Pronunciation Modeling in Spelling Correction for Writers of English as a Foreign Language

Adriane Boyd
Department of Linguistics
The Ohio State University
1712 Neil Avenue
Columbus, Ohio 43210, USA
adriane@ling.osu.edu

Abstract

We propose a method for modeling pronunciation variation in the context of spell checking for non-native writers of English. Spell checkers, typically developed for native speakers, fail to address many of the types of spelling errors peculiar to non-native speakers, especially those errors influenced by differences in phonology. Our model of pronunciation variation is used to extend a pronouncing dictionary for use in the spelling correction algorithm developed by Toutanova and Moore (2002), which includes models for both orthography and pronunciation. The pronunciation variation modeling is shown to improve performance for misspellings produced by Japanese writers of English.

1 Introduction

Spell checkers identify misspellings, select appropriate words as suggested corrections, and rank the suggested corrections so that the likely intended word is high in the list. Since traditional spell checkers have been developed with competent native speakers as the target users, they do not appropriately address many types of errors made by non-native writers and they often fail to suggest the appropriate corrections. Non-native writers of English struggle with many of the same idiosyncrasies of English spelling that cause difficulty for native speakers, but differences between English phonology and the phonology of their native language lead to types of spelling errors not anticipated by traditional spell checkers (Okada, 2004; Mitton and Okada, 2007).

Okada (2004) and Mitton and Okada (2007) investigate spelling errors made by Japanese writers of English as a foreign language (JWEFL). Okada (2004) identifies two main sources of errors for JWEFL: differences between English and Japanese phonology and differences between the English alphabet and the Japanese *romazi* writing system, which uses a subset of English letters. Phonological differences result in number of distinctions in English that are not present in Japanese and *romazi* causes difficulties for JWEFL because the Latin letters correspond to very different sounds in Japanese.

We propose a method for creating a model of pronunciation variation from a phonetically untranscribed corpus of read speech recorded by non-native speakers. The pronunciation variation model is used to generate multiple pronunciations for each canonical pronunciation in a pronouncing dictionary and these variations are used in the spelling correction approach developed by Toutanova and Moore (2002), which uses statistical models of spelling errors that consider both orthography and pronunciation. Several conventions are used throughout this paper: a *word* is a sequence of characters from the given alphabet found in the word list. A *word list* is a list of words. A *misspelling*, marked with *, is a sequence of characters not found in the word list. A *candidate correction* is a word from the word list proposed as a potential correction.

2 Background

Research in spell checking (see Kukich, 1992, for a survey of spell checking research) has focused on three main problems: non-word error detection, isolated-word error correction, and context-dependent word correction. We focus on the first two tasks. A non-word is a sequence of letters that

Proceedings of the NAACL HLT Student Research Workshop and Doctoral Consortium, pages 31–36,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

is not a possible word in the language in any context, e.g., English *thier. Once a sequence of letters has been determined to be a non-word, isolated-word error correction is the process of determining the appropriate word to substitute for the non-word.

Given a sequence of letters, there are thus two main subtasks: 1) determine whether this is a non-word, 2) if so, select and rank candidate words as potential corrections to present to the writer. The first subtask can be accomplished by searching for the sequence of letters in a word list. The second subtask can be stated as follows (Brill and Moore, 2000): Given an alphabet Σ, a word list D of strings $\in \Sigma^*$, and a string $r \notin D$ and $\in \Sigma^*$, find $w \in D$ such that w is the most likely correction. Minimum edit distance is used to select the most likely candidate corrections. The general idea is that a minimum number of edit operations such as insertion and substitution are needed to convert the misspelling into a word. Words requiring the smallest numbers of edit operations are selected as the candidate corrections.

2.1 Edit Operations and Edit Weights

In recent spelling correction approaches, edit operations have been extended beyond single character edits and the methods for calculating edit operation weights have become more sophisticated. The spelling error model proposed by Brill and Moore (2000) allows generic string edit operations up to a certain length. Each edit operation also has an associated probability that improves the ranking of candidate corrections by modeling how likely particular edits are. Brill and Moore (2000) estimate the probability of each edit from a corpus of spelling errors. Toutanova and Moore (2002) extend Brill and Moore (2000) to consider edits over both letter sequences and sequences of phones in the pronunciations of the word and misspelling. They show that including pronunciation information improves performance as compared to Brill and Moore (2000).

