Graphon: A comparison of grapheme-to-phoneme conversion performance between an automated system and primary grade students

by

Colette Joubarne

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Masters Degree in
Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Abstract

Grapheme-to-phoneme conversion is a necessary part of reading, whether by an automated system or by children. Automated methods play a key role in text-to-speech and automated speech recognition systems. Children learning to read develop grapheme-to-phoneme (G2P) conversion rules that they use extensively until they build up their orthographic lexicon.

Various solutions have been proposed for G2P conversion, each addressing specific problems and evaluated for different languages. In this thesis, I introduce a simple approach to G2P conversion that achieves good results, and compare these results to those of a study of children's reading accuracy in the primary grades. The comparison highlights areas of weakness in the children's reading skills, as well as particular phonemes for which the G2P system has difficulty. As part of the process, I also compare and discuss the wide range of discrepancies that exist between various French corpora.

Acknowledgments

I would like to thank those who made this thesis possible.

I am very grateful to my supervisor, Dr. Diana Inkpen, for her encouragement and guidance. Without her, this would have been a path not taken. I would also like to thank Dr. Alain Desrochers for his support and assistance in all things linguistic, as well as providing the link that I needed to pique my interest. Thank you to Dr. Stan Szpakowicz, whose comments pushed me beyond my self-imposed limits.

I would like to thank Dr. Greg Kondrak, from the University of Alberta, for providing the alignment algorithm, Aline, and Paul Huff for his Python implementation, PyAline.

Thank you to my reviewers, Dr. Jo McCutcheon, Kathy Lavigne, and Dr. Varada Kolhatkar.

I acknowledge support from Social Science Research Council (SSHRC) and to the Natural Sciences and Engineering Research Council (NSERC) of Canada for supporting this research work.

Most importantly, I would like to thank my husband and children for their encouragement, understanding, and most of all their patience on this long journey.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

Learning to read is a fundamental part of our education system. If a child is not able to achieve a proper foundation at the beginning, it can have an impact on their entire school career as well as their future. Teachers have many tools at their disposal, including pedagogical methods, student assessments, and grade-level literature, and yet roughly 40% of children fall behind their peers in reading levels, while only 5% actually have serious reading disabilities (Willows, 2008). If a child is not sufficiently skilled at reading by the end of Grade 3, they are unlikely to graduate from high school, a serious concern for parents and educators (Snow, Burns, & Griffin, 1998). In this thesis, I propose a computational system that simulates phonological processing of an early reader. For the purpose of this research, phonological processing refers to the process of sounding out a word without the benefit of lexical cues such as previous exposure to a word or any form of context.

Teaching approaches can be classified into three broad categories: “reading as a hierarchy of skills, reading as a psycholinguistic process, and reading as social practice” (Emmit, 1998, p. 2). According to Woolacott (2002), skills refer to the physical reading process, whereas the psycholinguistic process focusses on the learner and cognition. The social

and cultural context of readers and text characterizes reading as a social practice. The three physical early reading skills – phonemic awareness, decoding, and vocabulary combined are required to master the use of spelling-to-sound correspondence (Snow, Burns, & Griffin, 1998) which, according to much research, is the most difficult step in learning to read. According to Snow *et al*., "Cognitive studies of reading have identified phonological processing as crucial to skillful reading, and so it seems logical to suspect that poor readers may have phonological processing problems." The objective of this thesis is to determine whether it is possible to develop a system that simulates the phonological processing of an early reader and discover what we can learn during the process. Such a system can be used to test new pedagogical teaching methods for reading and provides a method to isolate problem areas during assessment, which could lead to targeted learning methods.

The foundation of the system is grapheme-to-phoneme conversion, which requires knowledge of a phonemic representation, generation of spelling-to-sound conversion rules, and a vocabulary as a training set. According to Ziegler et al. (2013), most previous computational models of reading have used a large vocabulary as part of a supervised training regimen. They maintain that this is highly implausible in reality, and propose a learning loop that begins with a small set of grapheme-phoneme correspondences. Rather than training with all of the words that should be learnt, I have limited the vocabulary used in my system to the words a child of a specific age has been exposed to.

1.2 Grapheme-to-Phoneme Conversion

There are two basic approaches to grapheme-to-phoneme (G2P) conversion. The first is a rule-based approach that involves manually identifying a set of rules for converting graphemes into phonemes. While this approach does not require training data and can produce quite accurate results, it requires in-depth knowledge of linguistic patterns of a given language, and a new set of rules must be developed for each language. Yvon *et al*. (1998) present an excellent comparison of eight rule-based systems for French, highlighting both their strengths and weaknesses, and the fact that they require 500 to 4000 rules, as well as look-ups for exceptions. All of this data must be carefully maintained and can be quite fragile if changes are required as a result of exceptions or additions to a language.

The second approach is to use machine learning techniques to generate rules. This approach does not require linguistic knowledge of a particular language and can easily be applied to different languages provided the training data is available. The rule generation can be fully automatic or can start with a small amount of hand seeding, which, as shown by Black *et al*. (1998) can improve the results. Rule generation can also be static or grow as more examples are encountered. Yvon (1994) compares various instance-based (IBL) and analogy-based learning (ABL) techniques. The main difference between the two techniques is that the first attempts to identify the regularities in the training data, whereas the second collects examples to be used later when a similar word is encountered.

These two techniques, IBL and ABL reflect the debate in the linguistic community regarding whether readers rely solely on lexical information (Glushko, 1979) or whether they use a dual-process (Baron & Strawson, 1976), which involves a lexical lookup for known items and the use of pronunciation rules for infrequent words. While it may be possible that adult readers rely solely on lexical information, it is unlikely that children learning to read do so, as their orthographic lexicon is initially empty. Typically, experiments in favor of the single-process approach to reading, such as those by Glushko (1979), and Kay & Marcel (1981) are conducted on adults who have had the opportunity to develop a comprehensive orthographic lexicon. It is more probable, as proposed by Sprenger *et al*. (1998), that early readers use the pronunciation rules to construct their lexicon.

In this project, I am attempting to simulate a child's ability to learn to read, as opposed to a text-to-speech system (TTS), which more closely simulates the reading patterns of an adult, who has a large orthographic lexicon, and only rarely encounters a word that was not seen before. Therefore, the TTS should take advantage of the much larger lexicon for lexical lookups, which are much faster than processing of phonological rules in humans, as well as for analogies, since there is a greater likelihood of a close match. In comparison, this system is simulating an environment in which the lexicon is relatively small, and the child is being tested on words that often have not been encountered.

Therefore, I have chosen to rely solely on pronunciation (or grapheme-to-phoneme conversion) rules to develop this system. This approach uses the frequency of occurrence of patterns to identify the resulting rules, which lends itself well to the theory that

exposure to words (or specific phonetic sequences) increases the likelihood that a person will be able to recognize it. This pattern was first noted by Cattell (1886). Cattell demonstrated that the frequency of occurrence of a word in a language affects the speed of recognition, but that there is a limit, beyond which further exposure has little or no effect.

1.3 Objectives

The purpose of this work was to improve upon Graphon, a graphon-to-phoneme conversion system (Martin & Inkpen, 2006) and to develop a computational system that simulates the phonological processing of an early reader. Additional goals were to develop a measure of the level of difficulty of words, and to identify errors in the Omnilex dataset.

1.4 Contributions

I have developed a system that provides a method to isolate problem areas during children's reading assessment. In turn, this could lead to targeted learning methods, as well as provide an automated approach to determine the reading level of children's literature. This included streamlining Graphon and improving its word accuracy, as well as improvements to PyAline (Huff, 2010). As part of the process, I also examined the benefits of weighted attributes versus attribute engineering.

To test my system, I have created a data set that represents the errors made during a reading assessment of 180 French children in Grades 1 – 3 (Carson & Desrochers, 2012). The reading assessment was performed by graduate students of the University of Ottawa Psychology department. The dataset was created by inputting data collected by the graduate students, since it was available only on paper. I also analyzed the data, in order to identify areas of difficulty. This analysis can be used to assign a level of difficulty to the word set.

The system also provides feedback to the input data sets regarding possible incorrect phonetic transcriptions and an analysis of the differences between commonly used French data sets.

1.5 Outline

In Chapter 1, I discuss work related to grapheme-to-phoneme conversion, text-to-speech systems, and issues related to children's literacy. In Chapter 3, I present the various data sets used in this project: Omnilex, Lexique3, Brulex, and Manulex. Chapters 0, 5 and 6 introduce the Graphon project, possible improvements and results. In Chapter 7, I propose the use of Graphon as a simulation system and analyze the results. Finally, in Chapter 8, I conclude with a review of contributions and learnings achieved as a result of the project, as well as a discussion of future work.

2 Related Work

2.1 Grapheme-to-Phoneme Conversion

Grapheme-to-phoneme conversion (G2P) is the process of converting a sequence of letters (graphemes) into a sequence of sounds (phonemes). This process is an integral part of text-to-speech (TTS) systems, as well as automatic speech recognition (ASR) systems.

Text-to-speech research within the domain of Natural Language Processing can roughly be categorized into three areas: transcription of individual words, pronunciation of words within context, and methods to reduce the training data required to generate results automatically.

Transcription of individual words is addressed by various grapheme-to-phoneme conversion methods. Typically, G2P conversion focusses solely on individual words, whereas phonetic pronunciation modelling addresses the issues of effects of words together. These issues include liaisons, muted 'e', homographs, lexical stress, speed of speech, and speaker's accent (see section 2.1.1 for a description of these issues). All of which are dependent on the context of a word and the origin of the speaker. Riley *et al*. (1999) use a tree-based dictionary to capture multiple pronunciations. Dou *et al*. (2009) use a data-driven approach to create a Support Vector Machine ranker to assign stress to the correct syllable.

Creation of training data is a tedious process. For many languages, limited data is available; therefore, any method that either helps in producing training data, or reduces the requirement for it, is beneficial. Phonetic base form generation involves expanding the training data using spoken samples. Bahl *et al*. (1991) combine acoustic samples and spelling-to-sound rules to automate the deduction of the phonetic base form to add new words to the vocabulary of a TTS system. Dwyer and Kondrak (2009) show that it is possible to achieve reasonable results with limited training data. They achieve an accuracy of 57.3% using 40,000 words compared to the accuracy 57.8% Black *et al*. (1998) achieved with 90,000 words.

Considerable research has been done on grapheme-to-phoneme conversion under various names. Letter-to-sound conversion applies to languages with a defined alphabet such as English and French. Grapheme-to-phoneme conversion expands this to languages that include Chinese characters, numerical digits, punctuation marks, and other symbols used in any of the world's writing systems. However, this distinction is not made in the literature, and the two terms are often used interchangeably, as are phonetic transcription, text-to-phoneme mapping, and other variations. Pagel *et al*. (1998) use top-down induction trees to generate letter-to-sound rules for English and French corpora. Andersen *et al*. (1996) compare two tree-structured approaches to the problem of grapheme-to-phoneme conversion. Seng *et al*. (2011) use a two-stage neural network to generate a couple of letter-to-phoneme conversion options, and then to predict the final output.

The task has been applied to various languages: French (Laurent, Meignier, & Deléglise, 2014), English (Bartlett, Kondrak, & Cherry, 2008), Arabic (Ramsay, Alsharhan, & Ahmed, 2014), Dutch and German (Jiampojomarn, Kondrak, & Sherif, 2007), etc.

Languages vary in their complexity with respect to grapheme-to-phoneme mapping. Divay *et al*. (1997) divide languages into three different systems: simple, mid-level, and complex. They describe Spanish and Swahili as simple systems that typically have undergone a spelling reform, resulting in well-defined letter-to-sound rule sets. While German has a large morphological system, it is described as a mid-level system, since it also has well-defined letter-to-sound rules. In contrast, English and French fall in the complex system category, being the most difficult languages for which to create letter-to-sound rules.

Each language presents its own difficulties, so while the intent of this project is to develop a tool that is applicable across languages, the student data available for comparison was only in French; therefore, the system has only been trained on French corpora. As such, further discussion is focused on issues particular to the French language and state-of-the art systems trained and tested in French.

2.1.1 Grapheme-to-Phoneme Conversion for French

Yvon *et al*. (1998) describe the French orthographical system as overly complex and full of irregularities. For example, even though the following all have different meanings, they are all pronounced the same (Pontes & Furui, 2010).

```
"vin" ("wine")
"vins" ("wines")
"vingt" ("20")
"il vint" ("he came")
"je vaincs"("I won")
"en vain" ("in vain")
```

Yvon *et al*. (1994) provide considerable detail about the state of the research in G2P conversion, which they claim did not include any research in French (at the time). In their review of rule-based approaches, they highlight problems such as acronyms, numbers and symbols, proper names, loan words, and sentence correctness, which takes into consideration the problems related to liaison. *Liaison* or *morphophonemics* (Divay & Vitale, 1997) is the pronunciation of an otherwise silent final consonant of a word in certain contexts. For example: *mes doigts* [mèdwa][1] vs. *mes amis* [mèzami]. In the first case the "s" in "mes" is latent, but not in the second. Pontes & Furui (2010) developed a prototype using C4.5 decision trees that improved phoneme prediction of *word final consonants* (or *liasons*).

The problem of a muted 'e', known as *elision of schwa* is specific to French. It can occur in the beginning, middle, or end of a word. Consider the following example, from (Divay & Vitale, 1997):

[1] This and all other transcriptions in this document use the International Phonetic Alphabet (IPA).

Muted ‘e’	Non-muted or added ‘e’
la poule [pul] jaune	*la poule* [pule] *est jaune*
petit [pti]	*Ouest-France* [wèste frâs]
tellement [tèlmâ]	*justement* [Zystemâ]

Transcription rules are general and do not always apply when speech is slowed down or sped up. In French, syllables may be added in slow or emphatic speech, and lost in fast or familiar speech. For example, *parce que* [parskö] can go from two to three syllables [parsekö] when slowed down, and *je te le dirai* [jeteledirè] can go from five to four [jetledirè], or even three [jteldirè] syllables when sped up.

In some cases, words have to be decomposed into morphemes for correct grapheme-to-phoneme conversion. This is referred to as *morphology* (Divay & Vitale, 1997). Usually in French, an ‘s’ between two vowels is pronounced [z], otherwise [s]. Words such as *tournesol*, *contresens* and *antisocial*, each need to be considered as two separate morphemes in order to transcribe them correctly. This morpheme decomposition is difficult.

Homographs are a problem not only in French. They occur when the same spelling results in different transcriptions. Yarowsky (1997) identifies seven major classes of homographs, only three of which apply to words (as opposed to numerics) – different parts of speech, same part of speech and proper names. For example: the pronunciation is sometimes a function of the grammatical category, which requires part-of-speech tagging as in the case of *fier* (‘proud’ or ‘to trust’), but may sometimes require a deeper semantic

understanding, as in the case of *fils* ([fis] ‘son’ vs. [fil] ‘thread’). Yarowsky uses colocation patterns and decision lists to disambiguate between different transcriptions.

