Automatic Generation of English Respellings

Bradley Hauer and Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{bmhauer,gkondrak}@ualberta.ca

Abstract

A *respelling* is an alternative spelling of a word in the same writing system, intended to clarify pronunciation. We introduce the task of automatic generation of a respelling from the word's phonemic representation. Our approach combines machine learning with linguistic constraints and electronic resources. We evaluate our system both intrinsically through a human judgment experiment, and extrinsically by passing its output to a letter-to-phoneme converter. The results show that the respellings generated by our system are better on average than those found on the Web, and approach the quality of respellings designed by an expert.

1 Introduction

Respellings are a widely employed method of conveying the pronunciation of English and foreign words, both in print and on the Web. For example, *Huatulco*, the name of a Mexican resort, is respelled as *'wah-tool-koh'* in a travel guide (Noble, 2012). The advantage of using respellings lies in removing the need for a separately defined phonetic transcription system. Since they contain only the letters of the Latin alphabet, their phonetic interpretation relies exclusively on orthographic intuitions of readers. For this reason, respellings are widely used in travel phrase books, medical compendia, and drug name pronunciation guides, among others.

Despite their utility, good respellings are not easy to create. Respellings found on the Web often contain errors or ambiguities. For example, *Henoch-Schoenlein purpura*, a skin disease, is respelled both as *'heh-nok shoon-line purr-puh-ruh'* and *'hen-awk sher-line purr-purr-ah'*. Does *'heh'* rhyme with *eh* [e] or with *Nineveh* [ə], or is it the same vowel as in *hen* [ɛ]? Clearly, if both respellings refer to the same pronunciation, at least one of them must be wrong. In addition, converting the pronunciation of a foreign name to English phonemes is in itself a non-trivial task.

In this paper, we focus on the task of generating respellings from the intended pronunciation given as a sequence of phonemes. We develop a stand-alone system that combines linguistic knowledge and resources with machine learning models trained on data mined from the Web and electronic dictionaries. One of our ultimate objectives is to aid writers by evaluating their respellings, improving them, or generating new candidates. Accordingly, we endeavour to maintain the generation and the evaluation stages as separate modules in our system.

The evaluation of respellings is a challenging problem. Since English spelling conventions are notoriously inconsistent, there is no algorithm for accurately predicting the pronunciation of an out-of-vocabulary word. The current state-of-the-art letter-to-phoneme (L2P) converters are typically reported with 10-30% error rates on dictionary words (Bisani and Ney, 2008). On the other hand, human readers often disagree on the details of the pronunciation implied by a respelling. In this paper, we conduct two kinds of evaluations: an automated verification with an independent L2P system, and an experiment with human participants that pass judgments on different respellings of the same word. We interpret the results as evidence that the output of our system compares favourably with typical respellings found on the Web.

Proceedings of NAACL-HLT 2013, pages 634–643,
Atlanta, Georgia, 9–14 June 2013.

2 Definitions and Conventions

Although Chomsky and Halle (1968) characterize English orthography as close to optimal, Kominek and Black (2006) estimate that it is about 3 times more complex than German, and 40 times more complex than Spanish. This is confirmed by lower accuracy of letter-to-phoneme systems on English (Bisani and Ney, 2008). A survey of English spelling (Carney, 1994) devotes 120 pages to describe phoneme-to-letter correspondences, and lists 226 letter-to-phoneme rules, almost all of which admit exceptions.

There is no consensus on how to best convey the pronunciation of an uncommon word in English. Most dictionaries employ either the International Phonetic Alphabet (IPA), or their own transcription schemes that incorporate special symbols and diacritics. Unfortunately, many readers are unfamiliar with phonetic transcription. Instead, respellings are often preferred by writers in the news and on the Web. In this section, we define the respelling task in detail.

2.1 Form of Respellings

A *respelling* is a non-standard spelling of a word, that is intended to better convey its pronunciation. We assume that the pronunciation is defined as a sequence of English phonemes, and that the respelling contains only the 26 letters of the alphabet, with optional hyphenation. Some transcription schemes combine respellings with special symbols for representing certain phonemes. For example, an otherwise purely alphabetic Wikipedia scheme employs the symbol ə for the vowel *schwa*. In our opinion, such devices destroy the main advantage of respellings, which is their universality, without attaining the precision of a true phonetic transcription. In fact, Fraser (1997) identifies the *schwa* symbol as the cause of many pronunciation errors.

