with the difference in WER being least at 40 hours, we again split errors based on word categories. Our findings are presented in Table 4.2, which gives a detailed view of word category-specific FRR, for *word_cats_v1*, observed at 40 hours of training data. Grapheme-based ASR now outperforms G2P-based ASR in four out of the 11 categories, tying for an additional two categories. With increased training data, grapheme-based ASR managed to outperform phoneme-based ASR in five of the 11 categories.

Interestingly, one of the categories includes generic Afrikaans words: the largest category of words in the test set. This might be attributed to noise-free pronunciation models or increased language regularity, but this also requires further investigation. The biggest disparity in performance occurs in the spelled out words category where grapheme-based ASR misrecognises twice as many words, and G2P-based nearly three times as many words as phoneme-based ASR.

4.5 System optimization

After initial system comparison and word category analysis, we continued to: (a) improve the phoneme-based dictionary, (b) review and verify word categories, and (c) optimise system parameters. While it would have been ideal to refine the dictionary first, and then conduct all experiments, external review of the dictionary took a substantial amount of time. We therefore conducted all experiments prior to this section using *v1* resources, but from this point onwards, used *v2* exclusively.

TABLE 4.2: *Category-specific FRR observed at 40 hours of training data.*

	% FRR		
Category	g-based	g2p	gold-dict
spelled out char	62.65	66.90	63.89
multi-category	37.57	35.87	27.52
acronyms	31.50	20.47	25.98
unknown words	25.07	25.07	25.66
spelled out words	23.24	28.47	10.89
foreign words	13.61	12.81	10.00
proper names	10.26	11.83	9.65
spelling errors	10.37	11.38	9.22
concatenation	5.24	5.12	6.33
partial words	6.20	6.20	8.27
generic Afr words	1.85	1.76	2.15

From *v1* to *v2*, the phoneme-based dictionary was improved. All pronunciations were manually reviewed by an external phonetician, and corrected as necessary. (An additional 146 pronunciation variants were added where applicable.) A small improvement (less than 1% absolute reduction in WER) is obtained when comparing the newly verified pronunciation dictionary with the dictionary used in the initial experiments.

Table 4.3 shows the number of pronunciations, pronunciation variants and PER of the improved phoneme-based dictionary (henceforth referred to as *phone_dict_v2*), *phone_dict_v1*, *G2P_dict* and *graph_dict*. PER is obtained by scoring all dictionaries against the newly improved *phone_dict_v2*. (The grapheme-based dictionary was again converted to default phonemes prior to comparison – see Section 3.3.5 for details.)

Secondly, word categories were manually reviewed by an external linguist, and corrected as necessary. The reviewed categorised word list (*word_cats_v2*) is compared against *word_cats_v1* in Table 4.4. After reviewing word categories, the categories themselves were revised by grouping together some less prominent categories and creating a new combined category for erroneous words. The revised categories include (1) generic within language words, (2) proper names, a combined category for single and multiple character (3) spelled out words, (4) foreign words and a combined category for (5) erroneous words. The erroneous words category contains (a) partial words, (b) concatenations and (c) spelling errors. Words that belong to more than one category are classified as (6) multiple category words and any remaining words (which include a small number of unknown words, abbreviations and acronyms) are classified as (7) other.

Finally, to ensure a fair comparison between grapheme-, phoneme- and G2P-based ASR, parameter optimisation was performed. Using the development set, for each system, a

search was performed to determine the optimal insertion penalty at a given amount of training data, five and 40 hours respectively.

During the initial experimentation, only matching results, that is, only those results that are present in all three different ASR systems, were used to calculate WER. From here onwards, failed recognitions – utterances from the test set unable to pass decoding successfully – are regarded as errors when calculating WER, both during system optimisation and all further experimentation. This translates to slightly higher, but more accurate WERs.

TABLE 4.3: *Number of pronunciations, number of pronunciation variants, PER and percentage of words correctly recognised for all dictionaries, using phone_dict_v2 as reference. All dictionaries contain 9 305 words.*

Dictionary	pronunciations	variants	% PER	% words correctly recognised
phone_dict_v1	9 795	490	0.4	98.2
phone_dict_v2	9 941	636	0	100
G2P_dict	9 305	0	5.1	84.9
graph_dict	9 305	0	56.3	1.1

TABLE 4.4: *Comparison of word categories in word_cats_v1 and word_cats_v2.*

Categories	word_cats_v1	word_cats_v2
abbreviations	1	3
acronyms	22	13
concatenated words	114	152
foreign words	204	208
generic Afr words	8185	7578
multi-category words	137	253
partial words	135	187
proper name	223	601
spelled out char	20	21
spelled out words	115	143
spelling errors	112	125
unknown	37	42

Figure 4.3 shows the baseline results for grapheme-, phoneme- and G2P-based ASR at four different training data sizes of five, 10, 20 and 40 hours. Comparable to observations in the initial experiments, the more training data is available, the less the performance degradation that is observed with grapheme-based ASR. The phoneme-based system outperforms both other systems at all training set sizes. Initially performing second best, G2P-based ASR is overtaken by grapheme-based ASR just after the 20-hour mark. Using a paired *t*-test, all performance differences are statistically significant at the $p=0.01$ level, except between grapheme- and G2P-based ASR at 10 and 20 hours.

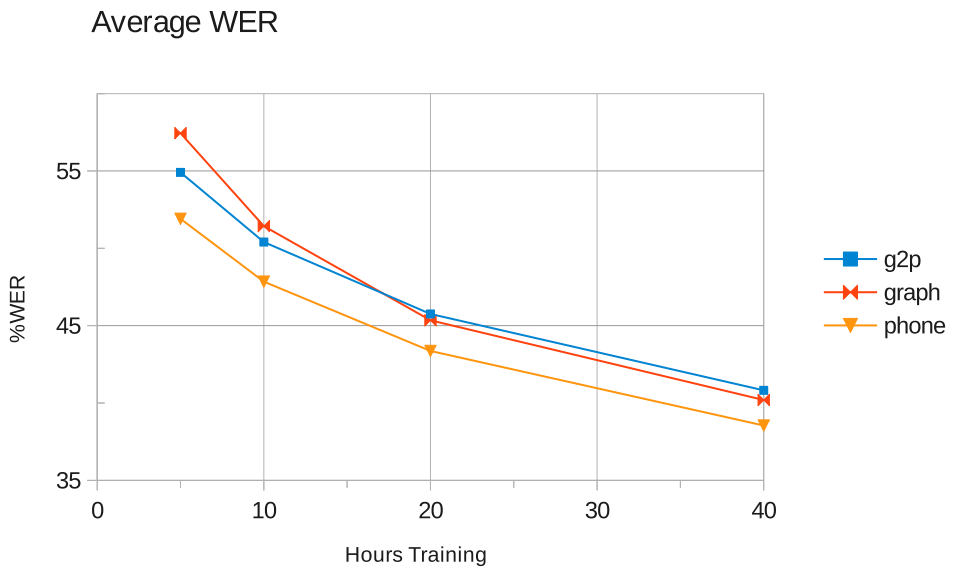

FIGURE 4.3: *Average WER of grapheme-based, G2P-based and phoneme-based ASR for training sizes of five, 10, 20 and 40 hours across four folds.*

Figure 4.4 shows the absolute difference in WER between (1) grapheme-based and G2P-based ASR, (2) grapheme-based and phoneme-based ASR and (3) G2P-based and phoneme-based ASR. The difference in WER between G2P- and phoneme-based ASR remains relatively stable, and a slight decline indicates that G2P-based ASR is unlikely to reach the same level of performance as that of a system trained using a hand-crafted dictionary.