Unlike English and a few other languages, French does not have lexical stress. While the meaning of a phrase can depend on which words are emphasized (phrasal-stress), the meaning of a word is not affected by the stress placed on a particular syllable.

Given that the purpose of this research is to compare the results with those of single word tests performed on students, only morphology is relevant to further discussion. The issues of liaisons, muted ‘e’ and speech rate only apply to phrases, and are therefore only relevant to complete TTS systems. Disambiguation of homographs requires context, which is also only relevant to a complete TTS system. Morphology comes into play during data set creation and the conversion process. A discussion on how this was handled in this project can be found in sections 3.3 and 4.4.

Another special case that is worth discussing is the handling of proper nouns. Accurate phonetic transcription of proper nouns is necessary for commercial applications of TTS technology. However, given today’s globalization, names can incorporate phonetic rules from various languages outside the language of study. As a result, the phonetic transcription of a proper noun is dependent on both the origin of the speaker and the origin of the noun itself (Laurent, Meignier, & Deléglise, 2014). Boula de Mareuil (2005) presents a comparison of four systems on 1500 proper nouns, with results ranging from 13% to 19% word error rate (WER). According to a study by Dufour (2008), errors transcribing proper nouns can also affect the accuracy of the surrounding words. He

shows that for out-of-vocabulary words (which include proper nouns) the error rate on words within a window of five words on either side can increase by up to 45%. Since the children in the study were only tested on general vocabulary, the issue of proper nouns is not addressed in this project.

2.1.2 State-of-the-Art in Grapheme-to-Phoneme Conversion

Letter-to-phoneme conversion has been researched extensively using a variety of techniques, each focusing on various problem areas. Results vary greatly depending on the language, the training being used, the type of test data being evaluated, and the amount of context being provided. Limiting our search to research performed using publicly available French corpora, we find that the recent experiments executed by Jiampojamarn *et al*. (Jiampojamarn & Kondrak, 2010), (Jiampojomarn, Cherry, & Kondrak, 2008), (Jiampojomarn, Kondrak, & Sherif, 2007) achieve the best results. They achieve an improvement of 1.2% and 0.3% using 1:1 alignment and M:M alignment by applying a supervised HMM method to the phoneme prediction. While they achieve some gains using an M:M aligner (see section 2.2), the greatest gains are seen using joint processing and online discriminative training, where they achieve a word accuracy of 94.5% on the Brulex data set. This is an iterative process that involves scoring each possible phoneme, searching for the highest scoring phoneme, and applying an update equation that will move the model away from incorrect outputs and toward the correct output.

While my approach was much simpler, using 1:1 alignment and basic machine learning algorithms, I achieved results comparable to other approaches, except for the joint processing and online discriminative training (Jiampojomarn, Cherry, & Kondrak, 2008).

2.2 Alignment

Any automated system first requires that the training data be aligned, letter-to-phoneme. Suontausta & Häkkinen (2000) use the Viterbi algorithm, along with a pronunciation dictionary to find the best alignment for their text-to-phoneme conversion. The pronunciation dictionary is language dependent.

ALINE (Kondrak, 2000) is an alignment algorithm that was developed for cognate alignment, but has been used by Jiampojamarn *et al*. (2008) and Huff (2010) for grapheme-to-phoneme alignment. Unlike most alignment algorithms, it is based on a similarity score rather than a distance score. The score is based on many of the articulatory features of the phonemes.

In some languages such as Spanish, there is almost a 1:1 mapping between letter and phoneme, while in others such as English and French, there can also be a 0:1, 1:0, M:1, and 1:M mapping between letters and phonemes (see section 4.1 for examples). Typically, these are addressed by the insertion of hyphens during alignment, and all alignments are assumed to be 1:1. Jiampojamarn *et al*. (2007) used a many-to-many aligner followed by a letter chunking bigram predictor to overcome this problem and achieved improvements in word accuracy of 3.3% to 90.6% over their baseline on the

Brulex data set. Jiampojamarn *et al*. (2008) followed this work with an aggregated alignment algorithm that first chose the best n alignments, and then used these to create a M:M alignment to be used by the M2M aligner, which along with their online discriminative systems yields results of 95.07% on the Brulex dataset.

GIZA++ is an open source toolkit that can handle both 1:1 and M:M alignments. It has been used by Rama *et al*. (2009) and Seng *et al*. (2011) in their letter-to-phoneme conversion using Statistical Machine Translation techniques and neural networks, respectively.

I opted to use the ALINE algorithm for this project for two reasons. First, it was used in the initial Graphon prototype, so its continued use allowed for valid comparison with the improvements made to the system, and switching to M:M alignment would have required a complete redesign of the system. Second, of the 1:1 alignment solutions available, the best results to date (Jiampojamarn & Kondrak, 2010) were achieved with Aline.

2.3 Machine Learning

Once the training data has been aligned, it needs to be examined so that the rules for grapheme-to-phoneme conversion can be learned. This process is called machine learning. "Machine learning is programming computers to optimize a performance criterion using example data or past experience." (Alpaydin, 2010). Machine learning is not only used in text-to-speech systems, it is used in various domains such as data mining, vision (i.e., facial recognition, recognition of facial expressions), text

classification, and robotics. Kranjc *et al*. (2015) used an active learning approach with a Support Vector Machine (SVM) to detect authors' attitude, emotions, and opinions from texts in real-time. Sultan *et al*. (2015) use an ensemble (ordered collection) classifier to achieve a 98.6% accuracy of facial expression recognition. Sebastiani (2002) presents the main machine learning approaches to text classification and their advantages over expert systems. These advantages include portability of domain, flexibility with respect to categories, and overall better accuracy.

There are three categories of machine learning. Supervised learning happens when a system is trained using sample data, and then once trained, it applies this learning to classify the test data. Unsupervised learning is used to cluster data into categories when there is no training data available. Reinforcement learning involves determining a sequence of actions based on initial training data and learning from past good action sequences and their results.

Learning grapheme-to-phoneme conversion rules falls into the category of supervised learning. Given the training set of aligned transcriptions, patterns are identified to generate a model that can account for most of the data. The accuracy of the model is dependent on the size and quality of the training data and the classification algorithm used. In general, no one algorithm is best, it depends on the application. Dou *et al*. (2009) used an SVM ranker to predict which phonemes to stress. Humphries *et al*. (1996) used decision trees to show that accent-specific pronunciations can improve speech recognition by 20%. Section 3.3 details the cleaning that is performed on the various

training sets to improve their quality. Section 4.2 introduces the classification algorithms used in this project.

3 Data Sets

3.1 Omnilex

The Omnilex database (Desrochers, 2006), was developed by the Psychology Department of the University of Ottawa, and contains 109,000 French words from all grammatical categories with the exception of proper nouns. The database contains information about various characteristics for each word, (i.e., structural, distributional, relational, and semantic). For the purpose of this project, only the manually annotated phonetic transcriptions were needed (see Table 1). These are specified using the APF Code (l'Alphabet Phonétique Français). See Appendix A for its correspondence with other notations.

During the original work of Martin and Inkpen (2006) on this project, only a subset of 20,000 words were used, this subset will be referred to as the Graphon Subset.

Table 1: Sample Grapheme - Phoneme Pairs from Omnilex

Graphèmes	Phonèmes
à cloche-pied	aklòSpjé
à contrecoeur	akôtrekër
à jeun	aZû
abaissé	abèsé
abaisser	abésé
abajoue	abaZu
abandon	abâdô
abandonné	abâdòné

3.1.1 Training Data vs. Test Data

Two approaches were used when training with the Omnilex data set. In some cases, the entire set was randomly divided into 10 sets, and each was used as testing, while the remaining 9 were used for training. This is known as 10-fold cross-validation. For comparison purposes with the work done by Martin and Inkpen (2006) the words used in their test set were removed from the Omnilex data set, and the remainder was used for training.

3.2 Lexique3

Lexique3 (New, Pallier, Ferrand, & Matos, 2001) is a database of 143,000 French words and their properties, such as grammatical category, gender, number, and frequency. For the purpose of this project, we are only interested in the phonetic transcription of the words (See Table 2). Lexique3 uses its own notation for phonetic transcription. See Appendix A for Lex3 code's correspondence with other notations.

Table 2: Sample Grapheme - Phoneme Pairs from Lexique3

Graphèmes	Phonèmes
abaissons	abEs§
abaissèrent	abEsER
abandon	ab@d§
abandonné	ab@done
abducteur	abdykt9R
abduction	abdyksj§

3.2.1 Training Data

For comparison with the work done by Martin and Inkpen (2006), the words used in their test set were removed from the Lexique3 data set, and the remainder was used for training.

3.3 Preprocessing of Omnilex and Lexique3

Both data sets were cleaned as follows:

- **Composite words**: Hyphens and spaces were removed since they are not handled by the alignment algorithm. Hyphenated words, such as *abat-jour* and words that always appear together, *appareil photo, aussitôt que* could have been separated and handled individually. I opted to remove the hyphen and space, and handled the composite words as one word. In some cases, the hyphens were even optional *aéro(-)club, aéroklëb.* The pronunciation would be the same with or without the hyphen or space. As discussed earlier (see

section 2.1.1 on morphology), since there exist numerous words in which the transcription is different because the word is composed of compound words, removing the existence of more compound words would only lessen the likelihood that a particular mapping would be selected. In addition, in a couple of cases, the transcription is actually different because the words appear together, for example *bon enfant* [bonâfâ], *ave maria* [avémarja]. While it is debatable whether these should be considered as words, the distinction is not always clear, i.e., *a capella* and *ad hoc.*

- **Abbreviations and numbers:** These were deleted since the phonetic transcription did not correspond to the grapheme, but to a representation of the grapheme. Abbreviations were defined as anything that contained a period, or consisted of all uppercase letters. For example, "5" is the representation of "cinq" which is pronounced "sêk", and "AM" is the representation of "A" "M" which is pronounced "aèm".

- **Variations in spelling**: Both variants were used with the same phonetic transcription.
    - bat(t)age abataZ -> abatage abataZ, abattage abataZ

- **Optional phonemes**: Phonetic transcriptions such as décevoir, dès(e)vwar indicate that the 'e' is optional. These optional phonemes were not optional in

the OmniLex-Lexique3 data set (this approach was also taken in the student word list).

- **Optional graphemes:** Optional graphemes and their corresponding phonemes were handled depending on the situation:
    - removed when not represented in the phonetic transcription (i.e. “abor(i), abòri” became “abor(i), abòri”)
    - moved to the front when that was accepted use (i.e., “croque(-)au(-)sel (à la), kròkosèl(ala)” became “à la croque(-)au(-)sel, ala kròkosèl”)
    - used (i.e. “cyn(o), sin(o)” became “cyno, sino”)
- **Missing transcriptions:** Entry was removed completely

This processing resulted in two data sets to use for training, Omnilex containing 90,000 transcriptions and Lexique3 containing 125,000.

3.4 OmniLex-Lexique3

Originally, the plan was to create the largest data set possible to be used for training by combining the Omnilex and Lexique3 databases.

The OmniLex-Lexique3 dataset was created by joining Omnilex and Lexique3 to produce a data set of approximately 163,000 words. There were however, various issues encountered when creating this combined data set.

- **Missing Phonemes**: Lexique3 does not distinguish between the APF sounds “a” and “à” as in “bas” and “patte”. As well, Lexique3 ignores both the aspirated “h” in “hop” and the non-aspirated “h” as in “haricot”.
- **Conflicts:** There were over 20,000 differences in transcriptions between Omnilex and Lexique3. This accounts for 22% of the Omnilex transcriptions. The primary differences were: the missing phoneme “à” as discussed above; a conflict between the representation of “o” and “ò”, and a discrepancy in the pronunciation of “e” in many scenarios. See Table 3 for examples and frequencies of these differences.

Table 3: Primary Differences between Omnilex and Lexique3 Transcriptions

Diff	**Word**	**Omnilex**	**Lexique3**	**Count**
à != a	bas	bà	ba	1415
e != ö	bedaine	bedèn	bödèn	3234
o != ò	berceau	bèrso	bèrsò	4998
ò != o	chlore	klòr	klor	5970
Total				**15617**

These 4 differences alone account for over 76% of the conflicts between the Omnilex and Lexique3 transcriptions.

Pierre Martin (2004) studied the variations in the pronunciation of three different vowels /i/, /y/ and /u/. He found large variations across gender, (see Table 4) confirming that manual transcription is a subjective exercise. However, such a large discrepancy is likely a result of the origin of the data sets, Omnilex developed in Canada, and Lexique3 developed in France. In his study, Pierre Martin (2004) found larger variations across regions, (see Table 5) with the largest occurring between Québec and other European countries.

Table 4: Percentage of Acoustic Devoicing by Gender and Source

Adapted from Martin (2004)

	Belgium		**France**		**Québec**		**Switzerland**	
	F	**M**	**F**	**M**	**F**	**M**	**F**	**M**
Loss of sound	0.5	0.2	1.2	0.5	9.0	7.1	0.6	1.2
Full devoicing	8.4	5.4	8.9	19.6	22.6	28.6	4.7	18.1
Partial devoicing	61.2	61.8	60.2	53.5	43.6	36.9	36.9	58.1
No devoicing	29.9	31.9	29.5	26.1	24.8	27.4	27.4	22.4

Table 5: Percentage of Acoustic Devoicing by Source

Adapted from Martin (2004)

	Belgium	**France**	**Québec**	**Switzerland**
Loss of sound	0.4	0.8	8.1	0.9
Full devoicing	6.4	14.4	25.6	11.9
Partial devoicing	61.5	57.0	40.2	62.8
No devoicing	31.3	27.5	26.1	24.2

Given that the student testing was performed on Canadian children, it was determined that the Omnilex data would therefore be the best training data. A comparison of transcriptions used in the student testing wordlist (Table 6) shows that there are fewer discrepancies between it and the Omnilex dataset than the Lexique3 dataset. This was confirmed by tests run on the test set used by the original Graphon, which had been generated from words found only in the Omnilex subset. The word accuracy was 90.1% using the Omnilex training data versus 69.4% using the Lexique3 training data.