In our system, we consistently use hyphens to segment multi-syllable respellings. Each syllable-size segment contains the representation of exactly one vowel phoneme, so that the number of segments matches the number of syllables.[1] However, the hyphenation need not correspond exactly to the actual syllable breaks. This approach has several advantages. First, individual syllables are easier to pronounce than an entire unfamiliar word. Second, hyphens limit the context that affects the pronunciation of a given letter (e.g. *th* in *Beethoven 'bayt-hoe-ven'*). Finally, hyphens indicate whether adjacent vowel letters, such as *oe* in *'hoe'*, represent one vowel phoneme or two.

Some respellings explicitly indicate the stressed syllable by expressing it in a different font. This is potentially helpful because unstressed vowels tend to be reduced, which changes their pronunciation. However, since the vowel reduction phenomenon is by no means universal, the readers may be unsure whether to apply it to, e.g. the final *o* in *'KWAT-ro'*. In this paper, we make no distinction between stressed and unstressed syllables; instead, we follow the principle that each syllable is to be pronounced as if it was a separate word. Nonetheless, it would be straightforward to project the stress indicators onto the appropriate syllables in the respellings generated by our system.

2.2 Quality of Respellings

There is no clear-cut distinction between good and bad respellings. The quality of a respelling is more of a subjective opinion rather than a verifiable fact. We propose to evaluate it according to the following three criteria: ambiguity, correctness, and preference.

A respelling is *ambiguous* if it is perceived as compatible with more than one pronunciation. Because most of the rules of English spelling have exceptions, it is rarely possible to demonstrate that a respelling is completely unambiguous. However, some respellings are clearly more ambiguous than others. For example, the digraph *ee* almost always represents the vowel [i], whereas the letter sequence *ough* can represent several different phonemes.[2] Respellings that contain highly ambiguous letter-phoneme mappings can be expected to be ambiguous themselves. Ambiguity is a property of a respelling itself, regardless of the intended pronunciation.

A respelling is *correct* if it accurately conveys the intended pronunciation to the reader. Unlike the am-

[1] Henceforth, we refer to "syllable-size segments" simply as "syllables".

[2] Compare *bough, cough, dough, tough, lough, through*.

biguity, correctness can be verified objectively for a particular reader, by comparing the intended pronunciation with the pronunciation inferred by the reader. A respelling that is judged correct with respect to one pronunciation cannot be judged correct with respect to a different pronunciation. Nevertheless, it is entirely possible that different readers will derive different pronunciations from the same respelling.

A respelling can be classified as unambiguous and yet incorrect by a given reader, but it cannot be judged as simultaneously ambiguous and correct. Indeed, an ambiguous respelling is compatible with at least two pronunciations, only one of which can be the intended pronunciation. Therefore, for a given reader, unambiguity is a necessary but not sufficient condition for correctness.

Given two unambiguous and correct respellings, a reader may prefer one over the other, perhaps because of the ease of inferring the intended pronunciation. For example, *'rode-ease-yew'* may be preferred to *'roh-dee-zyoo'* because the former is entirely composed of actual English words with unique pronunciation, whereas the latter contains an unusual consonant cluster *zy*. Preference is also expressed implicitly if only one of the alternative respellings is judged as unambiguous (or correct),

3 Related Work

Fraser (1997) describes an experiment in which 15 human subjects were asked to pronounce uncommon words after being shown a representation of their pronunciation. The respellings designed by the author were much more effective for that purpose than either the IPA phonetic transcription or phonemic respelling (Section 4.3). However, the creation of respellings was described as labour-intensive, and at least one of them was found to be sub-optimal during the experiment.

Williams and Jones (2008) propose respellings as a way of extending pronunciation lexicons by informants who lack linguistic training. Galescu (2009) reports that the addition of respellings of medical terms from an on-line dictionary improves the accuracy of an L2P system. The author identifies an automatic pronunciation-to-respelling system as future work.