FIGURE 4.4: *Average absolute difference of WER between grapheme-based and G2P-based ASR, grapheme-based and phoneme-based ASR, and G2P-based and phoneme-based ASR for training sizes of five, 10, 20 and 40 hours across four folds.*

Table 4.5 shows the WER and standard error for grapheme-, phoneme- and G2P-based ASR for increasing hours of training data. We calculate standard error (the standard deviation observed across the four cross-validation folds, divided by the square root of the number of folds) as an indication of statistical significance. This is the same data used to draw Figure 4.3.

TABLE 4.5: *WER and standard error (std err) of grapheme-based ASR (graph), phoneme-based ASR (phone) and G2P-based ASR (g2p) for five, 10, 20 and 40 hours of training data.*

	% WER			std err		
hours	graph	phone	g2p	graph	phone	g2p
5	57.44	51.91	54.90	0.45	0.72	0.62
10	51.43	47.85	50.40	0.41	0.58	0.66
20	45.34	43.36	45.76	0.47	0.44	0.48
40	40.19	38.40	40.82	0.37	0.46	0.35

4.6 Discussion: *v1* vs *v2* resources

The prior sections presented two different sets of results: results presented in sections 4.3 and 4.4 are more detailed but used only the initial resources available at the time; resources (and results) presented in Section 4.5 have been verified more systematically, specifically through manual review of dictionaries and word lists. Since the trends when comparing phoneme- and grapheme-based systems remained similar across the two sets of results, we included the more detailed results from the initial investigation as well.

4.7 Conclusion

In this analysis, grapheme-based systems do not reach the same level of performance as that of a system developed using a hand-crafted phonemic dictionary. This degradation in performance is primarily caused by very specific word categories, namely (1) spelled out words, (2) acronyms, (3) proper names and (4) foreign words. All these categories (except for acronyms) tend to have highly irregular relationships between graphemes and phonemes, confusing both the G2P-based and grapheme-based systems.

These categories are typically easy to identify. Spelled out words and acronyms tend to be short (and generic short words – which are not acronyms or spelled out words – tend to be known), and foreign words can mostly be identified using known word lists in relevant languages. Proper names tend to be more difficult to identify from text (unless capital letters are accurately retained during preprocessing). If capitalisation has

not been retained, text-based named entity recognition can be used to identify proper names [45]. Once identified, these categories tend to be small in comparison with the total number of words to be modelled.

Confirming our previous hypothesis – that grapheme-based performance degradation is primarily caused by irregular word categories – we now set out to determine if grapheme-based performance can be increased specifically, by 'regularising' the spelling of these irregular word categories.

Chapter 5

P2G Transliteration

Based on our findings in the previous chapter – that grapheme-based ASR performance degradation is primarily caused by word categories written in irregular ways – we set out to improve the performance of grapheme-based ASR by transforming the orthography of words from these categories. We train P2G rules able to 'regularise' (transliterate) the spelling of irregular words, and use these rules to adapt the spelling of such words, prior to incorporating them in an ASR system. This process of 're-spelling' irregular words is referred to as 'P2G transliteration'.

Previously identified problematic categories are first transliterated and the effect on ASR system performance is evaluated in isolation (one category at a time), before transliterating multiple categories within a single system. The combined system is then compared against baseline results (established in the previous chapter – see Section 4.5) and results are analysed for each category individually. Finally, the P2G transliteration technique is applied and evaluated in the context of a different language, namely Vietnamese.

5.1 Introduction

When building a new grapheme-based ASR system, our first step is to distinguish between words with regular and irregular pronunciations using word orthography only. (In cases where pronunciations are available, distinguishing between regular and irregular words is trivial; we however consider a more limited approach where all pronunciations are regarded as unknown.) While this is not possible for all irregular words, there are some easily identifiable categories (such as spelled out words, foreign words and proper names) that form the focus of this study. Focusing on word categories (instead of words) allows us to identify a large percentage of irregular words prior to system development.

Identifying all irregular word categories from orthography alone is not necessarily always possible. Still, foreign words can typically be identified using known word lists in various languages, spelled out words typically have a known structure and, if capitalisation has not been maintained, text-based named entity recognition can be used to identify proper names. Once these categories have been identified, our aim is to obtain pronunciations for these smaller sets of words, and use them to transform the original orthography of these words into an 'idealised' form that can be incorporated into a grapheme-based system. This is the heart of our P2G transliteration technique.

To determine when (and when not) to transliterate words, we transliterate each category in isolation and train and evaluate a separate ASR system using a development set. (See Section 3.2 for a description of datasets used.) Individual systems are then compared against the baseline grapheme-based system and categories from those systems that resulted in better overall performance are combined.

After the combined categories have been transliterated, we incorporate them in a single ASR system and perform a category-based analysis to (a) measure the total gain in performance, and (b) to ensure that transliterated categories do not negatively affect other categories. We find that although some categories do suffer a slight performance degradation, the combined gain in accuracy from transliterated categories is more than enough to yield an overall improvement in performance. Once all transliteration decisions have been made, a new set of results is obtained using the test set.

The rest of this chapter is organised as follows: Section 5.2 describes the development of the P2G transliteration technique. The technique is evaluated and analysed in Section 5.3. Applying P2G transliteration to a different language (Vietnamese) is explored in Section 5.4 and Section 5.5 concludes with a summary of our main observations and findings.

5.2 Technique development

We use generic in-language words – that is, words from the 'regular' category – to generate crude (broadly applicable but not accurate in detail) P2G rules. These rules learn the mapping between phonemes and graphemes and require both a phoneme string (pronunciation) and a grapheme string (spelling) during training. (For an example of training data see Table 5.1, with phoneme strings on the left and grapheme strings on the right.) For this study, initial rules are developed using second-order JSMs. These models are not refined on purpose, as we expect that capturing too much spelling detail will cause the P2G model to learn idiosyncratic spellings, resulting in irregular transliterations.

To transliterate an irregular word, we require a phonemic pronunciation for that word. Given the pronunciation of a word as input, the P2G model then predicts a new spelling. For example, transliterating the phoneme string /b @u f O r t/ of the proper name *Beaufort*, yields the transliterated orthographical word form *bouvord*. This transliterated word form is much closer to the observed orthography of regular Afrikaans words. Table 5.2 contains more examples of transliterated words, accompanied by their phoneme strings and original spellings.