2.2 Noisy Channel Spelling Correction

The spelling correction models from Brill and Moore (2000) and Toutanova and Moore (2002) use the noisy channel model approach to determine the types and weights of edit operations. The idea behind this approach is that a writer starts out with the intended word w in mind, but as it is being written the word passes through a noisy channel resulting in the observed non-word r. In order to determine how likely a candidate correction is, the spelling correction model determines the probability that the word w was the intended word given the misspelling r: $P(w|r)$. To find the best correction, the word w is found for which $P(w|r)$ is maximized: $argmax_w\ P(w|r)$. Applying Bayes' Rule and discarding the normalizing constant $P(r)$ gives the correction model:

$$argmax_w\ P(w|r) = argmax_w\ P(w)P(r|w)$$

$P(w)$, how probable the word w is overall, and $P(r|w)$, how probable it is for a writer intending to write w to output r, can be estimated from corpora containing misspellings. In the following experiments, $P(w)$ is assumed be equal for all words to focus this work on estimating the error model $P(r|w)$ for JWEFL.[1]

Brill and Moore (2000) allow all edit operations $\alpha \rightarrow \beta$ where Σ is the alphabet and $\alpha, \beta \in \Sigma^*$, with a constraint on the length of α and β. In order to consider all ways that a word w may generate r with the possibility that any, possibly empty, substring α of w becomes any, possibly empty, substring β of r, it is necessary to consider all ways that w and r may be partitioned into substrings. This error model over letters, called P_L, is approximated by Brill and Moore (2000) as shown in Figure 1 by considering only the pair of partitions of w and r with the maximum product of the probabilities of individual substitutions. $Part(w)$ is all possible partitions of w, $|R|$ is number of segments in a particular partition, and R_i is the i^{th} segment of the partition.

The parameters for $P_L(r|w)$ are estimated from a corpus of pairs of misspellings and target words. The method, which is described in detail in Brill and Moore (2000), involves aligning the letters in pairs of words and misspellings, expanding each alignment with up to N neighboring alignments, and calculating the probability of each $\alpha \rightarrow \beta$ alignment. Since we will be using a training corpus that consists solely of pairs of misspellings and words (see section 3), we would have lower probabilities for

[1] Of course, $P(w)$ is not equal for all words, but it is not possible to estimate it from the available training corpus, the Atsuo-Henry Corpus (Okada, 2004), because it contains only pairs of words and misspellings for around 1,000 target words.

$$P_L(r|w) \approx max_{R \in Part(r), T \in Part(w)} \prod_{i=1}^{|R|} P(R_i \rightarrow T_i)$$

$$P_{PHL}(r|w) \approx \sum_{pron_w} \frac{1}{|pron_w|} \max_{pron_r} P_{PH}(pron_w|pron_r)P(pron_r|r)$$

Figure 1: Approximations of P_L from Brill and Moore (2000) and P_{PHL} from Toutanova and Moore (2002)

$\alpha \rightarrow \alpha$ than would be found in a corpus with misspellings observed in context with correct words. To compensate, we approximate $P(\alpha \rightarrow \alpha)$ by assigning it a minimum probability m:

$$P(\alpha \rightarrow \beta) = \begin{cases} m + (1-m)\frac{count(\alpha \rightarrow \beta)}{count(\alpha)} & \text{if } \alpha = \beta \\ (1-m)\frac{count(\alpha \rightarrow \beta)}{count(\alpha)} & \text{if } \alpha \neq \beta \end{cases}$$

2.2.1 Extending to Pronunciation

Toutanova and Moore (2002) describe an extension to Brill and Moore (2000) where the same noisy channel error model is used to model phone sequences instead of letter sequences. Instead of the word w and the non-word r, the error model considers the pronunciation of the non-word r, $pron_r$, and the pronunciation of the word w, $pron_w$. The error model over phone sequences, called P_{PH}, is just like P_L shown in Figure 1 except that r and w are replaced with their pronunciations. The model is trained like P_L using alignments between phones.

Since a spelling correction model needs to rank candidate words rather than candidate pronunciations, Toutanova and Moore (2002) derive an error model that determines the probability that a word w was spelled as the non-word r based on their pronunciations. Their approximation of this model, called P_{PHL}, is also shown in Figure 1. $P_{PH}(pron_w|pron_r)$ is the phone error model described above and $P(pron_r|r)$ is provided by the letter-to-phone model described below.