Ghoshal et al. (2009) extract a large number of respellings from the Web, and show that they can be exploited to improve the accuracy of the L2P conversion by supplementing the data in pronunciation dictionaries. Can et al. (2009) further analyze the effect of using respellings on the accuracy of spoken-term detection (STD) systems.

4 Direct Methods

In this section, we discuss three direct methods of generating respellings: manual design, dictionary lookup, and phonemic respelling.

4.1 Manual Design

Respellings found on the Web and in news articles are usually ad-hoc creations of the authors of those texts. Respellings designed by different writers for the same word are rarely identical.[3] The quality of Web respellings vary.

The respellings found in specialized lexicons are more likely to be designed by experts, and are often guided by a set of respelling rules. Nevertheless, such respelling guides may also be ambiguous.[4] Regardless of the source, since respellings are often used for names and foreign words, no lexicon can be expected to provide a complete coverage.

4.2 Dictionary Lookup

Pronunciation dictionaries can be helpful in generating respellings. Assuming that we have a method of dividing pronunciations into syllables, a complete respelling of an out-of-dictionary word can in some cases be automatically derived from the list of syllable pronunciations. For example, *hyphy* can be respelled as *'high-fee'* by following such a procedure. If each of the syllables has a unique pronunciation, such respellings are arguably both unambiguous and correct.

Unfortunately, only a subset of potential phonemic syllables actually occur in a lexicon. Considering only the syllables of the CVC type (consonant-vowel-consonant), there are over ten thousand distinct possibilities (e.g., [bɛb], [bɛʃ], etc.), of which

[3] For example, the word *capoeira* is represented by 99 different respellings in the corpus of Ghoshal et al. (2009).

[4] For an example of a confusing respelling guide see `http://www.ama-assn.org/go/usan`.

fewer than three thousand can be found in the Combilex pronunciation dictionary (Richmond et al., 2009). While the dictionary lookup may produce attractive respellings, it is not sufficient for a standalone use.

4.3 Phonemic Respelling

A simple method that can produce a respelling for any word is to directly map each phoneme to a particular letter or a letter sequence that is frequently used to represent that phoneme. Phonemes such as [m], [d] and [f] are indeed closely associated with individual letters. This is not surprising since the Roman letters were originally created to represent single phonemes in Latin, and some of those phonemes also exist in English. However, many phonemes, especially vowels, have no obvious orthographic representation. One solution is to use digraphs such as *ee* and *aw*, but a number of phonemes, such as [aʊ] as in *loud*, have no mappings that work in all contexts.

The principal weakness of a phonemic respelling is its inflexibility, which often results in counterintuitive respellings. For example, many readers are baffled by respelling such as *'gee'* for *ghee* or *'john'* for *Joan*. Phonemic respelling tends to fail in cases where it generates a sequence of letters that is inherently ambiguous, or which pronunciation changes because of the context. On the other hand, mappings such as *uu* for [ʊ] and *ahy* for [aɪ], which never occur in real English words, are difficult to interpret for some readers.

In this paper, we adopt a context-free phonemic respelling scheme as the baseline, with the mappings from the online dictionary *Dictionary.com*, which differs from the system used in Wikipedia only in a few details.

5 Candidate Generation

In this section, we present our syllabification approach, as well as two generation modules: a trained phoneme-to-letter (P2L) model and a rule-based respeller.

5.1 Syllabification

Our respelling generation process is for the most part performed on the level of individual syllables.

	VOWEL	ONSET	LAX	CODA
nt	*			
ndən		*		
bæ			*	
dənm				*
bæn				

Table 1: Examples of syllables that violate phonotactic constraints.

Correct syllabification is by itself a non-trivial problem, but even if it was provided by an oracle, it might not correspond to the optimal segmentation of a respelling. For example, the word *trigonal* [trɪgənəl] is usually syllabified as *tri-go-nal*, but a better segmentation for the purposes of respelling is *trig-on-al*. We adopt an overgenerate-and-rank approach, whereby instead of committing to a specific word segmentation at the start of the process, we process multiple syllabification alternatives in parallel, one of which is ultimately selected at the respelling evaluation stage.