TABLE 5.1: *Example P2G model training data from the generic Afrikaans words category.*

phoneme string	word
/a f l E/	aflê
/a f r @ x t @ r s/	afrigters
/a f r @ x t @ N/	afrigting

TABLE 5.2: *Original orthographical word form, phoneme strings, transliterations and word categories of transliteration examples.*

original orthography	phoneme string	transliteration	category
federation	/f E d @ r @i S @ n/	vederysjen	foreign word
burundi	/b u r u n d i/	boeroendie	proper name
SMS	E s E m E s	esemes	spelled out word

5.2.1 Verifying the P2G model

In order to evaluate the appropriateness of the P2G model to the task at hand, we transliterate the entire phonemic pronunciation dictionary and measure the similarity between the transliterated and original spellings. We calculate the similarity using phone-based dynamic programming (PDP) scoring [46]. In its standard from, this process uses dynamic programming to measure the similarity between an aligned and (freely) decoded phone string: the dynamic programming score itself is used as a measure of similarity, and a trained scoring matrix can be used to compensate for frequent substitutions naturally occurring in the data being analysed. In this work, we use the PDP implementation to align two graphemic strings: the original and the transliterated spelling. These two strings are aligned using dynamic programming and a flat scoring matrix.

Alignment produces an alignment score with a value between -1 and 1, which we use as a similarity measure. An alignment score of 1 indicates that the transliterated spelling does not differ at all from the original spelling. A score of -1 indicates that every single letter is different. An ideal transliteration mechanism will produce a large number of '1'

values for words with a regular orthography, and a small proportion of lower values for more irregular spellings.

We anticipate that our restricted P2G models will accurately model most of the expected regular words from the generic Afrikaans word category. This assumption is based on the regular G2P relationship of these words. Conversely, we expect that these models will accurately model words from more irregular categories to a more limited degree, indicating that they are 'regularising' the original orthography.

Figure 5.1 depicts the P2G similarity score for each word category: for each similarity score listed on the x-axis, it displays the percentage of words of each category that achieve more than that score on the y-axis. According to this measure, generic Afrikaans words can be regarded as the most regular category with approximately 95% of words having a similarity score of greater than 0.5 and almost 50% of words with a similarity score of 1. In contrast, the spelled word category is the most irregular, with only approximately 12% of words achieving a score of greater than 0.

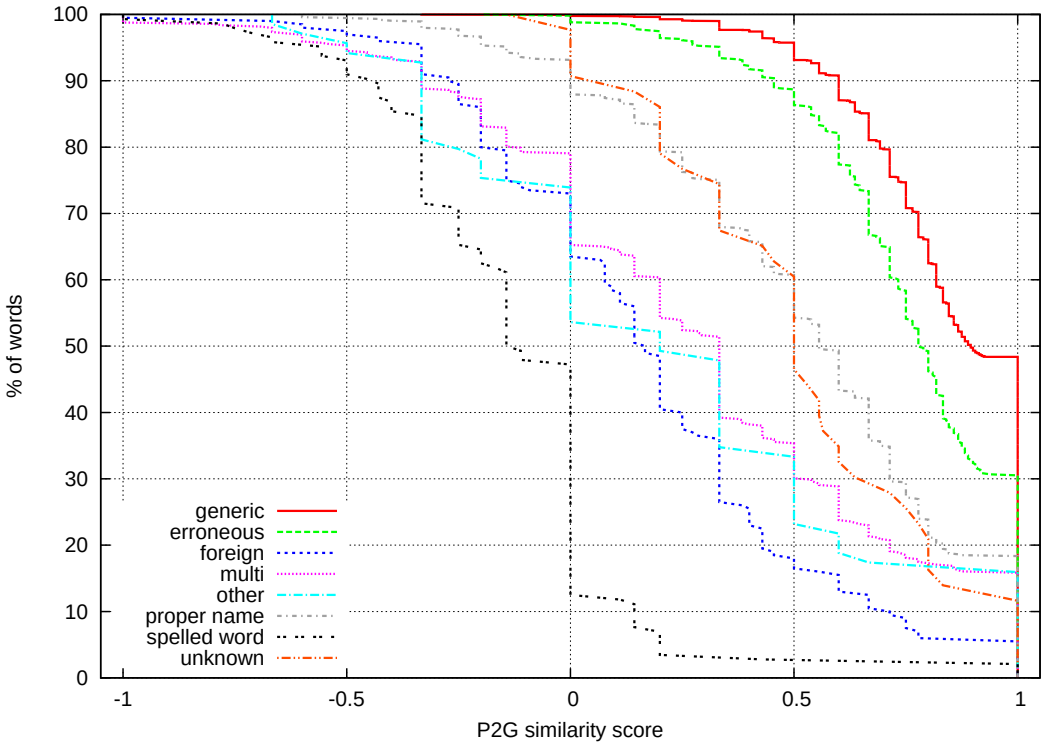

FIGURE 5.1: *Percentage of words from each category that achieve more than a given P2G similarity score.*

5.3 Evaluation and analysis

Experimenting with the development set indicated that transliterating specific categories in isolation resulted in a reduction in WER. These categories included: (1) proper names, (2) spelled out words and (3) foreign words. Proper names and spelled out words tend to have fairly long phoneme strings, but foreign words may be quite short. As additional pronunciation variants for already easily confusable short words are expected to be detrimental, we only transliterate foreign words with more than four characters. (Additional experiments on the development set indicated transliteration of shorter words to be detrimental.)

When transliterating a specific category, two different approaches can be used: the transliterations can either replace the original spelling or be added in addition to the original spelling as a 'spelling variant'. Experiments on the development set indicated that replacing the default grapheme strings of proper names results in a smaller performance increase, compared to adding transliterations as spelling variants. As observed in Figure 5.1, the system benefits by keeping the default grapheme strings of proper names and foreign words as variants. This might be because of non-standard pronunciations present in the acoustic data caused by variation in speaker pronunciation [1]. It was found that for spelled out words, a greater performance gain is achievable when the original grapheme strings are removed and only the transliterations are made available.

All initial experiments were performed on the development set, and a transliteration strategy selected prior to obtaining the first results on the evaluation sets. Only a single strategy was implemented once cross-validation started, as described in the following sections. Table 5.3 shows the results obtained on the development when transliterating different categories using different configurations.

Note, while experimenting on the development set, English words are treated as a separate category from foreign words. Transliteration with a minimum word length restriction of four characters was tested on English words only. Because of the similarity of both categories – from the perspective of Afrikaans they are all foreign – the same strategy identified for English words was applied to foreign words, before combining them into a single foreign word category used during evaluation.

As shown in Table 5.3, transliterating proper names with spelling variants results in a lower total WER. English words perform best when words shorter than five characters are not transliterated. Lower WER rates are observed when adding the original spellings

[1] Speakers unfamiliar with the true language of origin of a word, might pronounce such a word as if it is from the target language (implicitly following the same G2P rules as for generic words in the target language).

of English and foreign words as spelling variants (except when English short words are transliterated).