Table 6: Student Test Wordlist Differences

Student Test WordList Differences (grapheme - expected results, study expected results)	
Omnilex	**Lexique3**
ecchymose - ékimoz ekimoz exact - ègza ègzakt maniere - manjèr majèr parole - paròl parol	automne - òton òtòn beau - bò bo beaucoup - bòku boku borne - born bòrn bourreau - burò buro causalite - kòzalité kozalité cependant - söpâdâ sepâdâ chaos - kaò kao code - kod kòd corps - kor kòr devenir - dövnir devenir eau - ò o ecchymose - ékimòz ekimoz effet - éfè étè fermeture - fèrmötyr fèrmetyr fin - fê fè force - fors fòrs formidable - formidabl fòrmidabl garcon - garsô garsö gars - ga gà gaz - gaz gàz grace - gras gràs monsieur - mösjö mesjö niveau - nivò nivo obstine - opstiné òpstiné orchestre - orkèstr òrkèstr pate - pat pàt porte - port pòrt recevoir - rösvwar resevwar regard - rögar regar regarder - rögardé regardé sauve - sòvé sové second - sögô segô tenir - tönir tenir tombeau - tôbò tôbo tordu - tordy tòrdy vol - vol vòl yacht - jot jòt

absorbe - apsòrbé apsòbé lorsque - lòrsk lòrske promenade – pròmnad pròmenad revenir - revnir revenir	absorbe - apsorbé apsòbé lorsque - lorsk lòrske promenade - promnad pròmenad revenir - rövnir revenir
compte - kôt köt culture - kyltyr kyityr fin - fê fè garcon - garsô garsö independant - êdépâdâ ëdépâdâ longtemps - lôtâ lòtâ maintenant - mêtnâ mêtenâ rouquin - rukê rukë triomphe - trijôf trijöf vingt - vê vë	compte - kôt köt culture - kyltyr kyityr fin - fê fè garcon - garsô garsö independant - êdépâdâ ëdépâdâ longtemps - lôtâ lòtâ maintenant - mêtnâ mêtenâ rouquin - rukê rukë triomphe - trijôf trijöf vingt - vê vë

3.4.1 Training Data

While the OmniLex-Lexique3 data set was not used as the primary data set for Graphon, some tests were performed for comparison purposes. In these cases, the test data that was generated from elsewhere, i.e., reading assessments, was removed from the data set and the remainder was used for training.

3.5 Manulex

Manulex is a grade-level lexical database from French elementary school readers (Lété, Spreger-Charolles, & Colé, 2004). It contains frequency counts based on the grade level of the 1.9 million words found in French elementary school readers. As demonstrated by Cattell (1886), word frequency has been successfully used in models to simulate language development.

Training data was created by taking the Estimated Frequency of Usage per million words for the 1st-grade, 2nd grade, and 3rd-to-5th grade tables. Given that the purpose was to determine the exposure to graphemes, the wordform lexicon was used rather than the lemma lexicon. Hyphens and spaces were removed to create single words from hyphenated and multi-word expressions, so that they corresponded with the Omnilex data. The phonetic transcription of each word was taken from the Omnilex data; words found in the set of words that the students were tested on were removed, and finally, each word was replicated in the training data x number of times, where x is the frequency of the word, as provided by Manulex, to possibly some maximum.

A comparison of the word counts for Grades 1 to 3 can be found in Table 7. The total number of words a child is considered to have been exposed to by Grade 3 is roughly four times the number of words they were exposed to by Grade 1. However, the percentage of words that they have been exposed to a given number of times remains stable across grades. For example, less than 6% of all words have an exposure of greater than 50 times. The maximum count for words with a length greater than three occurs for the word “dans” in all cases, with a count of 1800 (Grade 1), 3591 (Grade 2) and 11910 (Grade 3).

Table 7: Summary of Manulex Word Counts across Grades 1 to 3

	Grade 1		Grade 2		Grade 3	
Count	**#**	**%**	**#**	**%**	**#**	**%**
1	4371	39.2	7282	38.7	15134	33.4
5	8142	73.0	13886	73.8	30818	68.0
10	9321	83.5	15796	84.0	35878	79.2
25	10375	93.0	17432	92.7	40610	89.6
50	10780	96.6	18084	96.2	42721	94.2
Total	11157		18808		45325	

Recognition improves as it becomes more automatic through frequent exposure, but there is also a limit beyond which the value of exposure has little effect (Cattell, 1886), (Laberge & Samuels, 1974). The limit was tested by varying the number of repetitions of a word from 50 to 250 and unlimited, and the lower limit by only including words which typically occurred more than some limit, 5 to 25. Words with a length of less than four were also not used, since they provided little useful data in terms of neighboring graphemes.

The Manulex data set was used for frequency counts only. These counts were applied to other data sets to provide training data. Once Graphon had been trained, it was tested using the same words the students had been tested on.

3.6 Brulex

The Brulex data set consists of orthographic, phonological, grammatical, and frequency information for roughly 36,000 French words. For use by the Graphon system, the Brulex coding (See Appendix A) needed to be converted to APF and then to ALINE Code. The Brulex data set was used for comparison with state-of-the-art systems. In order to simulate the 10-fold cross validation testing performed by other systems, the Brulex data set was randomly divided into ten sets, nine were used as training data and one was used for testing. This procedure was repeated 10 times and the reported accuracy is the average over the ten trials.

A comparison of the Brulex and Omnilex data sets shows that there are almost 17,000 words that have a different phonetic transcription. That is almost half of the Brulex data set. In comparison, there are roughly 8,000 words that differ in phonetic transcription between the Lexique3 and Brulex data sets. As mentioned earlier in section 3.4, this is likely due to the region of origin. Like Lexique3, Brulex was developed in France. While these differences are of interest, they are not relevant to the results in any of the scenarios in which the test data is taken from the training set. This is true in all cases with the exception of experiments run using the student test data. In that case, the phonetic transcriptions of the test words were generated by researchers from the University of Ottawa (Carson & Desrochers, 2012) and often differed from those in the various data sets.

Numerous research approaches have been tested on Brulex, using a 10-fold cross-validation; therefore, Brulex was used for comparison with other approaches.

3.7 Student Data

As part of their study of the effect of grapheme type, word length and word frequency on word reading accuracy, Carsons and Desrochers (2012) and their team conducted a reading assessment of children in Grades 1 to 6. See Appendix B for a sample data collection form. These forms were collected manually by the members of the team. For the purpose of this research, only data from Grades 1 to 3 were used, since the error rate in Grades 4 to 6 was so low as to not provide any useful learning. In addition, the importance of phonics on literacy skills is minimal beyond Grade 2 (Willows, 2008).

In the study, French children were asked to read 192 words (See Appendix C for the list of test words). The phonetic transcriptions of their responses were recorded if they differed from the correct transcription. The incorrect phonetic transcriptions were entered manually, so while there were roughly 130 students per grade, only the data from 40 students per grade was used, since this was an intensive process. Given that the purpose of this research is to show proof of concept, it was determined that the 40 students provided sufficient data to show a correlation between the system and the students.

3.8 Student Test Words

The test data used for comparison with the student results was the 192 word list used in the study along with the expected transcriptions used when the students were being evaluated.

4 Graphon

4.1 Alignment

The attributes used for assigning a phoneme to a particular grapheme are the neighbouring letters. In order to train the system, it is necessary to determine which grapheme corresponds to which phoneme.

In some cases, a phoneme can be represented by multiple graphemes, and one grapheme can represent multiple phonemes, so in order to learn from the training set, the graphemes must be aligned with the corresponding phonemes. This can present a number of problems, since the word and the phonetic transcription often differ in length. There are three possible reasons for this:

1. a silent letter (i.e., e in home)
2. a letter that results in two sounds (i.e., x in fax -> k, s)
3. two or more letters that result in one sound (i.e., Au in restaurant -> oL)

In these cases, hyphens (-) are used to align each letter to its corresponding phoneme. As this is a complete area of research in itself, it was decided to use an existing algorithm, ALINE, developed by Greg Kondrak (2000). ALINE, aligns words from different languages to calculate similarities and determines possible inheritances between languages. ALINE uses phonetic features underlying individual words to calculate

distance matrices for those languages. This project is using ALINE to align words and their phonetic transcriptions; therefore, the similarity score is not used.

```
b r a  v e
b r aF v -
Similarity score: 135.5

s c i e  n t i f i q u e
- s y aBN - t i f i k - -
Similarity score: 214.5

e f f a  c e
e - f aF s e
Similarity score: 151.5

r e e  l l e m e  n t
r e eL - l - m aBN - -
Similarity score: 166.5

e  m b e l l i r
aBN - b e - l i r
Similarity score: 166.5
```

Figure 1: Sample Alignments Produced by Aline

The original Graphon used Kondrak's Aline algorithm, which was developed in C++. This did not integrate well with the rest of the project developed in Java. As well, Aline could only handle 1000 word pairs at a time. This required breaking up the training and test data into individual files. Given that the data set size had increased by 10 fold, I opted to use PyAline, from Huff (2010), a Python implementation of Aline. Since

PyAline was developed primarily for calculating the similarity score, it did not always provide the complete alignment, chopping off starting and ending characters in certain cases. This required some debugging of the algorithm to address the issue. I will make the corrected version available.

Since neither ALINE nor PyAline used the Alphabet Phonetic Français (APF) used by Omnilex or Manulex for phonetic transcription, a conversion needed to be performed from APF to the Aline codes. See Appendix A for the conversion tables.

4.2 Classification

Once the training data is aligned, it needs to be examined to identify patterns (rules) based on a grapheme's neighbours (context). These patterns will classify the grapheme as corresponding to a particular phoneme. In Machine Learning, this process is known as classification. There are numerous tools to use to perform this process. For this project, WEKA (Hall, Frank, Holmes, Pfahringer, Reutemann, & Witten, 2009) was chosen for the Java API, the ease in implementing weighted engineering, and its familiarity.

In addition, various algorithms can be used for classification. Fernandez-Delgado *et al*. (2014) compare 179 classifiers from 17 families. Their results show that the family of classifiers with the best results is Random Forest (3 out of top 5), with Support Vector Machine (SVM) being second (4 out of top 10). Actual results depend on particular implementations and tuning. Another consideration is the time it takes to run a classifier.

Given the hundreds of tests that were performed, Random Forest is much more practical than SMO, since it was roughly 40 times faster to execute.

Graphon was tested on the following classifiers:

- Naïve Bayes for its implementation of weighted attributes
- J48 for comparison with original implementation of Graphon
- Random Forest – top overall classifier
- SMO – Weka's implementation of SVM – 2nd overall classifier

Naïve Bayes is one of the simplest types of classifiers and is typically used as a baseline method for comparison of classifiers. According to Fernandez-Delgado *et al*. (2014), of their comparison of 179 classifiers, not one Naïve Bayes classifier ranked in the top 50%. A Naïve Bayes classifier assumes that the value of a feature is independent of the value of all other features. As will be discussed in section 5.2, this assumption does not hold well for grapheme-to-phoneme conversion. Regardless, Naïve Bayes was used in this project as a baseline and because it is the only classifier for which Weka supports weighted attributes.

J48 is a Java implementation of the C4.5 algorithm used by Weka. Given a set of training data, the C4.5 algorithm, developed by Quinlan (1993), generates a decision tree by choosing a test based on a single attribute that will classify the training data into two or more classes (Wu, et al., 2008). This is repeated for each subset until all of the cases belong to the same class or some other criteria is met, such as the maximum depth of the tree. The tree is then pruned to avoid overfitting. Overfitting happens when the tree

classifies the training data almost perfectly, but introduces errors when applied to other data.

A set of rules can then be generated from the decision tree. Pontes and Furui (2010) used C4.5 decision trees to model liaisons in French for just this reason. They generated rules for a G2P converter, and then augmented these rules with some that applied to the words themselves, such as parts-of-speech labels. Contrary to the typical use of C4.5 decision trees, they did not prune their trees because they did not want to lose any information. They were not concerned with overfitting since the output of G2P converter was applied to other rules.

Random Forest involves generating multiple trees trained on a random sample of the training set, and then averaging the predictions from each of the trees to produce a final result. Overfitting is no longer a problem, because the trees are trained on different data sets. While the best results were achieved using an R implementation of Random Forest, of the Weka classifiers tested in their study, Fernandez-Delgado *et al*. (2014) found Random Forest and SVM classifiers to be the best. Performance of a classifier is dependent on a number of settings. Fernandez-Delgado *et al*. (2014) set the number of trees to 500 and unlimited depth. For this project, the default settings were used: 100 trees and unlimited depth.

Support Vector Machine (SVM) models the training data as points in space. The points are classified by finding a hyperplane that has the largest distance to the nearest data point. For a multiclass problem such as grapheme-to-phoneme conversion, the problem must be reduced into multiple binary classification problems. This is usually done by

defining a binary classifier as distinguishing between one label and all other labels. Dou *et al.* (2009) trained an SVM ranker to rank stress patterns. They achieved 96.2% word accuracy on prediction of stress patterns in English.

4.3 Evaluation

Two measures of evaluation are used throughout this project. Word Accuracy compares the number of correctly transcribed words to the total number of words. This measure is used to compare the results of different data sets, different algorithms, as well as state-of-the-art systems. Phoneme Accuracy compares the number of correctly classified phonemes based on the alignment produced by PyAline with the total number of phonemes. While phoneme accuracy is used throughout research on grapheme-to-phoneme conversion, it is dependent on the alignment algorithm; therefore, it is only used here to compare results within this project.

4.4 Grapheme-to-Phoneme Conversion

The Graphon project is a grapheme-to-phoneme converter. The original system was developed by Martin and Inkpen (2006) and was tested using a subset of the Omnilex database. The input to the system is a list of words and their corresponding phonetic transcription. The graphemes and phonemes are aligned using ALINE (Kondrak, 2000). The pairs of graphemes and phonemes are then used as input to a machine learning

system where it is determined, based on 1-9 preceding and following characters, which phoneme should be associated with which grapheme.

The Graphon system has been designed such that it can be used to evaluate pre-existing test data (Alternative 1), randomly created test data (Alternative 2), and even results generated otherwise than by Graphon (Alternative 3). A configuration file allows a user to specify the different input files, as well as the number of attributes and the classifier to use (see Figure 2).