Ideally, syllabification should conform to the phonotactic constraints of English, so that the resulting respellings are easy to pronounce. The consonant sonority should be rising in onsets, and falling in codas (Kenstowicz, 1994). We verify that syllables follow the sonority principle by following the formulation of Bartlett et al. (2009). The sonority constraints are not tested at the boundaries of the word, which are independent of the syllabification choice. We also incorporate another important principle of English phonotactics that asserts that lax vowels do not occur in open syllables (Rogers, 2000).

In our implementation, each candidate syllable is tested with respect to the following sequence of four violable constraints, ordered from the strongest to the weakest: (1) the syllable contains exactly one *vowel* phoneme; (2) the *onset* satisfies the sonority principle; (3) if the nucleus contains a *lax* vowel (except ə), the coda is non-empty; (4) the *coda* satisfies the sonority principle. For a syllabification to be accepted, all its syllables must satisfy the four constraints. However, if this results in rejection of all possible syllabifications, the constraints are gradually relaxed starting from the weakest.

As an example, consider the word *abandonment* [əbændənmənt], which has 18 different syllabifications satisfying the VOWEL constraint (Table 1). 8 of the 18 satisfy the ONSET constraint as well, but only two syllabifications satisfy all four constraints: [əb-æn-dən-mənt] and [ə-bæn-dən-mənt].

5.2 P2L Generator

The respelling problem can be viewed as a string transduction problem, with the transduction occurring between phonemes and letters. As such, it is directly related to the well-studied letter-to-phoneme conversion task. The difference is that the letters may not conform to the standard orthography of English. If we had a sufficiently large training set of pronunciation-respelling pairs, we could train a machine learning algorithm to directly generate respellings for any strings of English phonemes. However, such a training set is not readily available. The respellings in the corpus collected by Ghoshal et al. (2009) are not easily matched to the phonetic transcriptions, and few of them can be found in electronic pronunciation dictionaries. In addition, the quality of Web respellings vary greatly.

In place of a direct pronunciation-to-respelling model, we aim to model the orthographic intuitions of readers by deriving a phoneme-to-letter (P2L) transduction model from an English pronunciation dictionary. A possible criticism of such an approach is that our model may create ambiguous respellings, which abound in English orthography. However, we rely on a separate evaluation module to identify and filter ambiguous respellings at a later stage.

Our systems utilizes the DIRECTL+ program (Jiampojamarn et al., 2008), which was originally designed for L2P conversion. Since our basic unit is the syllable, rather than the word, we train our P2L model on a set of of 4215 pairs of monosyllabic words and their pronunciations extracted from the Combilex dictionary. We exclude syllables in multisyllabic words from training because their pronunciation is often affected by context. This is consistent with our expectation that the reader will pronounce each hyphen-delimited segment of the respelling as if it was an individual word.

Since the P2L training data consists of a relatively small set of syllables, we ensure that the phoneme-letter alignment is highly accurate. As a preprocessing step, we replace the letter x with *ks*, and we convert digraphs, such as *ch* and *th*, to single symbols. The alignment is performed by M2M-ALIGNER (Jiampojamarn et al., 2007), under the restriction that each phoneme is matched to either one or two letter symbols.

5.3 Context-Sensitive Respeller

A hand-crafted context-sensitive respeller is intended to complement the trained P2L model described in the previous section. It is similar to to the phonemic respelling approach described in Section 4.3 in that it converts each phoneme to a letter sequence. However, the mappings depend on adjacent phonemes, as well as on the CV pattern of the current syllable. In addition, more than one mapping for a phoneme can be proposed. We designed the mappings by analyzing their frequency and consistency in pronunciation dictionaries.

The process of candidate generation involves establishing the pattern of consonants in the input syllable. The consonant mappings are the same as in the baseline, except for [g] and [θ], while the vowels yield up to three different letter sequences. For example, [o] is mapped to *oh* as a default, but also to *o* if both onset and coda are empty, or to *o* followed by a consonant and a silent *e* if the coda is composed of a single consonant. So, given the syllable [tok] as input, the respeller produces two candidates: *tohk* and *toke*.

We make no claims about the completeness or optimality of the mappings, but in our development experiments we observed that the context-sensitive respeller contributes to the robustness of our system, and in some cases produces more attractive respellings that the P2L model.