TABLE 5.3: *Number of insertions (ins), deletions (del), substitutions (sub), percentage of correct words (cor) and WER, as measured when specific word categories (English words only, foreign words with English words removed, proper names and spelled out words) are transliterated in isolation with or without spelling variants and with or without short words, evaluated on the development set.*

category	spelling variants	transcribe short words	ins	del	sub	% cor	% WER
proper names	1	1	1192	148	1909	76.49	36.91
proper names	0	1	1223	141	1933	76.44	37.45
spelled out words	1	1	1240	145	1911	76.64	37.44
spelled out words	0	1	1052	146	1982	75.83	36.12
English words	1	1	1256	148	1949	76.18	38.09
English words	0	1	1263	153	1935	76.28	38.07
English words	1	0	1217	144	1930	76.44	37.38
English words	0	0	1255	154	1915	76.50	37.76
foreign words	1	1	1256	130	1914	76.78	37.49
foreign words	0	1	1282	135	1935	76.49	38.08

5.3.1 Improvement per word category

Figure 5.2 shows the reduction in WER when transliterating word categories in isolation as a bar chart. As a reference, the default grapheme-based and phoneme-based ASR systems are included. Actual WERs, together with the total number of transliterated words in the test set, are shown in Table 5.4. The standard error is included as a measure of statistical significance.

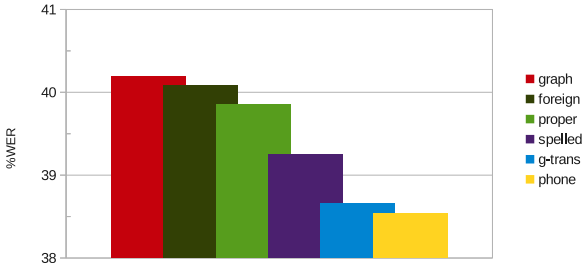

FIGURE 5.2: *Average WER, as measured when specific word categories (foreign words, proper names and spelled out words) are transliterated in isolation, in comparison to the baseline grapheme-based system (graph), the baseline phoneme-based system (phone) and the combined transliterated system (g-trans) evaluated on the test set.*

The total size of the test set is 207 174 tokens. Transliterating foreign words resulted in the smallest reduction in WER of 0.11% absolute. This is understandable because foreign words (including those shorter than five characters) only make up 0.82% of the test set. Transliterating proper names, which comprise 2.72% percent of the test set, resulted in a reduction in WER of 0.34% absolute. The largest reduction in WER of 0.94% absolute is achieved when transliterating spelled out words, which comprise 1.99% of the total test set.

TABLE 5.4: *Number of words transliterated, average WER and standard error (std err), as measured when specific word categories (foreign words, proper names and spelled out words) are transliterated in isolation, in comparison to the baseline grapheme-based system (graph), the baseline phoneme-based system (phone) and the combined transliterated system (g-trans) evaluated on the test set.*

Category	transliterated	% WER	std err
graph	0	40.19	0.37
foreign words	1698	40.08	0.25
proper names	5640	39.86	0.30
spelled out words	4127	39.25	0.39
g-trans	11465	38.66	0.38
phone	0	38.54	0.46

5.3.2 Analysing the effect of word categories (for combined system)

We combine the transliterated categories and incorporate them in a new grapheme-based system. Based on our observations regarding 'spelling variants' and word length, the combined transliterated categories consist of the following:

- proper names with spelling variants,
- foreign words (longer than four characters) with no spelling variants, and
- spelled out words with no spelling variants.

Table 5.5 provides a detailed view of our findings at 40 hours of training data for grapheme- and phoneme-based ASR, as well as a transliterated grapheme-based system consisting of a combination of transliterated proper names, spelled out words and foreign words. Scores are given as a percentage of how many times words from a specific category are misrecognised (substituted, deleted or inserted) out of the total number of words in that category, in other words, category-specific MER. MER percentages are colour-coded, with the worst performing system in red, second best in orange and best in green. When compared to the phoneme-based system, grapheme-based ASR performs worse in four categories, namely (1) proper names, (2) multi-category words, (3) spelled

out words, and (4) foreign words. Combining transliterated categories significantly lowers the MER for each of the individual categories while only slightly increasing the MER of other categories, with the worst case being 5.3% absolute for multi-category words (which is also the worst performing category for transliterated grapheme-based ASR).

It is interesting to note that grapheme-based ASR performs best of all systems in the generic Afrikaans words category. Though this merits further investigation, it is believed this indicates that grapheme-based ASR succeeds in modelling pronunciation variation at the acoustic level. The higher phoneme-based MER might also be caused by confusion introduced by unnecessary pronunciation variants.

TABLE 5.5: *Percentage categories comprise the test set and MER per category for grapheme-based (graph), transliterated grapheme-based (g-trans) and phoneme-based (phone) ASR with 40 hours of training data evaluated on test set.*

		% MER		
Category	% Total	graph	g-trans	phone
generic Afr words	89.47	26.1	27.1	28.0
proper names	2.72	52.6	47.2	44.7
multi-category	2.16	77.2	82.5	74.2
erroneous	2.13	57.7	59.8	64.4
spelled out word	1.99	93.7	79.3	77.5
foreign	0.82	77.0	65.8	62.4
other	0.70	70.9	73.7	76.3

5.3.3 Total gain

Combining all the systems that outperformed the baseline grapheme-based system, namely (1) proper names, (2) spelled out words and (3) foreign words, caused a 1.54% absolute decrease in total WER, exceeding the sum of its parts (1.39%). This results in grapheme-based ASR performance comparable to that of phoneme-based ASR with a difference of 0.12% absolute between systems. (Using a paired t-test, the performance difference is statistically insignificant at the $p=0.01$ and $p=0.05$ level. Figure 5.3 shows the total reduction in WER achieved with category-based P2G transliteration. Baseline grapheme-, phoneme- and G2P-based ASR results are included as a reference.

5.3.4 Effect of language modelling

Based on the results in the prior sections, we were interested in determining whether gains from transliteration disappear when a more realistic language model is used. We therefore compare WER when using the flat language model (used in all prior sections) and a basic SLM during decoding. (It must be noted that the same parameters - those

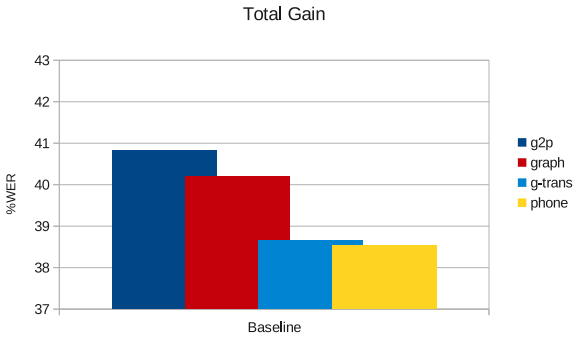

FIGURE 5.3: *Average WER for G2P-based, grapheme-based, transliterated grapheme-based, and phoneme-based ASR at 40 hours of training data for baseline systems using a flat language model.*

optimised for the flat language model - were used in both instances. See Section 4.5.) Separate bigram language models with modified Kneser-Ney discounting [47] were developed for each of the four training data folds using the SRILM toolkit [18].

Initial SLM results are compared to results from the flat language model in Figure 5.4, with the results obtained using the flat language model on the left and the SLM on the right. When using a language model, G2P-based ASR now outperforms the baseline grapheme-based system at 40 hours of training data, with phoneme-based ASR still performing best. It is encouraging to note that the gains obtained from the transliterated system are retained.