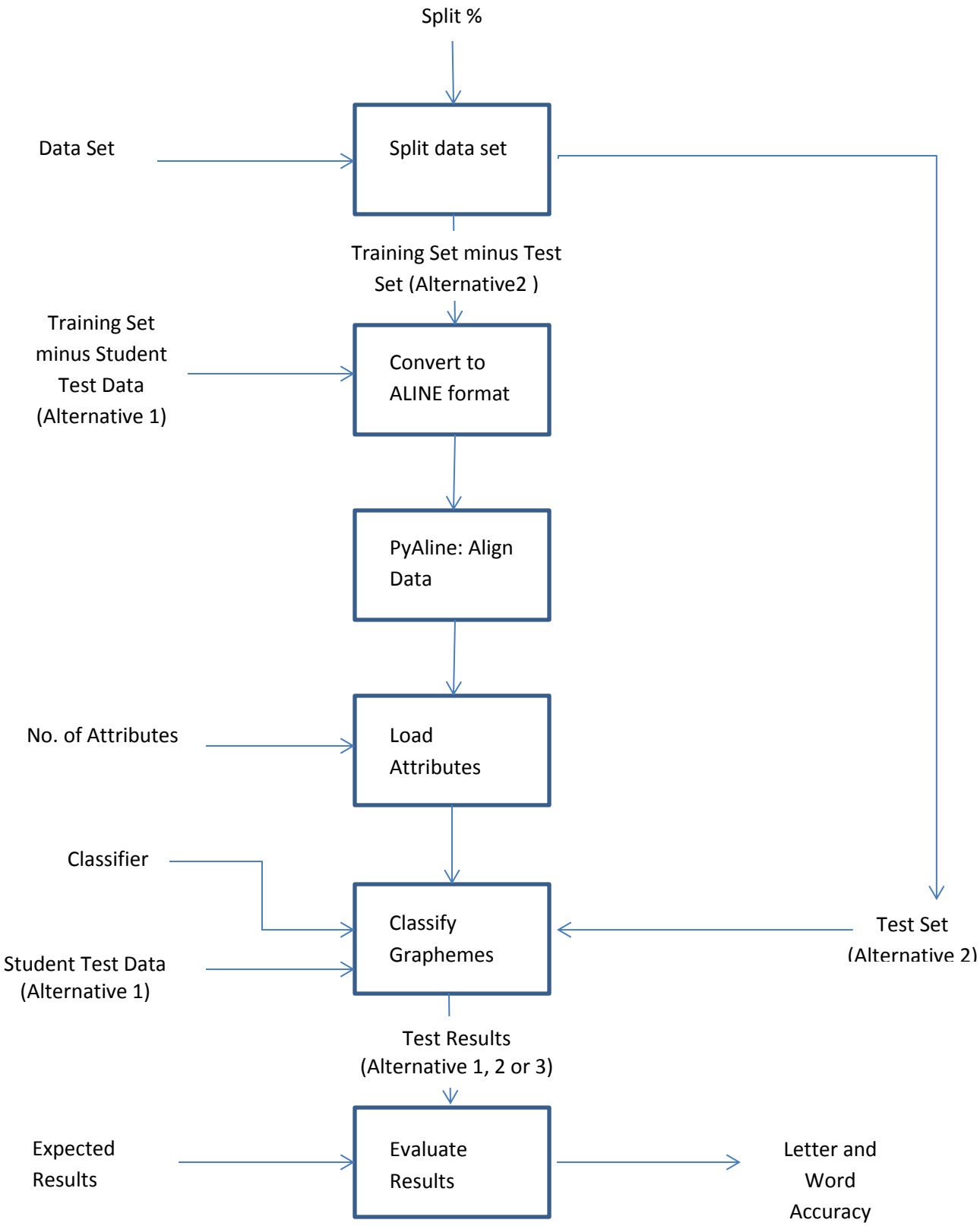

Figure 2: Graphon System Architecture

5 Analysis of Weights

As shown in Table 11, word accuracy increases as the number of neighbouring letters used as attributes increases, up until six attributes, and then as additional letters are used for classification, the accuracy starts falling. My theory is that the letters closer to the grapheme being classified should be given more weight during classification because they provide a stronger context for determining the resulting phoneme.

5.1 Weighted Attributes

Weka 3.7.2 supports a new input file format, .xrff that allows for the definition of weights for each attribute. For example:

```
<attribute name="petalwidth" type="numeric">
   <metadata>
     <property name="weight">0.9</property>
   </metadata>
</attribute>
```

The additional overhead in the format increased the file size from 20 to 50 times the size of a corresponding .arff file. (See Appendix D – Sample .arff File and Appendix E – Sample .xrff File). This may be problematic with large data sets. Testing was performed using two, three and four neighbours (four, six and eight attributes, respectively) with the attribute weighting being divided by two, three or four as the neighbour distance increased. As shown in Table 8, weighting of attributes provides a significant

improvement, 9.1% and 15.1% respectively between the baseline and the best results for six and eight attributes. In the case of four attributes, the improvement is only 0.7%. In the case of four and six attributes, the results show that halving the weight of each subsequent neighbour achieves the best word accuracy.

Table 8: Comparison of Weighted Attributes (Naïve Bayes)

(Omnilex 10-fold cross-validation)

# of Attributes (neighbours)	Weight Factor of Farther Neighbours	Phoneme Accuracy	Word Accuracy
4	1	92.7	64.9
4	0.5	92.8	**65.6**
4	0.33	92.5	64.6
4	0.25	92.2	63.2
6	1	90.2	56.1
6	0.5	92.7	**65.2**
6	0.33	92.6	65.1
6	0.25	92.2	63.5
8	1	88.9	49.9
8	0.5	92.6	64.6
8	0.33	92.6	**65.0**
8	0.25	92.2	63.4

5.2 Weighted Engineering

Although attribute weights can now be defined in Weka, they are only currently implemented for Naïve Bayes. Since Naïve Bayes had the poorest performance amongst all the classifiers (see Table 12), it was decided that another approach was required.

Matwin *et al*. (2010) introduced the concept of weight engineering while classifying medical papers, in an attempt to reduce the workload of experts in systematic reviews. They theorized that a number of binary features, such as publication type, should be given a higher weighting than nominal features, such as the count of individual words in an abstract. Rather than implement support for attribute weights in a classification algorithm, they preprocessed the attributes to simulate weights. They applied a weight multiplier to the binary features to give them more importance than the nominal attributes.

In this case, attributes were weighted by creating duplicate instances for the nearest attributes, to increase their frequencies, and thereby their weight.

To train a classifier, using 4 attributes, Graphon generates an .arff file for each possible grapheme. The .arff file contains an instance for each occurrence of a grapheme in the training data. Each instance details the attributes (neighbours) of the grapheme, and the corresponding phoneme. For the word “asplenium”, aligned as follows:

```
a s p l e n i u m
a s p l é n j ò m
```

Graphon creates an instance (letter+1 letter-1 letter+2 letter-2 class) for each grapheme in its respective .arff file. For the grapheme “e”, with four attributes, the following instance is created:

```
a s p l é n j ò m  →     n l j p é
```

To replicate a weighting where the closest neighbour has twice the weight of the next neighbour, I define the weighting as 2211. This weighting requires two instances be created instead of just one, where the second instance repeats the closest neighbour, but leaves the next neighbour blank:

```
a s p l é n j ò m  →     n l j p é
                         n l - - é
```

For the grapheme “e” with eight attributes, and a weighting of 87654321, where not only are the weights decreased as the neighbour gets further away, but a higher weight is given to the neighbour on the right than on the left, eight instances are created (letter+1 letter-1 letter+2 letter-2 letter+3 letter-3 letter+4 letter-4 class):

```
aF s p l e n j oL m →    n l j p oL s m aF e
                         n l j p oL s m –  e
                         n l j p oL s - -  e
                         n l j p oL - - -  e
                         n l j p -  - - -  e
                         n l j - -  - - -  e
                         n l - - -  - - -  e
                         n - - - -  - - -  e
```

To prove the validity of this approach, the experiment was repeated using the Naïve Bayes classifier and similar weightings, using weight engineering. First, the attributes were merely replicated to ensure that increasing the number of instances alone did not

affect the results. As shown in Table 9, the results do not significantly differ between a weighting of "1111" and "4444". Then, the weight factors were reproduced using the weighted engineering algorithm. This allowed for a comparison between weighted attributes and weight engineering. Table 9 shows almost identical results were achieved with a weight factor of 0.5 and a weighting of 4422 for both the four and six attribute scenarios. However, there is a difference of 3.7% in word accuracy with a weight factor of 0.3 and a weighting of 993311 for six attributes. It appears that, as long as the gap between the counts is relatively small, weight engineering can replicate the results of weighted attributes. However, as the gap increases this is no longer the case.

Table 9: Comparison of Weight Engineering (Naïve Bayes)

(Omnilex 10-fold cross-validation)

No. of Attributes	Weight Engineering			Weighted Attributes		
	Weighting	Word Accuracy	Phoneme Accuracy	Weight Factor	Word Accuracy	Phoneme Accuracy
4	1111	64.9	92.7	1	64.9	92.7
4	4444	64.6	92.7	-	-	-
4	4422	**65.6**	92.9	0.5	65.6	92.8
6	111111	56.5	90.8	1	56.5	90.8
6	666666	56.0	90.6	-	-	-
6	442211	61.4	91.9	0.5	61.4	91.9
6	993311	61.3	91.8	0.3	65.0	92.6

Given that the best results on the Omnilex data set were achieved with the Random Forest classifier and 8 attributes, I decided to use it to experiment with weighted engineering to see if these results could be improved. The results are shown in Table 10.

Table 10: Comparison of Weight Engineering (Random Forest)

(Omnilex 10-fold cross-validation)

No. of Attributes	Weighting	Word Accuracy	Phoneme Accuracy
8	11111111	90.1	98.0
8	22222222	90.0	98.0
8	22221111	89.3	97.9
8	21212121	89.2	97.9
8	12121212	79.0	96.0
8	44444444	90.2	98.1
8	44443322	88.7	97.8
8	44332211	88.6	97.8
8	88442211	88.8	97.8
8	87654321	89.0	97.8
8	88884422	88.8	97.8
8	88888888	90.3	98.1

While weighted attributes and weight engineering showed considerable improvements for the Naïve Bayes classifier (Table 9), weight engineering resulted in poorer performance for the Random Forest classifier (Table 10).

Naïve Bayes assumes that each feature is independent, so increasing or decreasing the weight of each feature does not have an effect on the weight of other features. In contrast, features are not independent in a Random Forest classifier, which is more the case in reality. The classification of a phoneme is dependent on a sequence of letters, not just the location. For example:

e x c e p t i o n n e l
eL k s eL p s y o - n eL l

d e c o r a t i v e
d e k oL r aF t i v –

g a s t r o t o m i e
g aF s t r o t oL m i –

The ‘t’ in ‘exceptionnel’ takes on the ‘s’ sound because it is followed by an ‘i’ and then an ‘o’. As can be seen in ‘decorative’, the following ‘i’ alone does not correctly classify the ‘t’, nor does an ‘o’ at neighbour + 1. Unlike attribute weighting, weighted engineering breaks up the attributes; therefore, it only works for Naïve Bayes.

6 Graphon Results

In Raphaelle Martin's initial development of the Graphon project, she implemented procedures to convert phonemes from the APF format used by Omnilex, and the format used by Aline (Kondrak, 2000) the alignment tool used. See Appendix A for the mappings and examples. Martin achieved an accuracy rate of 56% for word conversion using only the preceding and following letters and a J48 (Decision Tree in Weka) classifier. This rate fell to 17% when the nine preceding and nine following letters were used. When accuracy was measured on a phoneme basis, a rate of 91% was achieved for two attributes, and 72% for eighteen attributes.

However, there was a programming error in how the test data was generated, causing the preceding neighbours to be overwritten by the following neighbours. This resulted in low word accuracy for greater than two attributes. Once this was fixed, word accuracy jumped to 84.8% for four attributes, which makes more sense given the phoneme accuracy (see Table 11). Given the inaccurate calculation of word accuracy in the original Graphon project, comparison should only be done on phoneme accuracy. Increasing the training data improved phoneme accuracy from 91% to 96.9% for J48 and four attributes. Increasing the number of neighbours to six improved the accuracy to 97.7%.

Table 11: Graphon – Original vs. Improved

(Comparison of Number of Neighbours using the J48 Classifier)

No. of Attributes	Original		Corrected	
	Phoneme Accuracy	Word Accuracy	Phoneme Accuracy	Word Accuracy
2	--	--	91.5	59.7
4	91.0	56.6	96.9	84.8
6	--	--	97.7	**88.4**
8	--	--	97.6	88.2
10	--	--	97.6	88.1
18	71.8	17.0	97.6	88.0

A comparison of classifiers and number of attributes shows that Random Forest Classifier with eight attributes had the best results (see Table 12). Naïve Bayes has by far the worst results. As discussed earlier, this is likely a result of the assumption that each attribute is independent, which is not the case in G2P conversion. While the other classifiers are relatively close, each performs best on a different number of attributes. It is not surprising that Random Forest outperforms the other classifiers. Fernandez-Delgado *et al*. (2014) compared 179 classifiers using 121 data sets and found the top two classifiers to be different implementation of Random Forest. Therefore, while only four classifiers were evaluated in this project, it was deemed that the use of Random Forest with eight attributes would be the best option for comparison with the student data.

Table 12: Comparison of Classifiers

(Omnilex 10-fold cross-validation)

Classifier	**NB**		**J48**		**Random Forest**		**SMO**	
Accuracy	**Phoneme**	**Word**	**Phoneme**	**Word**	**Phoneme**	**Word**	**Phoneme**	**Word**
2 Attr	90.4	55.9	91.5	59.7	91.5	59.7	91.3	59.2
4 Attr	92.7	**64.9**	96.7	84.8	97.0	85.0	96.2	81.9
6 Attr	90.8	56.5	97.7	**88.4**	97.8	88.9	97.0	85.5
8 Attr	88.9	49.9	97.6	88.2	98.0	**90.1**	97.2	86.1
10 Attr	87.6	46.4	97.6	88.1	98.0	90.0	97.3	**86.3**

6.1 Graphon vs Other Algorithms

Graphon was run on the Brulex data set using 10-fold cross-validation, using Weka's Random Forest classifier with up to eight neighbours as attributes. The results were compared with an implementation of a TiMBL system (Daelmans & Van Den Bosch, 1997) using ALINE for alignment (Jiampojamarn & Kondrak, 2010), the joint processing and online discriminative system aligned using ALINE and an EM aggregated alignment method (Jiampojamarn & Kondrak, 2010), a system using many-to-many aligner with HMM (Jiampojomarn, Kondrak, & Sherif, 2007), and Demberg *et al*. (2007) joint n-Gram approach. With the exception of the joint processing and online discriminative training algorithm, Graphon's results are comparable to the state-of-the-art (see Table 13).

Table 13: Comparison of State-of-the-Art Algorithms

Method	Alignment	Data Set	Word Accuracy
TiMBL	Aline	Brulex	89.4
Discriminative	Aline	Brulex	94.6
Discriminative	EM-Aggr	Brulex	95.1
M:M+HMM	M:M	Brulex	90.9
Joint n-Gram		Brulex	89.1
Graphon RF 8	Aline	Brulex	90.0

Surpassing the state-of-the-art is not the goal of this project and while the word accuracy of 90% achieved on the Brulex system is as good as most systems, the accuracy achieved for the student test words using the Omnilex training data is only 80%. While this should be sufficient for working with the student data from Grade 1 to 3, where the accuracy ranges from 64 to 84.2%, it would be a problem if the study were extended to Grade 6, where the word accuracy rises to 92.8%. Again, it is worth noting that the best results are obtained using the training sets that originate completely or partially in the same region as the wordlist. As can be seen in Table 14, the best results for the student word list are achieved using the Omnilex training set, both of which originated in Canada, as opposed to the other training sets that originated completely or partially from France.