6 Candidate Selection

We aim at developing a stand-alone method for the assessment of respellings that could be applied regardless of their origin. We consider two criteria: correctness, which is evaluated against the intended pronunciation, and ambiguity, which is a property of the respelling itself. As was the case in the generation stage, the evaluation is performed at the level of syllables.

6.1 L2P Correctness Filter

The principal method of verifying the correctness of a respelling involves the application of a letter-to-phoneme (L2P) model trained on the word-pronunciation pairs extracted from an English dictionary. The generated pronunciation of each syllable is compared against its intended pronunciation; if any of the syllables fail the test, the entire respelling is rejected.

The L2P model is derived using the DIRECTL+ system. The main difference between the L2P model described in this section and the P2L model from Section 5.2 is that the input and output data are reversed. However, the L2P model is not simply a mirror image of the P2L model. Often the phonemic output of the composition of the two models is different from the initial phonemic input; e.g., [ro] → *row* → [raʊ]. This is because the intermediate orthographic string may be ambiguous. Furthermore, the L2P model is also intended to test the correctness of respellings that were generated with other methods.

Other differences between the two models pertain to the preprocessing of the training data, and the letter-to-phoneme alignment. As with the P2L model, the training data consists of a set of monosyllabic words from the Combilex dictionary. However, in order to make our correctness filter more conservative, we also remove all words that contain diacritics (e.g., *crêpe*), non-English phonemes (e.g., *avant*), or silent consonants (e.g., *limn*). The alignment is restricted to matching each letter symbol to at most one phoneme, and is derived with the ALINE phonetic aligner (Kondrak, 2000), which has been shown to outperform other 1-1 alignment methods (Jiampojamarn and Kondrak, 2010).

6.2 Vowel Counter

Syllables that contain multiple vowel groups may be confusing to readers even if they correctly represent the intended pronunciation. For example, readers might be unsure whether *takess* represents one or two syllables. A simple vowel counter is provided to filter out such syllables. The vowel filter accepts a syllable only if (a) it contains exactly one vowel group (e.g., *moe*), or (b) the second vowel group consists of a single *e* at the end of the syllable (e.g., *zake*).

6.3 SVM Ambiguity Classifier

This module is designed to compute a score that reflects the ambiguity of an orthographic syllable. The ambiguity score of a respelling is defined as the average of scores assigned to each of its syllables. The score can then be used to select the best respelling from a number of candidates generated by our system, or to rate a respelling from another source.

Since we have no explicit ambiguity annotations for respellings, we attempt instead to exploit ambiguity judgments that are implicitly made when respellings are created by human authors. We approach ambiguity as a binary classification task. For any given syllable, we wish to determine whether it is ambiguous (a negative instance), or unambiguous (a positive instance). Our assumption is that a syllable will not be respelled unless it is necessary due to ambiguity. For each observed word-respelling pair, we take all syllables from the respelling as positive instances, and all syllables in the original word that are not preserved in the respelling as negative instances. For example, the pair consisting of the word *cec-il-y* respelled as *'sehs-il-ee'* provides three positive instances: *sehs, il* and *ee*; and two negative instances: *cec* and *y*.

We extracted word-respelling pairs from the Web-derived corpora of Ghoshal et al. (2009). The syllable breaks in the respellings were mapped onto the original words using ALINE. In order to improve the quality of the data, we applied a letter-to-phoneme model to both the original words and their respellings, and removed pairs with divergent pronunciations (computed as normalized edit distance ≤ 0.8). After the filtering, we were left a set of 25067 word-respelling pairs containing 78411 training syllables, which yielded 47270 positive and 31141 negative instances.

For the classification task we utilize the SVM-light software package (Joachims, 1999). Each instance is represented by a set of binary indicator features. The features correspond to character n-grams (including syllable boundary markers) with the values of n ranging from 1 to 5. For example, the syllable *-il-* turns on the following features: *i, l, -i, il, l-, -il, il-, -il-*. The model learns which *n*-grams are characteristic of ambiguous or unambiguous syllables. For example, it classifies both *le* and *li* as am-

biguous, and *lee* as unambiguous. Apart from the binary classification, the classifier also provides a real-valued score for each syllable.