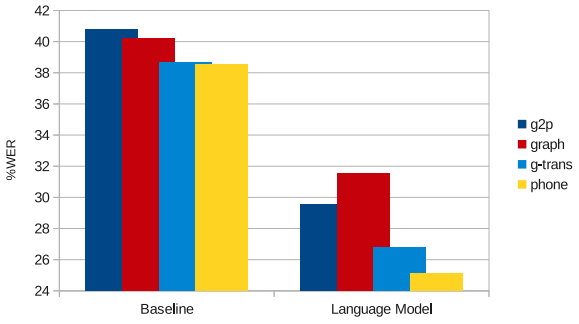

FIGURE 5.4: *Average WER for G2P-based (g2p), grapheme-based (graph), transliterated grapheme-based (g-trans), and phoneme-based (phone) ASR at 40 hours of training data for baseline systems using a flat language model on the left, and a basic SLM on the right.*

5.3.5 Asymptotic performance and model order

As a preliminary study to establish the ideal model order when training P2G rules, we train P2G rules using increasing model order. Using four-fold cross-validation, each

model (trained only on 'regular' generic Afrikaans words) is used to transliterate generic Afrikaans words from a mutually exclusive test set.

Figure 5.5 shows the P2G similarity scores for different model orders, when comparing the original spellings of generic Afrikaans words with the transliterated spellings. Similar to Figure 5.1, with a model order of two, approximately 50% of words have a similarity score of 1, indicating there was no difference between the transliteration and original spelling (for this 50%). When using a model order of four, nearly 95% of words have a similarity score of 1.

Additional experiments are performed to determine how much training data is needed to establish asymptotic performance. Transliterations obtained using limited amounts of training data are compared to 'ideal transliterations' obtained using all available training data. It was found that by using as few as 100 words to train a P2G model, using a model order of two to four, transliterations are 99% similar to 'ideal transliterations', that is, transliterations obtained using the much larger pronunciation dictionary. (Using a model order of one produces 96% similar transliterations.)

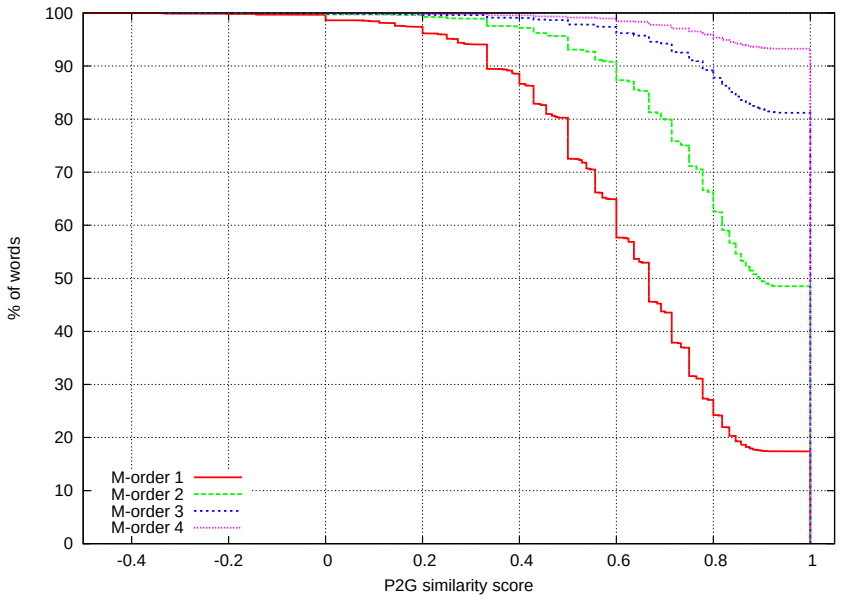

FIGURE 5.5: *Percentage of words achieving more than a given P2G similarity score for increasing model order, trained and evaluated on generic Afrikaans words.*

5.4 Additional language: Vietnamese

As a preliminary study to determine how well the observed transliteration results are transferable to other languages, the category-based P2G transliteration technique is

applied to Vietnamese. This study is conducted as part of the IARPA Babel program[2]. As part of this project both the data sets and ASR systems used are different to those used in the earlier part of the dissertation, providing an ideal environment to test the new technique.

The dataset used is the IARPA Babel Program Vietnamese language collection release IARPA-babel107b-v0.7. It contains both read and conversational speech and consists of a limited language pack (approximately 10 hours of transcribed training data) and a full language pack (approximately 120 hours of transcribed training data). Each language pack contains a manually created lexicon that covers most of the words. For our experimentation we use the limited language pack and its manually created 3117 word lexicon.

5.4.1 Analysis

A grapheme-based ASR system was developed alongside a phoneme-based ASR system using the Kaldi speech recognition toolkit [48]. (Even though HTK – used to develop the Afrikaans acoustic models – and Kaldi are different toolkits, both implement broadly similar HMM-based speech recognition approaches. Kaldi is a recently developed toolkit, incorporating many of the newer ASR techniques, not available in HTK.) The same approach was followed when developing both systems. Each triphone or trigraph had three states and were tied using decision tree clustering. Standard MFCCs (13 MFCCs with first- and second-order derivatives) are used with cepstral mean normalisation. All experiments were performed using a trigram language model, built using SRILM [18] with modified Kneser-Ney [47] smoothing.

We used the IARPA-babel107b-v0.7 development set as our test set. This test contains approximately 10 hours of audio data and has a vocabulary size of 2972 words. Baseline WERs for phoneme- and grapheme-based ASR measured 64.1% and 63.8% respectively.

Based on our previous findings regarding P2G transliteration and irregular word categories, we trained an additional grapheme-based system in which we transliterated three categories, namely (1) spelled out words, (2) spelled out single characters and (3) foreign words. All these words were identified automatically and transliterated simultaneously. Spelled out words had a known structure (underscore separated single letters) and foreign words were identified based on their morphological structure (Vietnamese has a specific consonant-vowel (CV) structure [49]). Foreign words were identified as such if

[2]See http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2013-11/BabelBaseOverview/ for an overview of the project

the graphemic structure did not conform to one of the following CV structures: C_1V; C_1VC_2; VC_2; V [3].

A very low frequency of occurrence was observed for two of the three transliterated categories, namely (1) foreign words and (2) spelled out words. Note that we only transliterated words with pronunciations. Words without pronunciations were considered out of vocabulary and were ignored during category-based analysis. Foreign words occurred 95 times and spelled out words 33 times in total in the test test. (Spelled out single characters occurred 2331 times.)

Table 5.6, Table 5.7 and Table 5.8 show our findings for foreign words, spelled out words and spelled out single characters respectively. These tables show the number of insertions, deletions and substitutions caused by different word categories, as well as the number of words correctly recognised, category-specific WER and category-specific MER. In addition the number of foreign false recognitions, that is, the number of substitutions (out of the total number of substitutions) that can be directly attributed to foreign words, spelled out words or spelled out single characters, are also given.