Table 14: Comparison of Results on Different Training Sets

(Graphon Random Forest Classifier using 8 attributes)

Data Set	Phoneme Accuracy	Word Accuracy
Omnilex	94.3	80.0
Lexique3	87.9	57.3
Omnilex-Lexique3	90.6	64.6
Brulex	86.3	52.6

7 Simulation Results

7.1 Graphon

The Manulex training set was modified to vary the word frequency from a minimum of 1 to 25 and a maximum of 25 to 150. In other words, if the frequency of a word is less than the minimum, it is deemed as too rare to make an impact on the learning of a child's phonetic rules, and is therefore not included in the training data. If the frequency of a word is greater than the maximum, it is believed that the learning to be obtained from this word had reached a limit and therefore no further occurrences are necessary.

As shown in

Table 15, the best results are achieved whenever an example is provided at least once, giving more examples to generate the rules. However, repeating examples of the same word appears to hit a maximum around 150 occurrences, with better results occurring for Grade 1 and 2 with no minimum and a maximum of 150. In comparison, with a larger dataset, such as in Grade 3, the maximum appears to be much lower, around five occurrences.

Table 15: Comparison of Omnilex and Student Data

(Graphon Random Forest Classifier using 8 attributes)

Frequency		Grade 1		Grade 2		Grade 3	
Min	**Max**	**Phoneme**	**Word**	**Phoneme**	**Word**	**Phoneme**	**Word**
1	**5**	92.6	74.0	93.4	78.1	**94.8**	**81.8**
5	**5**	86.9	52.1	88.7	57.3	90.4	66.1
1	**10**	92.8	74.5	93.4	78.0	95.2	80.6
5	**10**	87.5	54.2	88.2	57.6	90.5	66.0
10	**10**	86.0	51.0	87.6	55.2	89.6	62.5
1	**25**	91.9	71.2	93.9	79.6	95.0	81.3
5	**25**	86.5	52.6	87.8	57.6	90.7	65.6
10	**25**	85.8	50.5	86.9	53.1	89.6	62.0
25	**25**	80.3	39.3	81.6	40.1	87.7	57.8
1	**50**	92.2	72.9	93.8	79.1	94.4	79.6
5	**50**	86.6	51.6	87.3	54.5	90.5	66.5
10	**50**	85.5	51.0	86.8	53.6	89.8	64.6
25	**50**	79.5	36.3	82.8	44.8	87.4	56.0
50	**50**	69.2	21.1	77.3	31.6	85.7	53.1
1	**100**	91.8	70.1	93.7	78.6	94.8	81.3
5	**100**	86.3	53.6	87.9	57.6	90.2	63.4
10	**100**	85.6	48.4	86.1	54.2	89.6	63.0
25	**100**	79.5	37.1	81.5	44.3	87.8	57.1
50	**100**	69.9	20.0	77.6	35.1	86.6	52.4
1	**150**	**92.9**	**74.5**	**94.1**	**79.7**	94.9	79.6
5	**150**	87.7	56.5	88.3	57.6	90.5	65.6
10	**150**	84.3	47.4	86.6	55.2	89.4	62.3
25	**150**	79.6	34.6	82.2	43.2	82.3	55.5
50	**150**	66.8	18.9	80.0	40.5	86.1	51.6
1	**200**	92.6	72.4	93.6	79.1	94.1	79.7
1	**250**	92.3	71.4	93.3	77.5	94.9	79.7
1	**unlimited**	92.3	71.2	93.6	77.5	94.1	77.1
Students		83.8	64.3	92.5	82.4	93.7	84.8

As shown in Figure 3, Figure 4 and Figure 5, the effect of the maximum word frequency on word accuracy is relatively small. Each line is almost horizontal and does not vary per grade. In contrast, the effect of the minimum is as much as 50% in Grade 1, but gets progressively smaller in the older grades. Unlike children, Graphon benefits from a single exposure to a word to expand its rules; however, it does not memorize words, so increased exposure does not provide additional benefit. The effect of repeated exposure to common words is simulated by giving more weight to common grapheme-phoneme pairs.

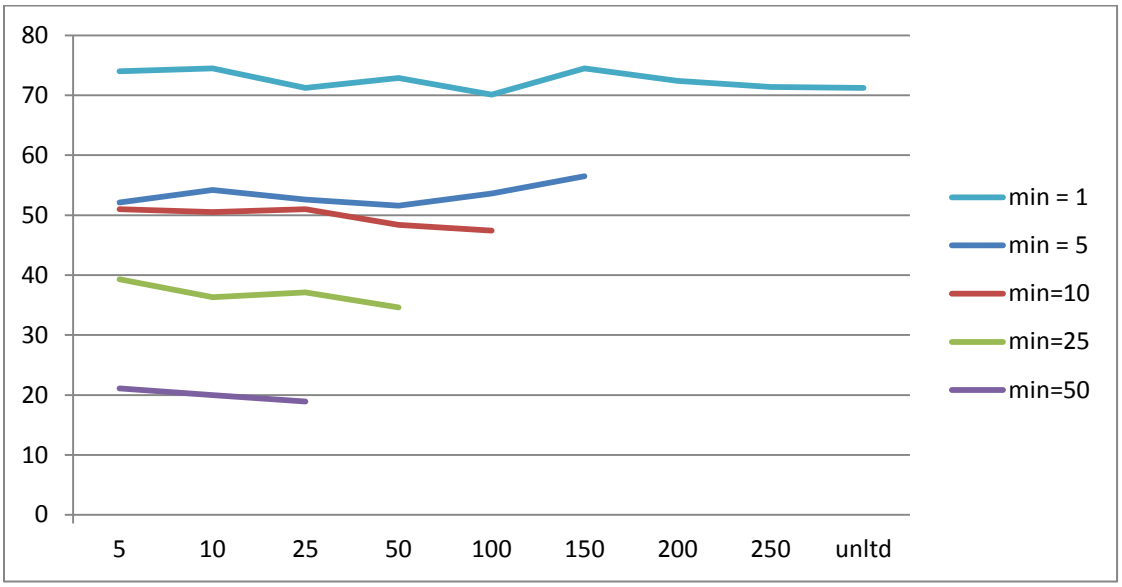

Figure 3: Word Frequency vs Word Accuracy – Grade 1

Word Accuracy is on the vertical axis, maximum word count is represented on the horizontal axis, and each line represents a minimum word count.

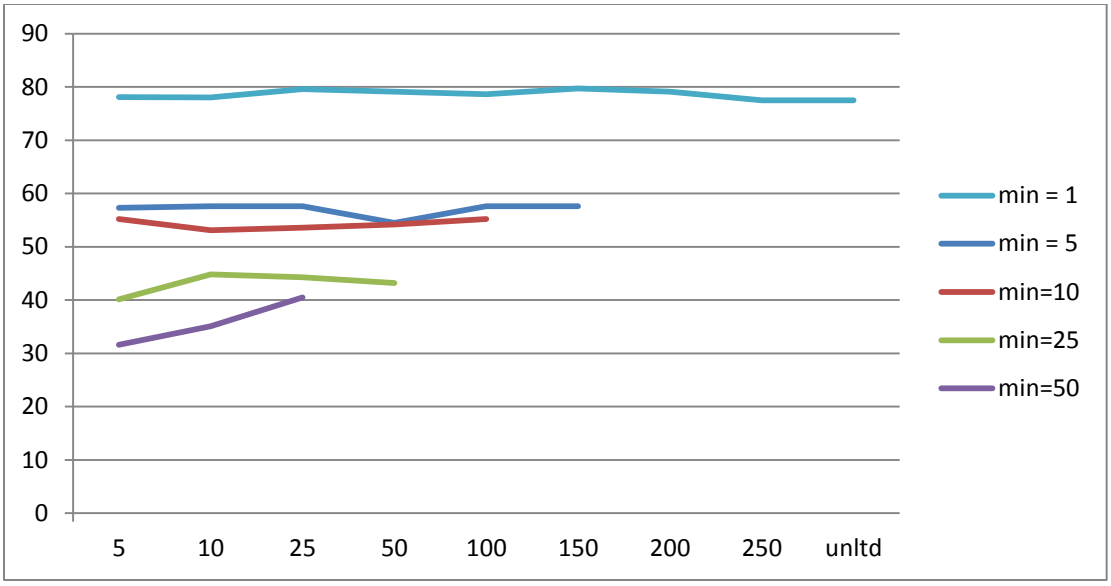

Figure 4: Word Frequency vs Word Accuracy – Grade 2

Word Accuracy is on the vertical axis, maximum word count is represented on the horizontal axis, and each line represents a minimum word count.

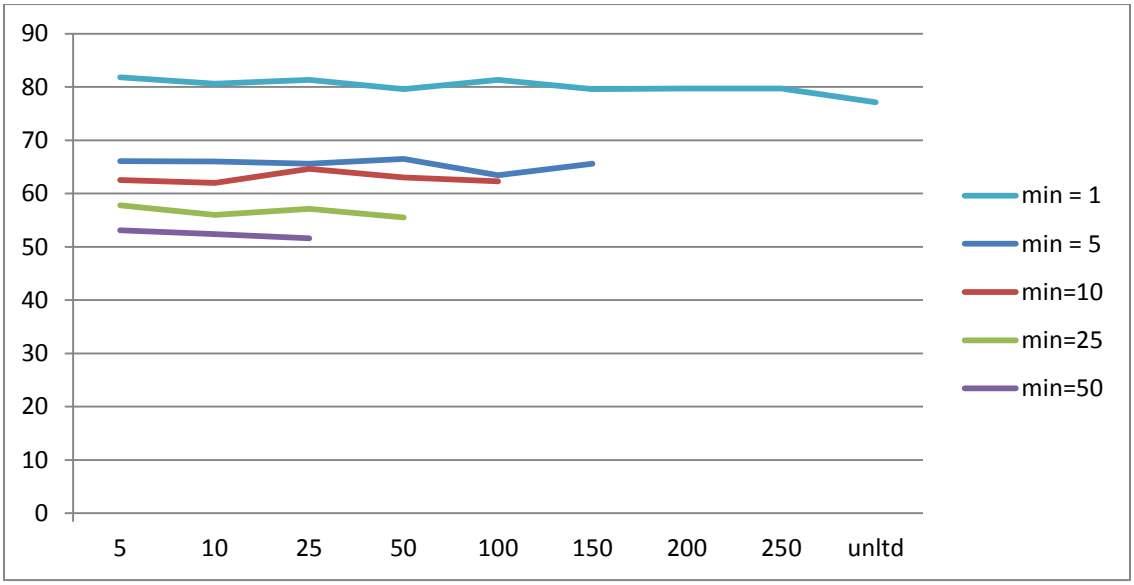

Figure 5: Word Frequency vs Word Accuracy – Grade 3

Word Accuracy is on the vertical axis, maximum word count is represented on the horizontal axis, and each line represents a minimum word count.

7.2 Graphon vs Students

So, how does the Graphon system compare to the students on a phoneme basis? By comparing the accuracy for each phoneme as generated by Graphon and the students, and calculating the Pearson correlation coefficient (see Table 16), we see that the correlations are always positive, implying that overall, Graphon and the students perform better on the same phonemes. However, the best correlation is not necessarily when the best results are achieved. This is confirmed by looking at the correlation with the Omnilex-Lexique3 training set (see Table 17), which achieved poorer results overall. It can be seen that it is even better correlated to the student data. This may reflect the fact that students may originate from different regions. The complete set of phoneme accuracies for Omnilex is reported in Appendix F – Phoneme Accuracies.

Table 16: Correlation of Student Data and Graphon trained using Omnilex Dataset

Frequency		Omnilex		
Min	**Max**	**Gr. 1**	**Gr. 2**	**Gr. 3**
1	**5**	0.398	0.589	0.320
5	**5**	0.289	0.333	0.266
1	**10**	0.482	**0.640**	0.245
5	**10**	0.333	0.370	0.215
10	**10**	0.447	0.232	0.397
1	**25**	0.453	0.446	0.059
5	**25**	0.305	0.353	0.195
10	**25**	0.462	0.254	0.393
25	**25**	0.274	0.446	0.231
1	**50**	0.436	0.324	0.187
5	**50**	0.368	0.316	0.174
10	**50**	0.420	0.319	0.395
25	**50**	0.195	0.568	0.240
50	**50**	0.245	0.465	0.402
1	**100**	**0.578**	0.398	0.336
5	**100**	0.376	0.405	0.082
10	**100**	0.375	0.274	0.352
25	**100**	0.185	0.600	0.270
50	**100**	0.216	0.476	0.305
1	**150**	0.504	0.174	0.261
5	**150**	0.301	0.393	0.163
10	**150**	0.430	0.275	0.357
25	**150**	0.227	0.462	0.292
50	**150**	0.457	0.381	0.311
1	**200**	0.494	0.426	0.418
1	**250**	0.455	0.264	0.255
1	**unlimited**	0.454	0.473	**0.487**

Table 17: Correlation of Student Data and Graphon trained on Omnilex-Lexique3 Dataset

Frequency		**Omnilex-Lexique3**		
Min	**Max**	**Gr. 1**	**Gr. 2**	**Gr. 3**
1	**25**	0.713	0.328	0.400
1	**50**	**0.713**	0.328	**0.721**
1	**100**	0.653	**0.485**	0.643
1	**150**	0.678	0.395	0.704

By graphing the phoneme accuracies for Graphon (using the Omnilex dataset with the Manulex frequencies) and the students (see Figure 6, Figure 7, and Figure 8), it becomes obvious that with the exception of a few outliers, there is a strong correlation. If we could improve the accuracies of those Graphon outliers that occur in each grade, and decrease the Graphon accuracy of the student outliers in Grade 1, the correlation would be improved. Decreasing the accuracy of Graphon by reducing the frequency of those phonemes should not be difficult. Increasing the Graphon accuracies in general would likely require modifying the machine learning algorithm; however, slight improvements might be possible by increasing the frequency of those phonemes in the training data.

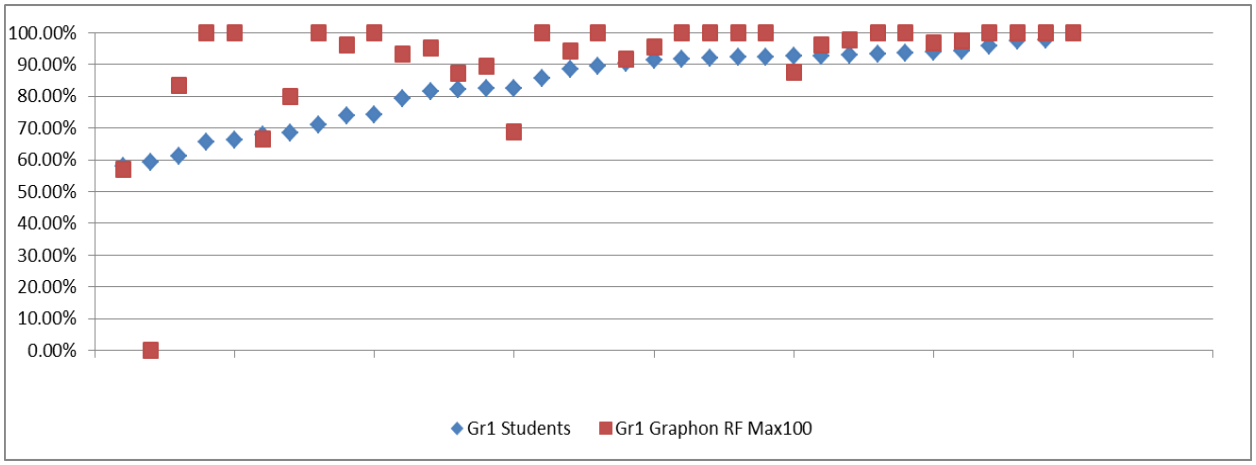

Figure 6: Comparison of Grade 1 Phoneme Accuracy

Comparison of phoneme accuracy for students vs. Graphon trained on Omnilex data set using frequency counts for Grade 1. Phonemes are represented by the horizontal axis, presented in increasing order of accuracy as represented by the vertical axis.