6.4 Lexical Reviser

Since the use of familiar English letter sequences makes the respellings easier to interpret (Fraser, 1997), we incorporate dictionary lookup (Section 4.2) into our system. When the pronunciation of a syllable happens to correspond to the pronunciation an actual dictionary word, the syllable may be respelled using that word. This is done as the final step in the generation process because dictionary words often receive poor scores from the SVM classifier on the account of their n-gram composition. The lexical reviser is restricted to optionally improving the top-ranked word respelling candidate as determined by the SVM classifier without altering its syllabification. For example, the respelling *'surr-sin-uss'* of *circinus* is modified to *'sir-sin-us'*. If more than one word can be used, we let the SVM classifier select the least ambiguous one.

7 System Overview

Our respelling generation system is a multi-stage process. The input is a sequence of phonemes representing the pronunciation of the word. We start by identifying acceptable syllabifications of phonemes as described in Section 5.1. For each syllable, we take up to five respelling candidates produced by the P2L model (Section 5.2), and between one and three candidates proposed by the context-sensitive respeller (Section 5.3). The next stage involves filtering the candidate respellings with the L2P model (Section 6.1), and the vowel counter (Section 6.2). If all candidates happen to be rejected, we retain the first output of the context-sensitive respeller as the default. The candidate respellings are then scored by the SVM model (Section 6.3). At this point the syllables are combined into word respellings, which are ranked according to their syllable score average. Finally, the lexical reviser described in Section 6.4 is applied to the top candidate in an attempt to further improve the result.

8 Evaluation

In this section, after describing our test sets, we present the results of two evaluation experiments: direct human judgment, and indirect validation with an L2P system.

8.1 Test Sets

Our two test sets were defined after the development of our system had been completed. There is no overlap between the test sets and any of our training sets. The first test set consists of 27 out of 30 words compiled by Fraser (1997) — 3 words from the original set were excluded because the corresponding respellings assume a non-rhotic variety of English. We refer to Fraser's respellings as *expert*, and consider them as the upper bound in terms of quality.

The second test set of 231 words (henceforth referred to as the *Web set*) was extracted from the corpus of Ghoshal et al. (2009) after performing additional data clean-up described in Section 6.3. We identified a subset of words for which we could find phonetic transcriptions composed of English phonemes on Wikipedia. In order to ensure that the respellings and the corresponding transcriptions reflect the same pronunciation, we adapted the Soundex algorithm to apply to phonetic transcriptions, and retained only the respelling/transcription pairs that yielded identical Soundex codes. We removed words that are found in the Combilex dictionary as those could be familiar to human judges. Since longer words are more challenging to respell, and more likely to exhibit variation in respellings from different sources, we retained only words containing at least eight phonemes.

8.2 Human Judgment

We conducted an experiment with human evaluators using a specially developed graphical annotation program with synthesized word pronunciations. The evaluators were students enrolled in an introductory linguistic course, who were not involved in our project. 13 out of 20 evaluators declared themselves as native speakers of English.

The evaluation process involves 40 randomly selected words: 10 from Fraser's set, and 30 from the Web set. For each word, the program displays in a random sequence three respellings, which are from

Source	Web set		Fraser's set	
	U	U&C	U	U&C
Baseline	43.0	25.5	41.0	20.0
Web	68.0	32.6	—	
Expert	—		72.0	46.0
Our system	70.0	41.3	67.0	38.5

Table 2: Human judgments on respellings in %: U - unambiguous; U&C - unambiguous & correct.

the following sources: (1) the Baseline approach described in Section 4.3, (2) our system, and (3) either expert design (for Fraser's set) or the Web (for the Web set). In order to reduce bias, the original spelling of the word is not shown. Each respelling is judged separately with regards to ambiguity, and those that are judged ambiguous are removed from further consideration. Next, an audio clip synthesized from the phonemic sequence representing the intended pronunciation is played through headphones. For each of the remaining respellings, the evaluators decide whether it is correct with respect to the recorded pronunciation. Finally, if more than one respelling have been judged both unambiguous and correct, the evaluators are asked to identify the one that they prefer.