Initially, the grapheme-based system only recognised four foreign words. After transliteration the grapheme-based system managed to recognise 15 words correctly in total. The phoneme-based system had most correct recognitions with a total of 19. Transliteration caused a 9.5% absolute reduction in category-specific WER, and 14.1% absolute reduction in category-specific MER over the standard grapheme-based system. (Unfortunately, because of the low frequency of occurrence of foreign words and spelled out words, no improvement in the overall performance of grapheme-based ASR was observed.)

TABLE 5.6: *Insertions (ins), deletions (del), substitutions (sub), correct words (cor), foreign false recognitions (ffr), WER and MER of foreign words for grapheme-based (graph), transliterated grapheme-based (g-trans) and phoneme-based (phone) Vietnamese systems.*

	ins	del	sub	cor	ffr	% WER	% MER
graph	0	16	75	4	14	95.8	98.6
g-trans	2	13	67	15	30	86.3	84.5
phone	2	12	64	19	28	82.1	80.4

For the spelled out words category, the category-specific WER and MER for grapheme-based ASR measured 87.9%. Interestingly the grapheme-based system caused no foreign false recognitions. Transliteration gives a considerable reduction in category-specific

[3] C_1=[p, b, ph, v, m, u, o, qu, w, t, th, đ, x, gi, n, l, tr, s, r, ch, d, nh, y, c, k, q, kh, g, gh, ng, ngh, h]
V=[a, ă, â, e, ê, i, o, ơ, ô, u, ư, y, ai, ao, au, âu, ay, ây, eo, êu, ia, iê, yê, iu, oa, oă, oe, oi, ôi, ơi, ua, uô, uâ, ưa, ươ, uê, ui, ưi, uơ, ưu, uy, iêu, oai, oay, uôi, uyê, ươi, ươu, yêu, uây]
C_2=[p, m, t, n, ch, nh, c, ng, ngh]

WER of 18.2% absolute when compared against the baseline grapheme-based system. The reduced category-specific WER of 69.7% is equivalent to the category-specific WER of the phoneme-based system and much closer to the 63.8% overall WER of the grapheme-based system.

TABLE 5.7: *Insertions (ins), deletions (del), substitutions (sub), correct words (cor), foreign false recognitions (ffr), WER and MER of spelled out words for grapheme-based (graph), transliterated grapheme-based (g-trans) and phoneme-based (phone) Vietnamese systems.*

	ins	del	sub	cor	ffr	% WER	% MER
graph	0	9	20	4	0	87.9	87.9
g-trans	2	7	14	12	13	69.7	65.7
phone	2	8	13	12	16	69.7	65.7

Transliteration of spelled out single characters did not yield any notable improvements, but results are included for the sake of completeness.

TABLE 5.8: *Insertions (ins), deletions (del), substitutions (sub), correct words (cor), foreign false recognitions (ffr), WER and MER of spelled out single characters for grapheme-based (graph), transliterated grapheme-based (g-trans) and phoneme-based (phone) Vietnamese systems.*

	ins	del	sub	cor	ffr	% WER	% MER
graph	192	636	866	829	735	72.7	67.1
g-trans	223	626	847	858	805	72.8	66.4
phone	212	586	845	900	828	70.5	64.6

5.5 Conclusion

In the previous chapter it was established that certain irregular word categories, namely (1) spelled out words, (2) proper names and (3) foreign words, are responsible for grapheme-based performance degradation. We found that by transliterating words from irregular categories, grapheme-based ASR performance can be improved considerably.

We developed a P2G model to adapt the spelling of these word categories. The P2G model was designed to be of limited complexity in order to prevent it from capturing too much spelling detail. We wanted the P2G model to capture the 'default' behaviour of the language (which is regular), so that when applied to irregular words, their spelling could be 'regularised'. Given a phoneme string, the orthography of an irregular word can then be transformed to an idealised orthography better suited to incorporation in a grapheme-based system.

We verified the P2G transliteration technique by transliterating all the words in the entire phonemic dictionary and by calculating the difference between the original spellings

and transliterations. As anticipated, our restricted P2G models accurately modelled 'regular' words (generic Afrikaans words) to a much greater extent than irregular words (spelled out words, proper names and foreign words).

Irregular word categories were first transliterated in isolation before combining the best performing categories into a single system. During isolated transliteration, we found that some categories benefitted by adding the original orthography of the word as a 'spelling variant', while for other categories (spelled out words) it was best to replace the original orthography with a transliteration.

An improved grapheme-based ASR system was developed using a combination of all the transliterated word categories that gave a performance increase over the baseline grapheme-based system. We ensured that transliterated categories did not negatively affect categories that were not transliterated. (A small reduction in performance was observed in some of the categories, but this was negated by the total gain in performance.)

The effect of language modelling was investigated, where we found that some of the gains obtained by the transliterated system were retained when using an SLM during recognition.

It is also highly encouraging that a very small pronunciation dictionary (as few as 100 words) was found to be sufficient to train P2G models required for transliteration.

Finally, category-based P2G transliteration was evaluated and analysed on an additional language, Vietnamese. Irregular word categories were identified automatically and transliterated using a P2G model trained on generic in-language words. Improvements in category-specific WER and MER were observed when comparing the default grapheme-based systems against the transliterated system.

Herewith we confirm that P2G transliteration, when applied to irregular word categories, can improve the performance of grapheme-based ASR. The next chapter concludes with a summary of our observations and findings.

Chapter 6

Conclusion

The main aim of this study was to determine whether it is possible to improve grapheme-based ASR through practical interventions at the lexical level. In this chapter we summarise our main findings and observations, discuss the significance of this study, and explore possibilities for future work.

6.1 Introduction

In this study, we proposed a technique for improving grapheme-based ASR by transliterating words from irregular word categories. Our focus is on word categories, as this makes it possible to identify a large percentage of problematic words prior to system development. Identifying irregular pronunciations from orthography alone is not necessarily possible, but foreign words can typically be identified using known word lists in various languages and spelled words typically have a known structure. Similarly, related work from text-based named entity recognition might be useful in identifying possible proper names in transcriptions.

We compared the performance of grapheme-, phoneme-, G2P- and transliterated grapheme-based ASR systems for Afrikaans. The initial baseline experiments showed that as more training data becomes available, at a context level of three (using triphones or trigrams), minimal effort grapheme-based ASR approaches the performance of a phoneme-based system developed using a hand-crafted dictionary.

The remaining discrepancy in WER of grapheme-based ASR is primarily caused by very specific word categories. We demonstrated how these irregular word categories can be transliterated and incorporated in a grapheme-based ASR system, with much less effort than is required to develop a phoneme-based system.

During transliteration the original orthography of irregular words is transformed to an orthography more amenable to incorporation in a grapheme-based ASR system. Using P2G transliteration, grapheme-based ASR performance comparable to that of a phoneme-based ASR system developed using a hand-crafted dictionary was achieved.

While the P2G transliteration technique was developed on Afrikaans, it was also applied to Vietnamese. Irregular word categories were automatically identified and transliterated using a P2G model trained on the remaining words. Transliteration successfully caused a reduction in category-specific WER and MER for both spelled out words and foreign words.

Our main findings and observations are summarised in Section 6.2.