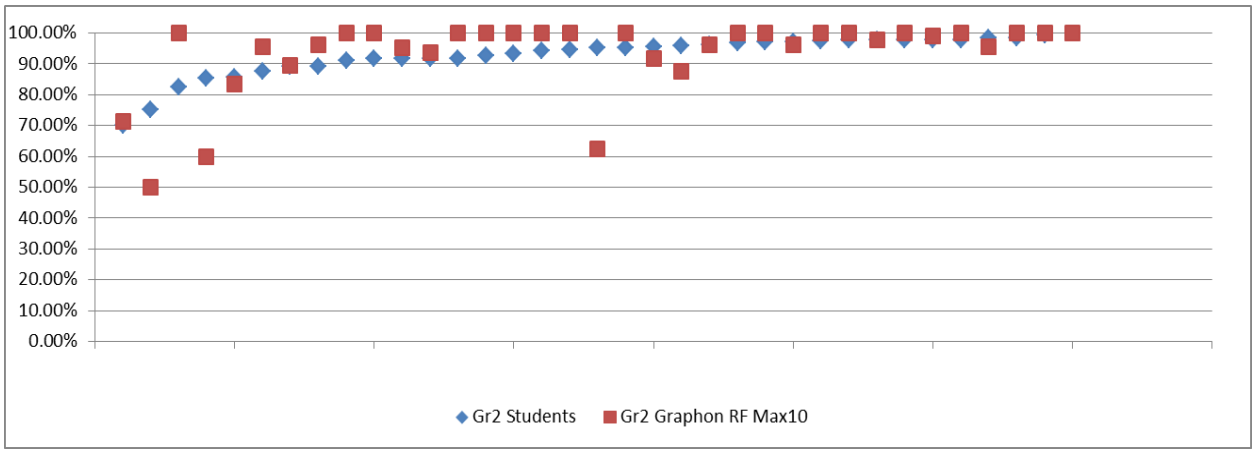

Figure 7: Comparison of Grade 2 Phoneme Accuracy

Comparison of phoneme accuracy for students vs. Graphon trained on Omnilex data set using frequency counts for Grade 2. Phonemes are represented by the horizontal axis, presented in increasing order of accuracy as represented by the vertical axis.

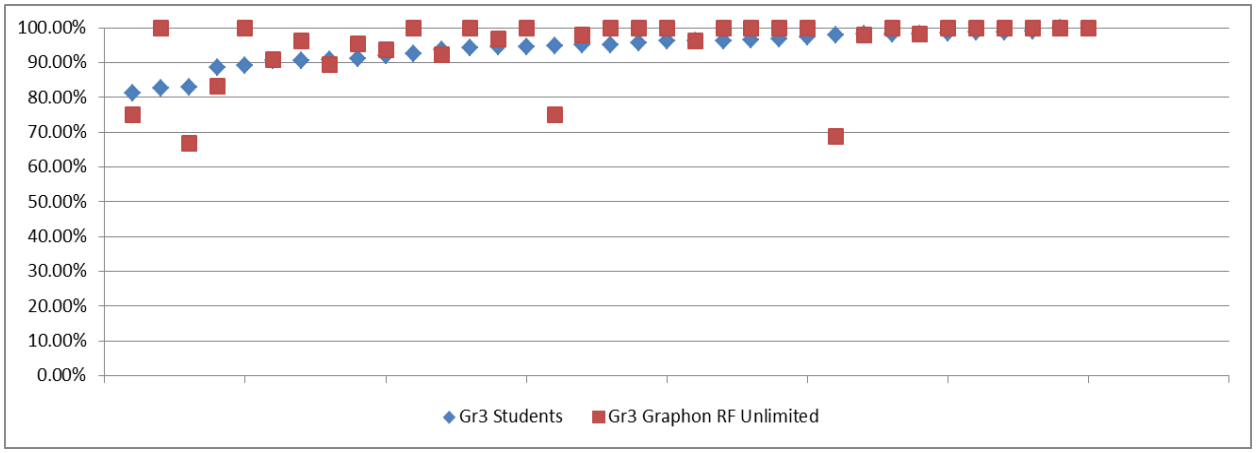

Figure 8: Comparison of Grade 3 Phoneme Accuracy

Comparison of phoneme accuracy for students vs. Graphon trained on Manulex data set using frequency counts for Grade 3. Phonemes are represented by the horizontal axis, presented in increasing order of accuracy as represented by the vertical axis.

7.3 Graphon Outliers

The phonemes with a difference in accuracy between the student results and Graphon of greater than +/- 20% are shown in Table 18. A positive difference represents a student outlier, whereas a negative difference represents a Graphon outlier.

Table 18: Graphon and Student Outliers

(Graphon Random Forest Classifier using 8 attributes and Omnilex Training Data)

Phoneme	Grade 1	Grade 2	Grade 3
à	-59.26	-25.00	
N	34.26		
j	33.67		
Z	28.92		
è	26.69		
ò	22.43		
ë	22.22		
e		-32.59	-29.12
ö		-25.23	

As can be seen in Appendix F – Phoneme Accuracies (Omnilex), the greatest student outlier is the phoneme [N], which is usually mapped to the grapheme “gn” found in test words “signe”, “ligne”, “vigne” and “digne”. Graphon achieves an accuracy of 100% for this mapping, whereas the Grade 1 students only average 65.7%. The difficulty is that in training data for Grade 1, the grapheme “gn” is only mapped to [N], which does not leave any room for error. Leaving even one instance of “gn” in the training data will result in an accuracy of 100%. The only way to improve the correlation with the student results is to introduce erroneous transcriptions similar to those provided by the students, for example, by mapping “gn” to [g] or [n].

Modifying the training data so that roughly 20% of the instances of “gn” map to “g”, and 20% to “n”, results in an accuracy of 75% for the grapheme “gn”. This correlates better with the student accuracy of 65.7% than the previous Graphon accuracy of 100%.

The largest Graphon outlier is the phoneme [à], where Graphon failed on 4/4 words (*gaz, gars, pâte, grâce),* whereas the Grade 1 students achieved an accuracy of 59.26%. In two of the four cases, the [à] mapped to [a], and in the other two [à] mapped to [-]. Overall improvement to the accuracy of Graphon is necessary to address the Graphon outliers.

7.4 Errors in Omnilex

One of the goals of this project was to evaluate the manual transcriptions defined in the Omnilex dataset. According to Martin (2004) and Elovitz (1976), this is a function of the region for which it has been developed. At least, the discrepancies found between Omnilex and the student test words should be revisited. However, I would also suggest that the 20,000 differences found between Omnilex and Lexique3, as well as the 17,000 differences found between Omnilex and Brulex should be revisited by native Canadian speakers.

7.5 Degree of Difficulty

One of the goals of this project was to propose an approach for assigning a degree of difficulty to a word. The degree of difficulty is useful for reading assessment of primary school children. It can be applied to individual words or to complete texts.

For books, there are multiple scales used within the education community. The Reading Recovery Scale[2] organizes English and Spanish books along a continuum of 20 levels for early primary grade children. Fountas and Pinnell[3] have classified close to 50,000 elementary school texts into A to Z+ levels. Ma *et al*. (2012) achieve an accuracy of 83% when ranking a small corpus of children's books when using visual as well as text-based features.

EVALEC (Sprenger-Charolles, Colé, Béchaennec, & Kipffer-Piquard, 2005) is one of many batteries of tests that have been developed to evaluate the French reading skills of children from Grade 1 to 4. The tests were developed by evaluating the accuracy and response time of approximately 100 children on a collection of words. This is an improvement over the BELEC (Mousty & Leybaert, 1999) tests, which were developed on 200 children from Grade 2 to 4, and did not take into consideration response time.

Development of a degree of difficulty that did not require testing of children would simplify the process. The assumption was that the degree of difficulty of a word would be a function of the phonemes that make up the word. However, there does not appear to be any correlation between the individual accuracies of each phoneme and the word

[2] http://www.readingrecovery.org
[3] http://www.fountasandpinnellleveledbooks.com

accuracy achieved by the students. The product, the sum, and the minimum of the phoneme accuracies generated by the student data and by Graphon, were compared to the word accuracies. The best and most consistent correlation was roughly 35% between the sum of the student phoneme accuracies and the word accuracies. A comparison of the correlations is shown in Table 19.

Table 19: Correlations between Word Accuracies and Calculated Word Difficulties

Grade	Student Phoneme Accuracy			Graphon Phoneme Accuracy		
	Sum	**Product**	**Min**	**Sum**	**Product**	**Min**
1	**0.36**	0.36	0.39	0.23	0.23	0.16
2	**0.36**	0.28	0.34	0.10	0.10	0.1
3	**0.31**	0.20	0.23	0.06	0.03	-0.04

As shown in Figure 9, the student phoneme accuracy increases across the board as the grade level increases; however, the same cannot be said for the overall word accuracy (Figure 10). In some cases, the word accuracy is actually worse for older students.

There must be more involved in the degree of difficulty of words than the difficulty of individual phonemes. Perhaps additional features should be considered, such as interaction of phonemes, the overall length of the word, and the frequency of the word, all of which impact the lexical procedure. The lexicality effect (the superiority of high frequency words compared to pseudowords) (Sprenger-Charolles, Colé, Béchaennec, & Kipffer-Piquard, 2005) can overshadow the effect of the degree of difficulty of the component phonemes of a word.

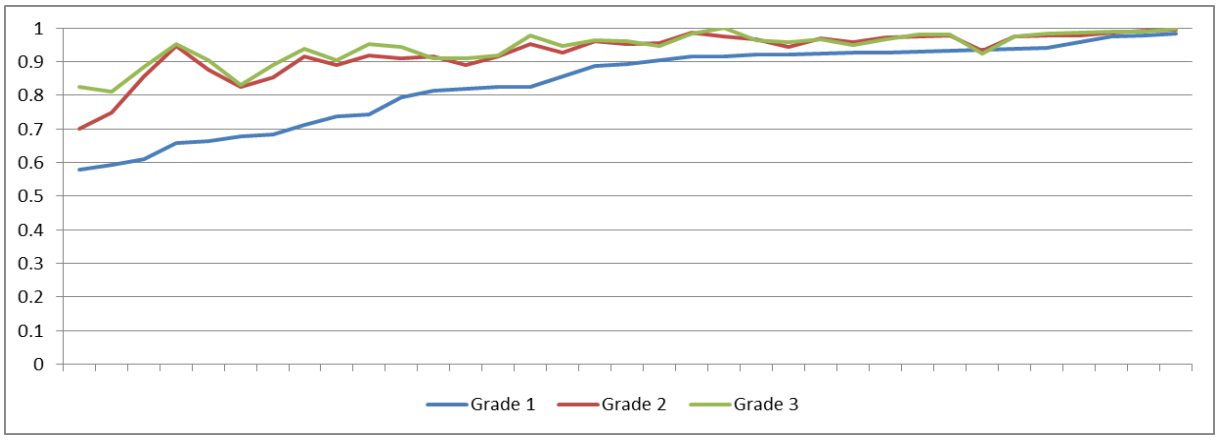

Figure 9: Comparison of Phoneme Accuracy across Grades

Comparison of phoneme accuracy for students. Phonemes are represented by the horizontal axis, and presented in increasing order of accuracy (vertical axis) for Grade 1 students.

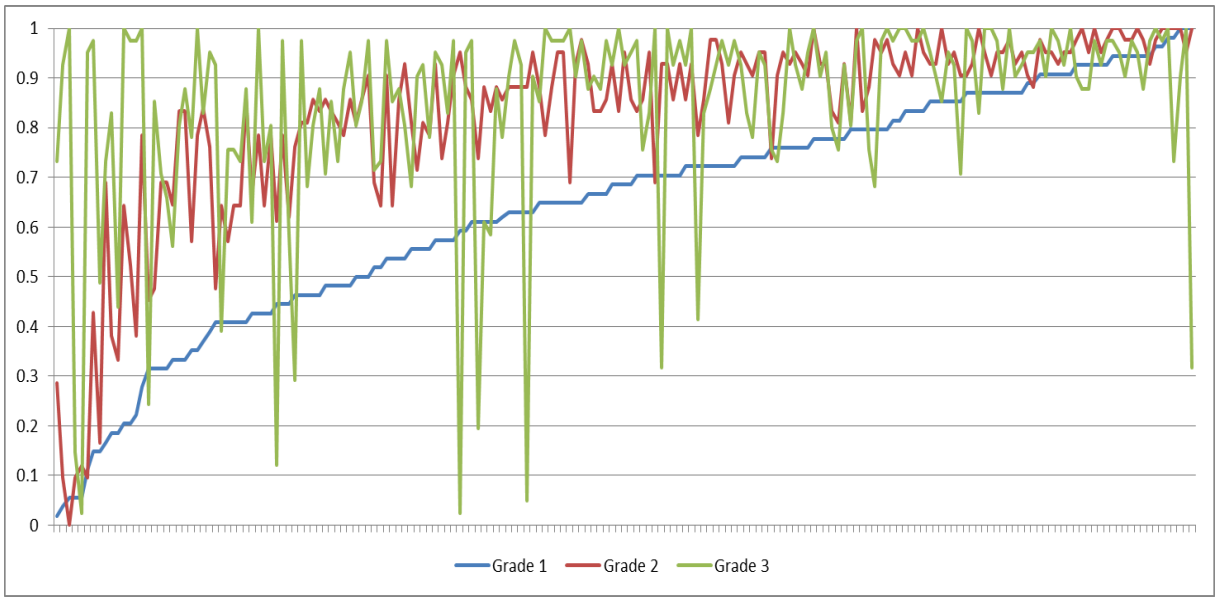

Figure 10: Comparison of Word Accuracy across Grades

Comparison of word accuracy for students. Words are represented by the horizontal axis, and presented in increasing order of accuracy (vertical axis) for Grade1 students.

8 Conclusions and Future Work

8.1 Conclusions

To summarize, word accuracy is improved by increasing the size of the training set, using more attributes, and applying a weight to these attributes. Even one occurrence of a word with its grapheme-to-phoneme mapping and up to eight attributes is beneficial, whereas there is a limit on the number of occurrences of a word at which the benefit stops. None of these statements is surprising. What is of interest is the comparison of data sets, not with regard to the words they contain, but to their different transcriptions for the same language but from different regions.