The results of the experiment are shown in Table 2. Our system significantly outperforms both Web respellings and the Baseline approach in terms of unambiguity and correctness. In addition, the respellings produced by our system are more likely to be preferred over the Web respellings, and more than twice as likely to be preferred over the baseline respellings than vice versa. The results on the small Fraser's set are less conclusive, but suggest that in terms of overall quality our system is much closer to the upper bound than to the baseline.

8.3 Automated Appraisal

Human evaluation is expensive and limited in terms of the number of variant respellings. Moreover, human judgements may be biased by previously seen respellings or by the familiarity with the standard spelling of a word. An automated evaluation is much less constrained, and facilitates an ablation study to determine the relative importance of various components of our system.

Source	Web set		Fraser's set	
	WA	PA	WA	PA
No respelling	13.0	76.2	14.8	76.3
Baseline	8.2	78.9	7.4	71.0
Web	14.3	77.9	—	
Expert	—		37.0	85.6
Our system	58.0	93.0	70.4	95.6

Table 4: Word accuracy (WA) and phoneme accuracy (PA) of *eSpeak* on respellings.

Source	Web set	
	WA	PA
Full system	58.0	93.0
w/o lexical reviser	57.6	93.1
w/o context-sensitive respeller	56.7	92.8
w/o P2L generator	51.9	92.1
w/o L2P correctness filter	33.8	88.0
w/o syllable breaks	20.8	83.9

Table 5: Accuracy of *eSpeak* on respellings produced by variants of our system.

eSpeak is a publicly available speech synthesizer[5] that can also convert text into phonemic sequences. The letter-to-phoneme component for English utilizes about five thousand rules, and a dictionary of about three thousand words, names, and abbreviations. In our evaluation, we treat *eSpeak* as a "black box" which translates a respelling into its most likely pronunciation. By determining if there is a match between the output of *eSpeak* and the intended pronunciation, we directly test the correctness of the respelling, and indirectly also its ambiguity.

The results of the automated evaluation are shown in Table 4. The accuracy on the original orthography is low, which is unsurprising since the test sets contain mostly rare, unusually spelled words. Neither the baseline nor the Web respellings are significantly easier for *eSpeak* than the original words. On the other hand, respellings generated by our system make a massive difference, boosting phoneme accuracy to well over 90% on both sets. They are also significantly more effective than the expert respellings.

Table 5 shows the results of our system on the

[5] `http://espeak.sourceforge.net`

No.	Spelling	IPA	Web/HF respelling	Score	System respelling	Score
1	*Incirlik*	[inʤirlik]	*injirlik*	1/6	*een-jeer-leek*	4/6
2	*Captopril*	[kæptəprɪl]	*kap-toh-pril*	1/6	*cap-tuh-prill*	4/6
3	*Coquitlam*	[kokwɪtləm]	*ko-kwit-lam*	1/6	*koh-quit-lumb*	4/6
4	*Karolina*	[kɑrəlinɑ]	*karo-leena*	4/6	*car-awl-ee-nah*	1/6
5	*subluxation*	[səbləkseʃən]	*sub-luck-say-shun*	3/5	*suh-bluck-say-shun*	1/5
6	*swingle*	[swɪŋgəl]	*swing-gl*	0/5	*swing-gull*	2/5
7	*cockatrice*	[kɑkətraɪs]	*kok-a-trice*	0/7	*cock-uh-trice*	4/7
8	*recalesce*	[rikəlɛs]	*ree-ka-less*	1/5	*re-cull-ess*	3/5
9	*jongleur*	[ʒɑŋglər]	*jong-gler*	7/9	*zhahng-gler*	0/9
10	*ylang-ylang*	[ilæŋilæŋ]	*ee-lang-ee-lang*	5/5	*eel-ang-eel-ang*	1/5

Table 3: Examples of respellings.

Web set with various modules disabled, which provides an estimate of their importance. Neither the context-sensitive respeller nor dictionary lookup seem to contribute much to *eSpeak*'s performance. On the other hand, disabling the P2L generator produces a significant drop in word accuracy, while removing the L2P correctness filter almost doubles the phoneme error rate. Interestingly, removing syllable breaks from the output of the full system has an even greater negative impact.