6.2 Summary of findings

We compared grapheme- and phoneme-based ASR systems to determine the cause of grapheme-based performance degradation. During this comparison we found that, for the Afrikaans case study:

- Grapheme-based performance degradation is caused by specific word categories, the most important ones being: (1) foreign words, (2) spelled out words, and (3) proper names. These categories are mostly easily identifiable and all tend to be irregular.
- The more training data that is available, the less the difference in performance between grapheme- and phoneme-based ASR becomes.
- With 20 hours of training data, Afrikaans grapheme-based ASR is able to outperform state-of-the-art G2P-based performance

The P2G transliteration technique was developed to 'regularise' words from irregular categories. Words are transliterated using a P2G model that captures the default (regular) behaviour of a language. While analysing the P2G transliteration technique (using Afrikaans data) we observed that:

- A limited detail P2G model, accurately models most expected regular words. Conversely, it accurately models more irregular words to a more limited degree.
- The *spelled out words* category is the most irregular category, followed by *foreign words* as the second most irregular category. *Generic Afrikaans words* is the most regular category.

- Different approaches can be followed to transliterate word categories. The original spellings of words can either be replaced with transliterations, or the original spellings can be added as 'spelling variants'. Adding spelling variants proved to be beneficial for proper names.
- For easily confusable categories, specifically foreign words, we found not transliterating short words to be beneficial.
- The biggest reduction in WER is achieved when transliterating spelled out words.
- Combining transliterated word categories into a single ASR system, we observed that:
  - Grapheme-based ASR performance can be improved considerably, reducing WER from 40.19% to 38.66%.
  - The total performance gain when combining transliterated categories (1.54% absolute), is greater than the sum of isolated gains (1.39% absolute).
  - Some performance gains from P2G transliteration are maintained when using a SLM.
  - Grapheme-based performance comparable to phoneme-based performance can be achieved.
- A very small pronunciation dictionary (as few as 100 words) is sufficient to train P2G models required for transliteration.

In order to establish if our results were transferable to a different language, we applied P2G transliteration to Vietnamese. Evaluating the P2G transliteration technique on Vietnamese showed that:

- Standard grapheme-based ASR is able to outperform phoneme-based ASR.
- Irregular categories, specifically (1) spelled out words and (2) foreign words, can be identified automatically. Spelled out words have a known structure and foreign words do not conform to Vietnamese morphology.
- Transliterating words from these irregular categories reduces category-specific WER and category-specific MER.
- The greatest reduction in category-specific WER and MER is observed when transliterating spelled out words.
- Transliterating spelled out single characters can introduce some confusion.

During the data preparation phase we observed that it takes a considerable amount of time to verify a phonemic pronunciation dictionary. As an incidental observation, we found that identifying known constituents in compounds is a very effective method to obtain pronunciation for Afrikaans words. Using a modified version of Morfessor to break up words into constituents that already exist in a dictionary, the pronunciations of these constituents can be concatenated to form new pronunciations. This method also proved to be highly accurate, with 94.6% of words having correct pronunciations.

6.3 Significance of contribution

In phoneme-based ASR, phonemes serve as an intermediate mapping between pronunciation – that which is being modelled – and spelling of words. Grapheme-based ASR relies on the regular G2P relationship of a language, using graphemes directly as the linguistic units to model. Grapheme-based ASR also has certain advantages:

- Grapheme-based ASR can model all pronunciation variation at the acoustic level, nullifying the need to craft pronunciation variants to accommodate different dialects.
- No linguistic knowledge is required to interpret ASR results.
- In resource-scarce environments, grapheme-based ASR allows for the rapid development of speech technologies.
- Used in system combination, grapheme-based ASR provides additional information and can improve ASR performance.

Many resource-scarce languages, specifically those that have recently adopted a writing system, are believed to be fairly regular [50], making them prime candidates for grapheme-based ASR. Resource-scarce languages often co-exist with more dominant languages, resulting in a high frequency of code-switched words. (See for example [51].) Proper names and code-switched words, specifically English words, tend to be highly irregular, negatively affecting grapheme-based performance and limiting the scope of ASR applications. (One cannot use an ASR system for call routing if the system cannot accurately recognise proper names.) Being able to adapt the spelling of these irregular words to conform to the 'default' spelling system of a language makes them much easier to model [52].

To address grapheme-based inadequacies, research typically focuses on creating sophisticated grapheme-based modelling techniques, or on improving existing techniques (see

Section 2.5). The technique presented in this study uses standard speech recognition approaches. We do not change 'how' words are modelled but rather 'what' is being modelled.

Based on the findings of this study, we now have a different way of thinking about the problem. For example, if expert linguistic knowledge is unobtainable for a specific resource-scarce language, it becomes impractical to train P2G rules and to obtain pronunciations for irregular words as well. We can, however, modify the transliteration mechanism. Instead of using a P2G model and phonemic pronunciations, people can be employed to provide transliterations based on their perceived knowledge of the relationship between the sounds of a language and its spelling system. These manual transliterations can then be incorporated directly in a grapheme-based system. Given enough data, over time, the acoustic models themselves can even be used to transliterate irregular words.

Contributions from this study include two publications:

- Grapheme- and phoneme-based ASR systems were compared, and word categories responsible for grapheme-based performance degradation identified in [53].
- Category-based P2G transliteration was developed and analysed in [52].

This study also contributes a categorised word list for Afrikaans containing 9 375 unique words.

6.4 Future work

In future work, we aim to investigate the extent to which these results are transferable to different languages. We would like to determine objective measures for when the original orthographic form of a word should be replaced with transliterations and when it should be kept as a spelling variant. We are also interested in the sensitivity of the proposed technique to different P2G algorithms.

Our main interest lies in the possibility of bypassing phonemes altogether during transliteration. Immediate future work will include manual transliteration of irregular words using a crowdsourcing platform. We intend to have non-native language speakers transliterate problematic words 'phonetically', that is, spelling the words according to how they sound, given some limited knowledge about the target spelling system.

Ultimately, we would like to establish if, given sufficient data, it may be possible to train G2G rules from transliterated dictionaries, or to train such rules based on transliterations made directly by ASR systems.

6.5 Conclusion

Typically, grapheme-based ASR systems are negatively affected by words from irregular categories. P2G transliteration can be used to 'regularise' the spelling of irregular words and can improve the performance of grapheme-based ASR.

Bibliography

[1] B. H. Juang and L. R. Rabiner. Automatic speech recognition – A brief history of the technology development. *Encyclopedia of Language and Linguistics, Elsevier*, pages 1–24, 2005.

[2] B. Mimer, S. Stüker, and T. Schultz. Flexible decision trees for grapheme based speech recognition. In *Proc. Conference Elektronische Sprachsignalverarbeitung (ESSV)*, 2004.

[3] M. Davel and O. Martirosian. Pronunciation dictionary development in resource-scarce environments. In *Proc. Interspeech*, pages 2851–2854, 2009.

[4] M. Killer, S. Stuker, and T. Schultz. Grapheme based speech recognition. In *Proc. Eurospeech*, pages 3141–3144, 2003.