While there is some value in evaluating grapheme-to-phoneme conversion algorithms based on a specific data set, it appears that more flexibility and options need to be built into these systems. It should be possible to train a system using multiple training sets for a given language, and accept any of the possible transcriptions as correct.

While additional improvements are still required, with an initial correlation of more than 70% for Grades 1 and 3, it has been shown that it may be possible to build a system that can simulate a child's learning. Graphon could be used to generate a list of words with a specific degree of difficulty, measure the difficulty of a word, or test the effect on level of reading of exposure to a particular series of words.

As shown in Table 20, four phonemes account for the lowest three results in all three grades. Increasing exposure to these phonemes in Grade 1 could possibly achieve the greatest improvements in reading scores in the primary grades.

Table 20: Lowest Phoneme Accuracy for Students

Phoneme	Grade 1	Grade 2	Grade 3
Z	57.94	70.07	82.58
à	59.26	75.0	81.10
ë	61.11		
ÿ		82.54	82.93

8.2 Future Work

Given the amount of manual effort required to input the student data, only a portion was used for this project. In the future, it would be nice to include more data per grades as well as including Grades 4 to 6. More student data for Grades 1 to 3 may help to smooth out the outliers. While the accuracy rates of children in Grade 4 to 6 exceed the current maximum accuracy of the Graphon system, if this is ever improved, it would be interesting to see if it correlates well with the higher grades.

Only the results recorded on actual words were used as part of this project. The children were also tested on pseudowords. Training the system on the complete Manulex data set,

and testing on the pseudowords might provide a better correlation with the student results, since it would limit the students results to phonemic awareness, and remove the bias introduced by the lexicality effect (Sprenger-Charolles, Colé, Béchaennec, & Kipffer-Piquard, 2005) as discussed in section 7.5.

In an effort to increase the overall accuracy of the Graphon system, three areas for improvement are suggested by this research. The first is improvements to the data set, the second to alignment algorithm, and the third to the classifier.

Improvements to the transcriptions in the Omnilex data set, primarily those that differed from the student word list, and any that would be suggested by these discrepancies, would likely result in improvements as the training data would then better reflect the same transcription rules used to generate the expected output of the test data.

The largest improvements in word accuracy achieved by Jiampojamarn *et al*. (2010), whose research is currently state-of-the-art, were a result of using an M:M alignment algorithm rather than 1:1 as implemented by ALINE.

Lastly, improvements to the classifier could include: increasing the number of trees and implementing weighted attributes for the Random Forest classifier to see if the gains achieved by the weight attributes with the Naïve Bayes classifier could be replicated.

9 Bibliography

Alpaydin, E. (2010). *Introduction to Machine Learning* (2nd ed.). (T. Dietterich, Ed.) Cambridge, Massachusetts.

Andersen, O., Kuhn, R., Lazaridès, A., Dalsgaard, P., Haas, J., & Nöth, E. (1996). Comparison of two tree-structured approaches for grapheme-to-phoneme conversion. *Proceedings of the International Conference on Spoken Language Processing*, *3*, pp. 1700-1703. Philadelphia, PA.

Bahl, L., Das, S., deSouza, P., Epstein, M., Mercer, R., Merialdo, B., et al. (1991). Automatic phonetic base form determination. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, *1*, pp. 173-176. Toronto, Canada.

Baron, J., & Strawson, C. (1976). Use of Orthographic and Word-Specific Knowledge in Reading Words Aloud. *Journal of Experimental Psychology: Human Perception and Performance, 2*(3), 386-393.

Bartlett, S., Kondrak, G., & Cherry, C. (2008). Automatic Syllabification with Structured SVMs for Letter-To-Phoneme Conversion. *ACL HLT*, (pp. 568-576).

Black, A., Lenzo, K., & Pagel, V. (1998). Issues in building general letter to sound rules. *ESCA Workshop on Speech Synthesis,*, (pp. 77-80).

Boula de Mareuil, P. (2005). Evaluating the pronunciation of proper names by four French grapheme-to-phoneme converters. *Interspeech.*

Carson, R., & Desrochers, A. (2012). How grapheme type, word length and word frequency influence word reading accuracy across the primary grades: Evidence from French. *19th Annual Meeting of the Society for the Scientific Study of Reading.*

Cattell, J. (1886). The time taken up by cerebral operations. *Mind, 11*, 220-242, 377-392, 534-538.

Daelmans, W., & Van Den Bosch, A. (1997). Language-independent data-oriented grapheme-to-phoneme conversion. In J. Van Santen, R. Sproat, J. Olive, & J. Hirschberg, *Progress in Speech Synthesis* (pp. 77-89).

Demberg, V., Schmid, H., & Mohler, G. (2007). Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. *Association of Computationl Linguistics*, (pp. 96-103). Prague, Czech Republilc.

Desrochers, A. (2006). OMNILEX : Une base de données sur le lexique du français contemporain. *Cahiers Linguistiques d'Ottawa, 34*, 25-34.

Divay, M., & Vitale, A. (1997). Algorithms for Grapheme-Phoneme Translation for English and French: Applications for Dataabase Searches and Speech Synthesis. *Computational Linguistics, 23*(4), 495-523.

Dou, Q., Bergsma, S., Jiampojamarn, S., & Kondrak, G. (2009). A Ranking Approoach to Stress Prediction for Letter-to-Phnoneme Conversion. *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, (pp. 118-126). Suntec, Singapore.

Dufour, R. (2008). From prepared speech to spontaneous speech reconginition system: a comparative study applied to French language. *IEEE/ACM CSTST Student Workshop*, *1*, pp. 595-599. Cergy, France.

Dwyer, K., & Kondrak, G. (2009). Reducing the Annotation Effort for Letter-to-Phoneme Conversion. *Proceeding of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, (pp. 127-135).

Elovitz, H., Johnson, R., McHugh, A., & Shore, J. (1976). *Automatic Translation of English Text to Phonetics by Means of Letter-to-Sound Rules.* Naval Research Laboratory, Communication Sciences Division, Washington, D.C.

Emmit, M. (1998). *Understanding phonics and its role in literacy education*. Retrieved from www.discover.tased.edu.au/english/Emmitt.html

Fernandez-Delgado, M., Cernadas, E., & Barro, S. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classidifcation Problems? *Journal of Machine Learning Research, 15*, 3133-3181.

Glushko, R. (1979). The Organization and Activation of Orthographic Knowledge in Reading Aloud. *Journal of Experimental Psychology: Human Perception and Performance, 5*(4), 674-691.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA Data Mining Software: An Update;. *SIGKDD Explorations, 11*(1).

Huff, P. (2010). *PyAline: Automatically Growing Language Family Trees Using the ALINE Distance.* Thesis, Brigham Young University.

Humphries, J., Woodland, P., & Pearce, D. (1996). Using Accent-Specific Pronunciation Modelling for Robust Speech Recognition. *ICSLP*, (pp. 2324-2327).

Jiampojamarn, S., & Kondrak, G. (2010). Letter-Phoneme Alignment: An Exploration. *ACL*, (pp. 780-788).

Jiampojomarn, S., Cherry, C., & Kondrak, T. (2008). Joint Processing and Discriminative Training for Letter-to-Phoneme Conversion. *ACL HLT*, (pp. 905-913).

Jiampojomarn, S., Kondrak, G., & Sherif, T. (2007). Applying Mant-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. *NAACL HLT*, (pp. 372-379).

Kay, J., & Marcel, A. (1981). One process, not two, in reading aloud: Lexical analogies do the work of non-lexical rules. *Quarterly Journal of Experimental Psychology, 33*, 397-413.

Kondrak, G. (2000). A new algorithm for the alignment of phonetic sequence. *North American Chapter of the Association for Computational Linguistics*, (pp. 288-295).

Kranjc, J., Smailovic, J., Podpecan, V., Grcar, M., Znidarsic, M., & Lavrc, N. (2015). Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the ClowdFlows platform. *Information Processing & Management, 51*(2), 187-203.

Laberge, D., & Samuels, J. (1974). Towards a theory of automatic information processing in reading. *Cognitive Psychology, 6*, 293-323.

Laurent, A., Meignier, S., & Deléglise, P. (2014). Improving recognition of proper nouns in ASR through generating and filtering phonetic transcriptions. *Computer Speech and Language, 28*, 979-996.

Lété, B., Spreger-Charolles, L., & Colé, P. (2004). Manulex: A grade-level lexical database from French elementary-school readers. *Behavior Research Methods, Instruments, & Computers, 36*, 156-166.

Ma, Y., Fosler-Lussier, E., & Lofthus, R. (2012). Ranking-based readability assessment for early primary children's literature. *NACL: Human Language Technologies*, (pp. 548-552). Montreal, Canada.

Martin, P. (2004). DÉVOISEMENT VOCALIQUE EN FRANÇAIS. *La Linguistique, 40*(2), 3-21.

Martin, R., & Inkpen, D. (2006). *Implémentation d'un traducteur phonétique pour le français basé sur les données.*

Matwin, S., Kouznetsov, A., & Inkpen, D. (2010). A new algorithm for reducing the workload of experts in performing systematic reviews. *Journal of the American Medical Informatics Association*.

Mousty, P., & Leybaert, J. (1999). Evaluation des habiletés de lecture et d'orthographe au moyen de la BELEC: données longitudinales auprès d'enfants francophones testés en 2° et 4° années. *Revue Européenne de Psychologie Appliquée, 49*, 325-342.

New, B., Pallier, C., Ferrand, L., & Matos, R. (2001). Une base de données lexicales du français contemporain sur internet: LEXIQUE. *L'Année Psychologique*, 447-462.

Pagel, V., Lenzo, K., & Black, A. (1998). Letter to sound rules for accented lexicon compression. *International Conferenceon Spoken Language Processing*, (pp. 2015-2018). Sydney, Australia.

Pontes, J., & Furui, S. (2010). Predicting the phonetic realizations of word-final consonants in context – A challenge for French grapheme-to-phoneme converters. *Speech Communication, 52*, 847-862.

Quinlan, R. (1993). *C4.5: programs for machine learning.* San Francisco, CA: Morgan Kaufmann Publishers Inc.

Rama, T., Singh, A., & Kolachina, S. (2009). Modeling letter-to-phoneme conversion as a phrase based statistical machine translation problem with minimum error rate training. *North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, *1*, pp. 90-95. Boulder, Co.

Ramsay, A., Alsharhan, I., & Ahmed, H. (2014). Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model. *Computer Speech and Language, 28*, 959-978.

Riley, M., Byrne, W., Finke, M., Khudanpur, S., Ljolje, A., McDonough, J., et al. (1999). Stochastic pronunciation modelling from hand-labelled phonetic corpora. *Speech Communications, 29*, 209-224.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, 34*(1), 1-47.

Seng, K., Iribe, Y., & Nitta, T. (2011). Letter-to-phoneme conversion based on two-stage neural network focusing on letter and phoneme contexts. *International Speech Communication Association*, (pp. 1885-1888). Florence, Italy.

Snow, E., Burns, M., & Griffin, P. (1998). *Preventing Reading Difficulties in Young Children.* National Academy of Sciences, National Research Council, Washington.

Sprenger-Charolles, L. (1998). Reading and Spelling Acquisition in French: The Role of Phonological Mediation and Orthographic Factors. *Journal of Experimental Child Psychology, 68*, 134-165.

Sprenger-Charolles, L., Colé, P., Béchaennec, D., & Kipffer-Piquard, A. (2005). French normative data on reading and related skills from EVALEC, a new computerized battery of tests (end Grade 1, Grade2, Grade3, and Grade 4). *Revue Européenne de Psychologie Appliquée, 55*, 157-186.

Sultan, Z., & Jaffar, M. (2015). Facial expressions recognition using an ensemble of feature sets based on key-point descriptors. *Imaging Science, 63*(3), 160-167.

Suontausta, J., & Häkkinen, J. (2000). Decision tree based text-to-phoneme mapping for speech recognition. *International Conference on Spoken Language Processing*, *2*, pp. 831-834. Bejing, China.

Willows, D. (2008). Reducing Literacy Failure through Teacher Development: Implementing a Balanced and Flexible Literacy Diet. *Education Canada, 48*, 20-24.

Woolacott, T. (2002). *Teaching Reading in the Upper Primary School: A comparison of two teachers' approaches.* Australian Association for Research in Education, Brisbane.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *Knowlege and Information Systems, 14*(1), 1-37.

Yarowsky, D. (1997). Homograph Disambiguation in Text-to-Speech Synthesis. In J. Van Santen, R. Sproat, J. Olive, & J. Hirschberg, *Progress in Speech Synthesis* (pp. 157-171). New York: Springer-Verlag.

Yvon, F. (1994). Self-Learning techniques for grapheme-to-phoneme conversion. *Onomastica Research Colloquium.* London.

Yvon, F., Boula De Mareuil, P., D'Alessandro, C., Aubergé, V., Bagin, M., Bailly, G., et al. (1998). Objective evaluation of grapheme to phoneme conversion for text-to-speech synthesis in French. *Computer Speech and Language, 12*, 393-410.

Ziegler, J., Perry, C., & Zorzi, M. (2013). Modelling reading development through phonological decoding and self-teaching: implication for dyslexia. *Philosophical Transactions of the Royal Society, 369*.