8.4 Analysis

Each of 20 evaluators judged 3 variant respellings of 40 different words. The average number of judgments per word was 7.4 for the 27 words in Fraser's set, and 2.8 for the 212 words in the Web set (due to random selection, 19 words from the Web set were not judged). Table 3 shows examples of respellings that were judged by at least five evaluators. The score columns indicate the proportion of the evaluators that judged a particular respellings as unambiguous and correct. The baseline respellings are not included as their scores were rarely higher than the scores of the other respellings for a given word. An interesting exception is *palimpsest*, for which the baseline respelling is identical to the actual spelling of the word.

Examples 1-5 in Table 3 come from the Web set, while examples 6-10 are from Fraser's set. The low scores of the first three Web respellings can be attributed to specific letter-to-phoneme mappings: [i]→*i*, [ə]→*oh*, and [ə]→*a*. Each of the examples 3-5 indicate the evaluators' acceptance of a particular respelling device: silent letters, multi-syllable units, and dictionary words. In examples 6-8, the syllables immediately after the first hyphen in Helen Fraser's respellings seem to be problematic. The expert respelling of *jongleur* is considered correct even though the initial *j* suggests [ʤ], not [ʒ]. Finally, the last example demonstrates that the hyphenation choice can result in very different judgments.

9 Conclusion

In this paper, we introduced the task of automatically generating respellings from the given pronunciation. We investigated the characteristics of good respellings, and discussed three direct methods of their creation. We proposed a system that combines supervised and unsupervised learning with phonetic and orthographic principles. The evaluation experiment involving human participants indicates that the respellings produced by our system are better on average than those found on the Web. The automated verification demonstrates that they are also much easier to interpret for a rule-based text-to-speech converter. In the future we plan to address the related tasks of improving existing respellings, and assisting writers in creating respellings without direct access to the phonemic representations.

Acknowledgements

We thank Aditya Bhargava and Clarke Chomyc for their contribution to the creation of data sets, and Ben Tucker for advice on the complexities of the human evaluation experiment. This research was partially funded by the Natural Sciences and Engineering Research Council of Canada.

References

Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. 2009. On the syllabification of phonemes. In *Proc. of HLT-NAACL*, pages 308–316.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Doğan Can, Erica Cooper, Arnab Ghoshal, Martin Jansche, Sanjeev Khudanpur, Bhuvana Ramabhadran, Michael Riley, Murat Saraçlar, Abhinav Sethy, Morgan Ulinski, and Christopher White. 2009. Web derived pronunciations for spoken term detection. In *Proc. of ACM SIGIR*, pages 83–90.

Edward Carney. 1994. *A Survey of English Spelling*. Routledge.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper & Row.

Helen Fraser. 1997. Dictionary pronunciation guides for English. *International Journal of Lexicography*, 10(3):181–208.

Lucian Galescu. 2009. Extending pronunciation lexicons via non-phonemic respellings. In *Proc. of HLT-NAACL: Short Papers*, pages 129–132.

Arnab Ghoshal, Martin Jansche, Sanjeev Khudanpur, Michael Riley, and Morgan Ulinski. 2009. Web-derived pronunciations. In *Proc. of ICASSP*, pages 4289–4292.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2010. Letter-phoneme alignment: An exploration. In *Proc. of ACL*, pages 780–788.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Proc. of HLT-NAACL*, pages 372–379.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proc. of ACL*, pages 905–913.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schalkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Michael Kenstowicz. 1994. *Phonology in Generative Grammar*. Blackwell.

John Kominek and Alan W Black. 2006. Learning pronunciation dictionaries: Language complexity and word selection strategies. In *Proc. of HLT-NAACL*, pages 232–239.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proc. of NAACL*, pages 288–295.

John Noble. 2012. *Mexico*. Lonely Planet, 13th edition.

Korin Richmond, Robert Clark, and Sue Fitt. 2009. Robust LTS rules with the Combilex speech technology lexicon. In *Proc. of Interspeech*, pages 1295–1298.

Henry Rogers. 2000. *The Sounds of Language*. Pearson.

Briony Williams and Rhys James Jones. 2008. Acquiring pronunciation data for a placenames lexicon in a less-resourced language. In *Proc. of LREC*.