2.3 Letter-To-Phone Model

A letter-to-phone (LTP) model is needed to predict the pronunciation of misspellings for P_{PHL}, since they are not found in a pronouncing dictionary. Like Toutanova and Moore (2002), we use the n-gram LTP model from Fisher (1999) to predict these pronunciations. The n-gram LTP model predicts the pronunciation of each letter in a word considering up to four letters of context to the left and right. The most specific context found for each letter and its context in the training data is used to predict the pronunciation of a word. We extended the prediction step to consider the most probable phone for the top M most specific contexts.

We implemented the LTP algorithm and trained and evaluated it using pronunciations from CMUDICT. A training corpus was created by pairing the words from the size 70 CMUDICT-filtered SCOWL word list (see section 3) with their pronunciations. This list of approximately 62,000 words was split into a training set with 80% of entries and a test set with the remaining 20%. We found that the best performance is seen when $M = 3$, giving 95.5% phone accuracy and 74.9% word accuracy.

2.4 Calculating Final Scores

For a misspelling r and a candidate correction w, the letter model P_L gives the probability that w was written as r due to the noisy channel taking into account only the orthography. P_{PH} does the same for the pronunciations of r and w, giving the probability that $pron_w$ was output was $pron_r$. The pronunciation model P_{PHL} relates the pronunciations modeled by P_{PH} to the orthography in order to give the probability that r was written as w based on pronunciation. P_L and P_{PHL} are then combined as follows to calculate a score for each candidate correction.

$$S_{CMB}(r|w) = logP_L(r|w) + \lambda logP_{PHL}(r|w)$$

3 Resources and Data Preparation

Our spelling correction approach, which includes error models for both orthography and pronunciation (see section 2.2) and which considers pronunciation variation for JWEFL requires a number of resources: 1) spoken corpora of American English (TIMIT, TIMIT 1991) and Japanese English (ERJ, see below) are used to model pronunciation variation, 2) a pronunciation dictionary (CMUDICT, CMUDICT 1998) provides American English pronunciations for the target words, 3) a corpus of spelling errors made by JWEFL (Atsuo-Henry Corpus, see below) is used to train spelling error models and test the spell checker's performance, and 4) Spell Checker Oriented Word Lists (SCOWL, see below) are adapted for our use.

The **English Read by Japanese Corpus** (Minematsu et al., 2002) consists of 70,000 prompts containing phonemic and prosodic cues recorded by 200 native Japanese speakers with varying English competence. See Minematsu et al. (2002) for details on the construction of the corpus.

The **Atsuo-Henry Corpus** (Okada, 2004) includes a corpus of spelling errors made by JWEFL that consists of a collection of spelling errors from multiple corpora.[2] For use with our spell checker, the corpus has been cleaned up and modified to fit our task, resulting in 4,769 unique misspellings of 1,046 target words. The data is divided into training (80%), development (10%), and test (10%) sets.

For our word lists, we use adapted versions of the **Spell Checker Oriented Word Lists**.[3] The size 50 word lists are used in order to create a general purpose word list that covers all the target words from the Atsuo-Henry Corpus. Since the target pronunciation of each item is needed for the pronunciation model, the word list was filtered to remove words whose pronunciation is not in CMUDICT. After filtering, the word list contains 54,001 words.

4 Method

This section presents our method for modeling pronunciation variation from a phonetically untranscribed corpus of read speech. The pronunciation-based spelling correction approach developed in Toutanova and Moore (2002) requires a list of possible pronunciations in order to compare the pronunciation of the misspelling to the pronunciation of correct words. To account for target pronunciations specific to Japanese speakers, we observe the pronunciation variation in the ERJ and generate additional pronunciations for each word in the word list. Since the ERJ is not transcribed, we begin by adapting a recognizer trained on native English speech. First, the ERJ is recognized using a monophone recognizer trained on TIMIT. Next, the most frequent variations between the canonical and recognized pronunciations are used to adapt the recognizer. The adapted recognizer is then used to recognize the ERJ in forced alignment with the canonical pronunciations. Finally, the variations from the previous step are used to create models of pronunciation variation for each phone, which are used to generate multiple pronunciations for each word.

4.1 Initial Recognizer

A monophone speech recognizer was trained on all TIMIT data using the Hidden Markov Model Toolkit (HTK).[4] This recognizer is used to generate a phone string for each utterance in the ERJ. Each recognized phone string is then aligned with the canonical pronunciation provided to the speakers. Correct alignments and substitutions are considered with no context and insertions are conditioned on the previous phone. Due to restrictions in HTK, deletions are currently ignored.