Appendix A - Table of Phoneme Conversions

Aline Code	APF	Lex3	Brulex	Examples
aB	à	a	A	bas, pâte
aF	a	a	a	plat, patte
eC	e	ɜ	^	le, premier
e	é	é	é	blé, peler
eL	è	E	(	respect, lait, jouet
i	i	i	i	il, vie, lyre
oL	ò	ò	o	mort, donner
o	o	O	O	pot, dôme, eau, gauche
u	u	U	u	route, genou
uF	y	y	v	rue, vêtu
oF	ö	2	E	deux, muqueuse
oFL	ë	9	e	peur, meuble
eLN	ê	5	ê	matin, train, plein
aBN	â	@	â	vent, sans
oLN	ô	§	ô	bon, ombre
˜oFN	û	1	û	lundi, brun, parfum
y	j	J	ï	paille, pied
w	w	W	ü	oui, nouer
wF	ÿ	8	ÿ	lui, fuir
p	p	p	p	père, soupe
t	t	t	t	tas, vite
k	k	k	k	casse, qui, sac, képi
b	b	b	b	bal, robe
d	d	d	d	dos, aide
g	g	g	g	gare, bague
f	f	f	f	fou, neuf, photo
s	s	s	s	sable, ceci, tasse, nation
sV	S	S	/	chat, tache
v	v	v	v	vil, rêve
z	z	z	z	zèbre, maison
zV	Z	Z	j	gel, juin
l	l	l	l	lent, sol
r	r	r	R	riz, terre, venir
m	m	m	m	main, femme
n	n	n	n	nez, tonne, animal

nP	N	N	N	vigne, agneau
gN	G	G	@	camping
h	h		'	hop !
x	x	x	x	Jota

Appendix B – Sample Student Data Collection Form

Phase2_V1_Fa_partie II – Lecture de mots

#	Code	Stimulus	Transcription phonétique	Autre réponse D SEG ER SR	Transcrire
1		vu	vy ✓	D SEG ER SR	
2		sa	sa ✓	D SEG ER SR	
3		ne	ne ✓	D SEG ER SR	
4		ma	ma ✓	D SEG ER SR	
49	lecmA 33	sentiment	sâtimâ	D SEG ER SR	e
50	lecmA 88	scier	sjé	D SEG ER SR	stié
51	lecmA 04	mère	mèr ✓	D SEG ER SR	
52	lecmA 86	clef	klè	D SEG ER SR	Kèf
53	lecmA 10	rivière	rivjèr ✓	D SEG ER SR	
54	lecmA 66	cèdre	sèdr ✓	D SEG ER SR	
55	lecmA 36	question	kèstjô ✓	D SEG ER SR	
56	lecmA 75	vingt	vê ✓	D SEG ER SR	
57	lecmA 90	saoul	su	D SEG ER SR	saul
58	lecmA 74	pied	pjé ✓	D SEG ER SR	
59	lecmA 05	juste	Zyst ✓	D SEG ER SR	
60	lecmA 57	recevoir	r(e)sevwar; res(e)vwar	D SEG ER SR	K
61	lecmA 72	caractérologie	karaktérolòZi	D SEG ER SR	gi
62	lecmA 13	brume	brym ✓	D SEG ER SR	
63	lecmA 78	sept	sèt ✓	D SEG ER SR	
64	lecmA 51	grâce	gras	D SEG ER SR	a
65	lecmA 07	tenir	t(e)nir ✓	D SEG ER SR	
66	lecmA 11	problème	pròblèm ✓	D SEG ER SR	
67	lecmA 06	ami	ami ✓	D SEG ER SR	
68	lecmA 44	mouchoir	muSwar ✓	D SEG ER SR	
69	lecmA 62	gêne	Zèn	D SEG ER SR	
70	lecmA 77	hier	jè	D SEG ER SR	gijèr
71	lecmA 69	gravité	gravité ✓	D SEG ER SR	
72	lecmA 40	muet	mÿè	D SEG ER SR	é
73	lecmA 23	promenade	pròm(e)nad ✓	D SEG ER SR	
74	lecmA 14	tiède	tjèd	D SEG ER SR	tièd
75	lecmA 02	vu	vy ✓	D SEG ER SR	
76	lecmA 61	gage	gaZ ✓	D SEG ER SR	
77	lecmA 46	bouquet	bukè	D SEG ER SR	é
78	lecmA 53	ciel	sjèl ✓	D SEG ER SR	
79	lecmA 63	glace	glas ✓	D SEG ER SR	
80	lecmA 58	caractère	karaktèr ✓	D SEG ER SR	

	ecmA 37	sauvé	sové ✓	D SEG ER SR	
	lecmA 50	guère	gèr ✓	D SEG ER SR	
83	lecmA 96	scandaleux	skâdalö	D SEG ER SR	lër
84	lecmA 82	compter	kôté ✓	D SEG ER SR	
85	lecmA 94	automne	òtòn; otòn ✓	D SEG ER SR	
86	lecmA 95	eucharistie	ökaristi	D SEG ER SR	..S...
87	lecmA 56	regard	r(e)gar	D SEG ER SR	+d
88	lecmA 19	timidité	timidité ✓	D SEG ER SR	
89	lecmA 28	effet	éfé	D SEG ER SR	é
90	lecmA 65	code	kòd	D SEG ER SR	god
91	lecmA 67	généreux	Zénérö ✓	D SEG ER SR	
92	lecmA 81	longtemps	lôtâ ✓	D SEG ER SR	
93	lecmA 29	signe	siN ✓	D SEG ER SR	
94	lecmA 85	gars	gà	D SEG ER SR	gar
95	lecmA 12	manière	manjèr	D SEG ER SR	nier
96	lecmA 01	mur	myr ✓	D SEG ER SR	

Appendix C – Student Test Words

absorbé	compter	gerbe	nuque	sauvé
aiguille	connaitre	germain	obstiné	scandaleux
ami	connu	geste	œil	scander
automne	conscience	glace	orchestre	science
avertir	corps	gouverneur	paon	scier
beau	cri	grâce	paquet	second
beaucoup	cube	grave	parmi	sentiment
borne	culture	gravitè	parole	sept
bouquet	déçu	guère	pâte	signe
bourreau	descendre	guérison	père	simultanéité
boutique	devenir	guise	pied	six
brin	digne	hier	piscine	sixième
brume	dix	huit	poids	survivre
brut	doigt	humain	poli	susceptible
caractère	donc	indépendant	porte	tempérament
caractère	doute	instinct	pouls	tenir
caractérologie	eau	intérieur	pourquoi	tiède
catégorique	ébène	jupe	pourtant	timidité
causalité	ecchymose	juste	problème	tombeau
céder	effet	liberté	projet	tordu
cèdre	eucharistie	ligne	promenade	travers
célébrer	exact	lire	propriété	triomphe
celle	femme	loin	pur	valet
cependant	fermeture	longtemps	quel	véritable
cercueil	fin	lorsque	question	vérité
certain	fine	lumière	quoi	vigne
ceux	flaque	maintenant	rastaquouère	vingt
chaos	fontaine	mal	ravi	vite
chez	force	manière	recevoir	vol
chœur	formidable	marmite	regard	vraiment
chrétien	fourbe	mauvais	regarder	vu
ciel	gage	mère	régulier	yacht
cire	garçon	monsieur	répondre	
cité	gars	mouchoir	respect	
clef	gaz	mouton	revenir	
clerc	gêne	muet	rivière	
code	général	mur	rondin	
cœur	généreux	murmure	rouquin	
comprendre	genre	nature	route	
compte	gens	niveau	saoul	

Appendix D – Sample .arff File

```
@RELATION ë
@ATTRIBUTE letter+1{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter-1{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter+2{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter-2{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter+3{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter-3{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter+4{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter-4{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter+5{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE letter-5{f,ï,g,î,d,e,b,ë,c,ê,é,a,è,ç,n,o,l,ä,m,j,â,k,h,i,à,-
,w,v,u,ü,t,s,r,û,q,p,ù,ö,ô,z,y,x,ñ}
@ATTRIBUTE class{é,è,-}
@DATA
k,o,a,n,y,a,a,c,k,-,é
n,u,e,o,-,f,-,-,-,-,è
t,o,e,b,-,-,-,-,-,-,è
l,o,-,n,-,e,-,d,-,e,è
s,o,s,b,e,-,r,-,-,-,è
l,o,-,n,-,e,-,d,-,e,è
r,o,-,b,-,o,-,l,-,g,è
l,o,-,n,-,e,-,r,-,è,è
n,o,e,f,r,-,-,-,-,-,è
n,o,e,f,r,-,-,-,-,-,è
l,o,-,n,-,-,-,-,-,-,è
m,r,-,t,-,s,-,l,-,e,è
l,a,-,g,-,-,-,-,-,-,è
l,a,i,r,e,s,n,i,-,d,è
n,u,e,o,r,f,-,-,-,-,è
n,u,e,o,r,f,-,-,-,-,è
s,o,s,b,e,-,-,-,-,-,è
-,u,-,g,-,i,-,b,-,m,-
-,u,-,g,-,i,-,a,-,s,-
-,u,-,g,-,i,-,x,-,e,-
-,u,-,g,-,i,-,a,-,r,-
-,u,-,g,-,i,-,c,-,-,-
-,u,-,g,-,i,-,t,-,n,-
-,u,-,g,-,i,-,a,-,s,-
-,u,-,g,-,i,-,a,-,b,-
-,u,-,g,-,i,-,a,-,-,-
```

Appendix E – Sample .xrff File

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE dataset
[
   <!ELEMENT dataset (header,body)>
   <!ATTLIST dataset name CDATA #REQUIRED>
   <!ATTLIST dataset version CDATA "3.7.9">

   <!ELEMENT header (notes?,attributes)>
   <!ELEMENT body (instances)>
   <!ELEMENT notes ANY>   <!--  comments, information, copyright, etc. -->

   <!ELEMENT attributes (attribute+)>
   <!ELEMENT attribute (labels?,metadata?,attributes?)>
   <!ATTLIST attribute name CDATA #REQUIRED>
   <!ATTLIST attribute type (numeric|date|nominal|string|relational) #REQUIRED>
   <!ATTLIST attribute format CDATA #IMPLIED>
   <!ATTLIST attribute class (yes|no) "no">
   <!ELEMENT labels (label*)>   <!-- only for type "nominal" -->
   <!ELEMENT label ANY>
   <!ELEMENT metadata (property*)>
   <!ELEMENT property ANY>
   <!ATTLIST property name CDATA #REQUIRED>

   <!ELEMENT instances (instance*)>
   <!ELEMENT instance (value*)>
   <!ATTLIST instance type (normal|sparse) "normal">
   <!ATTLIST instance weight CDATA #IMPLIED>
   <!ELEMENT value (#PCDATA|instances)*>
   <!ATTLIST value index CDATA #IMPLIED>   <!-- 1-based index (only used for
instance format "sparse") -->
   <!ATTLIST value missing (yes|no) "no">
]
>

<dataset name="Rel" version="3.7.9">
   <header>
      <attributes>
         <attribute name="letter+1" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">1.0</property>
            </metadata>
```

```
         </attribute>
         <attribute name="letter-1" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">1.0</property>
            </metadata>
         </attribute>
         <attribute name="letter+2" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.25</property>
            </metadata>
         </attribute>
         <attribute name="letter-2" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.25</property>
            </metadata>
         </attribute>
         <attribute name="letter+3" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
```

```
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.0625</property>
            </metadata>
         </attribute>
         <attribute name="letter-3" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.0625</property>
            </metadata>
         </attribute>
         <attribute name="letter+4" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.015625</property>
            </metadata>
         </attribute>
         <attribute name="letter-4" type="nominal">
            <labels>

<label>œ</label><label>f</label><label>ï</label><label>g</label><label>î</label
><label>d</label><label>e</label><label>b</label><label>ë</label><label>c</labe
l><label>ê</label><label>é</label><label>a</label><label>è</label><label>ç</lab
el><label>n</label><label>o</label><label>l</label><label>ä</label><label>m</la
bel><label>ã</label><label>j</label><label>â</label><label>k</label><label>h</l
abel><label>i</label><label>à</label><label>-
</label><label>w</label><label>v</label><label>u</label><label>ü</label><label>
t</label><label>s</label><label>r</label><label>û</label><label>q</label><label
```

```
>p</label><label>ö</label><label>ô</label><label>z</label><label>y</label><labe
l>x</label><label>ñ</label>
            </labels>
            <metadata>
               <property name="weight">0.015625</property>
            </metadata>
         </attribute>
         <attribute class="yes" name="classVal" type="nominal">
            <labels>
               <label>è</label><label>â</label>
            </labels>
         </attribute>
      </attributes>
   </header>
   <body>
      <instances>
         <instance>
            <value>n</value>
            <value>s</value>
            <value>g</value>
            <value>e</value>
            <value>e</value>
            <value>n</value>
            <value>r</value>
            <value>n</value>
            <value>è</value>
         </instance>
         <instance>
            <value>k</value>
            <value>b</value>
            <value>e</value>
            <value>-</value>
            <value>o</value>
            <value>-</value>
            <value>f</value>
            <value>-</value>
            <value>è</value>
         </instance>
         <instance>
            <value>n</value>
            <value>l</value>
            <value>d</value>
            <value>-</value>
            <value>e</value>
            <value>-</value>
            <value>r</value>
            <value>-</value>
            <value>â</value>
         </instance>
      </instances>
   </body>
</dataset>
```

Appendix F – Phoneme Accuracies (Omnilex)

	Gr1 Graphon	Gr1 Student	Gr2 Graphon	Gr2 Student	Gr3 Graphon	Gr3 Student
à	0.00%	59.26%	50.00%	75.00%	75.00%	81.10%
â	95.24%	81.48%	95.24%	91.61%	95.24%	91.06%
a	97.92%	92.98%	97.92%	97.62%	97.92%	98.07%
b	100.00%	92.42%	100.00%	97.05%	100.00%	96.52%
d	100.00%	93.16%	100.00%	97.71%	100.00%	98.12%
é	89.36%	82.52%	93.62%	91.64%	93.62%	91.90%
e	68.75%	82.60%	62.50%	95.09%	68.75%	97.87%
è	100.00%	74.31%	100.00%	91.87%	97.87%	95.12%
ê	93.33%	79.29%	100.00%	91.13%	100.00%	94.27%
f	100.00%	95.79%	100.00%	97.84%	100.00%	98.67%
g	100.00%	85.69%	100.00%	92.63%	100.00%	94.61%
i	96.23%	92.76%	96.23%	97.18%	100.00%	96.66%
k	87.23%	82.02%	89.36%	89.05%	89.36%	90.95%
l	100.00%	97.41%	100.00%	98.49%	100.00%	98.78%
m	100.00%	98.44%	100.00%	99.40%	100.00%	99.85%
n	95.65%	91.41%	95.65%	98.48%	100.00%	98.45%
N	100.00%	65.74%	100.00%	94.64%	100.00%	95.12%
o	100.00%	93.60%	100.00%	93.29%	100.00%	92.46%
ö	80.00%	68.40%	60.00%	85.23%	100.00%	89.02%
ë	83.33%	61.11%	83.33%	85.71%	83.33%	88.50%
ò	96.30%	73.87%	96.30%	89.07%	96.30%	90.51%
ô	100.00%	89.37%	100.00%	95.13%	100.00%	96.14%
p	96.67%	93.95%	100.00%	97.54%	100.00%	97.40%
r	97.30%	94.14%	99.10%	97.84%	98.20%	98.23%
s	94.23%	88.63%	96.15%	96.16%	96.15%	96.29%
S	100.00%	91.67%	100.00%	97.62%	100.00%	100.00%
t	91.67%	90.34%	91.67%	95.60%	96.67%	94.59%
u	100.00%	92.22%	100.00%	94.29%	100.00%	95.77%
y	100.00%	92.15%	100.00%	96.70%	100.00%	96.34%
v	100.00%	97.85%	100.00%	99.22%	100.00%	98.98%
w	87.50%	92.59%	87.50%	95.83%	75.00%	94.82%
ÿ	66.67%	67.90%	100.00%	82.54%	66.67%	82.93%

j	100.00%	66.33%	95.45%	87.64%	90.90%	90.48%
z	57.14%	57.94%	71.43%	70.07%	100.00%	82.58%
Z	100.00%	71.08%	100.00%	91.58%	92.31%	93.81%