[5] S. Kanthak and H. Ney. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 845–848, 2002.

[6] E. G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck. Automatic speech recognition without phonemes. In *Proc. Eurospeech*, pages 129–132, 1993.

[7] J. Dines and M. Magimai Doss. A study of phoneme and grapheme based context-dependent ASR systems. In *Proc. Machine Learning for Multimodal Interaction (MLMI)*, pages 215–226, 2008.

[8] M. Wolff, M. Eichner, and R. Hoffmann. Measuring the quality of pronunciation dictionaries. In *Proc. ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*, 2002.

[9] M. Davel and E. Barnard. Pronunciation prediction with Default&Refine. *Computer Speech and Language*, 22(4):374–393, 2008.

[10] R. Rasipuram and M. Magimai Doss. Improving grapheme-based ASR by probabilistic lexical modeling approach. In *Proc. Interspeech*, Lyon, France, Aug. 2013.

[11] M. Janda, M. Karafiát, and J. Černockỳ. Dealing with numbers in grapheme-based speech recognition. In *Text, Speech and Dialogue*, pages 438–445. Springer, 2012.

[12] Y. H. Sung, T. Hughes, F. Beaufays, and B. Strope. Revisiting graphemes with increasing amounts of data. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4449–4452. IEEE, 2009.

[13] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2nd edition, 2008. ISBN 0131873210.

[14] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[15] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

[16] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738, 1990.

[17] J. C. Wells et al. SAMPA computer readable phonetic alphabet. *Handbook of standards and resources for spoken language systems*, 4, 1997.

[18] A. Stolcke et al. SRILM-an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, Colorado, USA, September 2002.

[19] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proc. IEEE*, 88(8):1270–1278, 2000.

[20] M. Bisani and H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. Interspeech*, 2002.

[21] O. Andersen, R. Kuhn, A. Lazarides, P. Dalsgaard, J. Haas, and E. Noth. Comparison of two tree-structured approaches for grapheme-to-phoneme conversion. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 1700–1703, Philadelphia, USA, 1996.

[22] A. Black, K. Lenzo, and V. Pagel. Issues in building general letter to sound rules. In *3rd ESCA Workshop on Speech Synthesis*, pages 77–80, Jenolan Caves, Australia, November 1998.

[23] P. Taylor. Hidden Markov models for grapheme to phoneme conversion. In *Proc. Interspeech*, pages 1973–1976, 2005.

[24] M. J. Dedina and H. C. Nusbaum. PRONOUNCE: a program for pronunciation by analogy. *Computer Speech and Language*, 5(1):55–64, 1991.

[25] Y. Marchand and R. I. Damper. A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26(2):195–219, 2000.

[26] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex systems*, 1(1):145–168, 1987.

[27] W. Daelemans, A. Van Den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–41, 1999.

[28] K. Torkkola. An efficient way to learn English grapheme-to-phoneme rules automatically. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 199–202, Minneapolis, USA, April 1993.

[29] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

[30] R. Kneser and H. Ney. Improved backing-off for M-gram language modeling. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 181–184, 1995.

[31] B. Réveil, J-P. Martens, and H. van den Heuvel. Improving proper name recognition by adding automatically learned pronunciation variants to the lexicon. In *Proc. Conference on International Language Resources and Evaluation (LREC)*, pages 2149–2154. European Language Resources Association (ELRA), 2010.

[32] N. Cremelie and L. ten Bosch. Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters. In *Proc. ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

[33] B. Maison, S. F. Chen, and P. S. Cohen. Pronunciation modeling for names of foreign origin. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 429–434. IEEE, 2003.

[34] Q. Yang, J-P. Martens, N. Konings, and H. van den Heuvel. Development of a phoneme-to-phoneme (p2p) converter to improve the grapheme-to-phoneme (g2p) conversion of names. In *Proc. Conference on International Language Resources and Evaluation (LREC)*, pages 287–292, 2006.

[35] M. Magimai Doss, S. Bengio, and H. Bourlard. Joint decoding for phoneme-grapheme continuous speech recognition. In *Proc. IEEE International Conference*

on Acoustics, Speech, and Signal Processing (ICASSP), volume 1, pages 177–180. IEEE, 2004.

[36] M. Magimai Doss, T. A. Stephenson, H. Bourlard, and S. Bengio. Phoneme-grapheme based speech recognition system. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 94–98. IEEE, 2003.

[37] M. H. Davel. *Pronunciation modelling and bootstrapping.* PhD thesis, University of Pretoria, 2005.

[38] N. J. de Vries, J. Badenhorst, M. H. Davel, E. Barnard, and A. de Waal. Woefzela - an open-source platform for ASR data collection in the developing world. In *Proc. Interspeech*, pages 3176–3179, August 2011.

[39] J. Badenhorst, A. de Waal, and F de Wet. Quality measurements for mobile data collection in the developing world. In *Proc. Spoken Languages Technologies for Under-Resourced Languages (SLTU)*, pages 139–145, Cape Town, South Africa, 2012.

[40] M. Davel and F. de Wet. Verifying pronunciation dictionaries using conflict analysis. In *Proc. Interspeech*, pages 1898–1901, Tokyo, Japan, 2010.

[41] M. Creutz and K. Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor. *Publications in Computer and Information Science, Report A*, 81, 2005.

[42] G. B. van Huyssteen and M. M. van Zaanen. Learning compound boundaries for Afrikaans spelling checking. In *Pre-Proc. Workshop on International Proofing Tools and Language Technologies*, pages 101–108, July 2004.

[43] A. C. Morris, V. Maier, and P. Green. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2004.

[44] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Veltchev, and P. Woodland. *The HTK Book.* Cambridge University Engineering Department, http://htk.eng.cam.ac.uk/, 2005.

[45] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[46] M. H. Davel, C. J. van Heerden, and E. Barnard. Validating smartphone-collected speech corpora. In *Proc. Spoken Languages Technologies for Under-Resourced Languages (SLTU)*, pages 68–75, 2012.

[47] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. Association for Computational Linguistics (ACL)*, pages 310–318. Association for Computational Linguistics, 1996.

[48] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The Kaldi speech recognition toolkit. In *IEEE workshop on automatic speech recognition and understanding*, 2011.

[49] Wikipedia. Vietnamese language, 2013. URL `http://en.wikipedia.org/wiki/Vietnamese_language`.

[50] A. W. Black and A. F. Llitjos. Unit selection without a phoneme set. In *Proc. IEEE Workshop on Speech Synthesis*, pages 207–210. IEEE, 2002.

[51] T. Modipa, F. de Wet, and M. H. Davel. Implications of Sepedi/English code switching for ASR systems. In *Proc. Pattern Recognition Association of South Africa (PRASA)*, pages 64–69, 2013.

[52] W. D. Basson and M. H. Davel. Category-based phoneme-to-grapheme transliteration. In *Proc. Interspeech*, pages 1956–1960, Lyon, France, Aug. 2013.

[53] W. D. Basson and M. H. Davel. Comparing grapheme-based and phoneme-based speech recognition for Afrikaans. In *Proc. Pattern Recognition Association of South Africa (PRASA)*, pages 144–148, 2012.