The frequency of phone alignments for all utterances in the ERJ are calculated. Because of the low phone accuracy of monophone recognizers, especially on non-native speech, alignments are observed between nearly all pairs of phones. In order to focus on the most frequent alignments common to multiple speakers and utterances, any alignment observed less than 20% as often as the most frequent alignment for that canonical phone is discarded, which results in an average of three variants of each phone.[5]

4.2 Adapting the Recognizer

Now that we have probability distributions over observed phones, the HMMs trained on TIMIT are modified as follows to allow the observed variation. To allow, for instance, variation between `p` and `th`, the states for `th` from the original recognizer are inserted into the model for `p` as a separate path. The resulting phone model is shown in Figure 2. The transition probabilities into the first states

[2] Some of the spelling errors come from an elicitation task, so the distribution of target words is not representative of typical JWEFL productions, e.g., the corpus contains 102 different misspellings of `albatross`.

[3] SCOWL is available at http://wordlist.sourceforge.net.

[4] HTK is available at http://htk.eng.cam.ac.uk.

[5] There are 119 variants of 39 phones. The cutoff of 20% was chosen to allow a few variations for most phones. A small number of phones have no variants (e.g., `iy`, `w`) while a few have over nine variants (e.g., `ah`, `l`). It is not surprising that phones that are well-known to be difficult for Japanese speakers (cf. Minematsu et al., 2002) are the ones with the most variation.

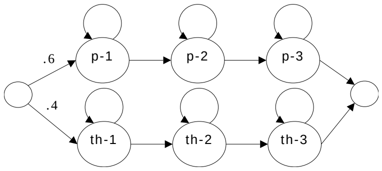

Figure 2: Adapted phone model for p accounting for variation between p and th

of the phones come from the probability distribution observed in the initial recognition step. The transition probabilities between the three states for each variant phone remain unchanged. All HMMs are adapted in this manner using the probability distributions from the initial recognition step.

The adapted HMMs are used to recognize the ERJ Corpus for a second time, this time in forced alignment with the canonical pronunciations. The state transitions indicate which variant of each phone was recognized and the correspondences between the canonical phones and recognized phones are used to generate a new probability distribution over observed phones for each canonical phone. These are used to find the most probable pronunciation variations for a native-speaker pronouncing dictionary.

4.3 Generating Pronunciations

The observed phone variation is used to generate multiple pronunciations for each pronunciation in the word list. The OpenFst Library[6] is used to find the most probable pronunciations in each case. First, FSTs are created for each phone using the probability distributions from the previous section. Next, an FST is created for the entire word by concatenating the FSTs for the pronunciation from CMUDICT. The pronunciations corresponding to the best n paths through the FST and the original canonical pronunciation become possible pronunciations in the extended pronouncing dictionary. The size 50 word list contains 54,001 words and when expanded to contain the top five variations of each pronunciation, there are 255,827 unique pronunciations.

5 Results

In order to evaluate the effect of pronunciation variation in Toutanova and Moore (2002)'s spelling correction approach, we compare the performance of the pronunciation model and the combined model with and without pronunciation variation.

We implemented the letter and pronunciation spelling correction models as described in section 2.2. The letter error model P_L and the phone error model P_{PH} are trained on the training set. The development set is used to tune the parameters introduced in previous sections.[7] In order to rank the words as candidate corrections for a misspelling r, $P_L(r|w)$ and $P_{PHL}(r|w)$ are calculated for each word in the word list using the algorithm described in Brill and Moore (2000). Finally, P_L and P_{PHL} are combined using S_{CMB} to rank each word.

5.1 Baseline

The open source spell checker *GNU Aspell*[8] is used to determine the baseline performance of a traditional spell checker using the same word list. An *Aspell* dictionary was created with the word list described in section 3. *Aspell*'s performance is shown in Table 1. The 1-Best performance is the percentage of test items for which the target word was the first candidate correction, 2-Best is the percentage for which the target was in the top two, etc.

5.2 Evaluation of Pronunciation Variation

The effect of introducing pronunciation variation using the method described in section 4 can be evaluated by examining the performance on the test set for P_{PHL} with and without the additional variations. The results in Table 1 show that the addition of pronunciation variations does indeed improve the performance of P_{PHL} across the board. The 1-Best, 3-Best, and 4-Best cases for P_{PHL} with variation show significant improvement ($p<0.05$) over P_{PHL} without variation.

5.3 Evaluation of the Combined Model

We evaluated the effect of including pronunciation variation in the combined model by comparing the performance of the combined model with and without pronunciation variation, see results in Table 1. Despite the improvements seen in P_{PHL} with pronunciation variation, there are no significant differences between the results for the combined model with and without variation. The combined model

[6] OpenFst is available at http://www.openfst.org/.

[7] The values are: $N = 3$ for the letter model, $N = 4$ for the phone model, $m = 80\%$, and $\lambda = 0.15$ in S_{CMB}.

[8] GNU Aspell is available at http://aspell.net.

Model	1-Best	2-Best	3-Best	4-Best	5-Best	6-Best
Aspell	44.1	54.0	64.1	68.3	70.0	72.5
Letter (L)	64.7	74.6	79.6	83.2	84.0	85.3
Pronunciation (PHL) without Pron. Var.	47.9	60.7	67.9	70.8	75.0	77.3
Pronunciation (PHL) with Pron. Var.	50.6	62.2	70.4	73.1	76.7	78.2
Combined (CMB) without Pron. Var.	64.9	75.2	78.6	81.1	82.6	83.2
Combined (CMB) with Pron. Var.	65.5	75.0	78.4	80.7	82.6	84.0

Table 1: Percentage of Correct Suggestions on the Atsuo-Henry Corpus Test Set for All Models

Rank	Aspell	L	PHL	CMB
1	enemy	enemy	**any**	enemy
2	envy	envy	Emmy	envy
3	energy	money	Ne	**any**
4	eye	emery	gunny	deny
5	teeny	deny	ebony	money
6	Ne	**any**	anything	emery
7	deny	nay	senna	nay
8	**any**	ivy	journey	ivy

Table 2: Misspelling `*eney`, Intended Word `any`

with variation is also not significantly different from the letter model P_L except for the drop in the 4-Best case.

To illustrate the performance of each model, the ranked lists in Table 2 give an example of the candidate corrections for the misspelling of `any` as `*eney`. *Aspell* preserves the initial letter of the misspelling and vowels in many of its candidates. P_L's top candidates also overlap a great deal in orthography, but there is more initial letter and vowel variation. As we would predict, P_{PHL} ranks `any` as the top correction, but some of the lower-ranked candidates for P_{PHL} differ greatly in length.

5.4 Summary of Results

The noisy channel spelling correction approach developed by Brill and Moore (2000) and Toutanova and Moore (2002) appears well-suited for writers of English as a foreign language. The letter and combined models outperform the traditional spell checker *Aspell* by a wide margin. Although including pronunciation variation does not improve the combined model, it leads to significant improvements in the pronunciation-based model P_{PHL}.

6 Conclusion

We have presented a method for modeling pronunciation variation from a phonetically untranscribed corpus of read non-native speech by adapting a monophone recognizer initially trained on native speech. This model allows a native pronouncing dictionary to be extended to include non-native pronunciation variations. We incorporated a pronouncing dictionary extended for Japanese writers of English into the spelling correction model developed by Toutanova and Moore (2002), which combines orthography-based and pronunciation-based models. Although the extended pronunciation dictionary does not lead to improvement in the combined model, it does leads to significant improvement in the pronunciation-based model.

Acknowledgments

I would like to thank Eric Fosler-Lussier, the Ohio State computational linguistics discussion group, and anonymous reviewers for their helpful feedback.

References

Brill, Eric and Robert C. Moore (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of ACL 2000*.

CMUDICT (1998). CMU Pronouncing Dictionary version 0.6. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Fisher, Willam (1999). A statistical text-to-phone function using ngrams and rules. In *Proceedings of ICASSP 1999*.

Kukich, Karen (1992). Technique for automatically correcting words in text. *ACM Computing Surveys* 24(4).

Minematsu, N., Y. Tomiyama, K. Yoshimoto, K. Shimizu, S. Nakagawa, M. Dantsuji, and S. Makino (2002). English Speech Database Read by Japanese Learners for CALL System Development. In *Proceedings of LREC 2002*.

Mitton, Roger and Takeshi Okada (2007). The adaptation of an English spellchecker for Japanese writers. In *Symposium on Second Language Writing*.

Okada, Takeshi (2004). A Corpus Analysis of Spelling Errors Made by Japanese EFL Writers. *Yamagata English Studies* 9.

TIMIT (1991). TIMIT Acoustic-Phonetic Continuous Speech Corpus. NIST Speech Disc CD1-1.1.

Toutanova, Kristina and Robert Moore (2002). Pronunciation Modeling for Improved Spelling Correction. In *Proceedings of ACL 2002*.