ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Evaluation of Many-to-Many Alignment Algorithm by Automatic Pronunciation Annotation Using Web Text Mining

Keigo Kubo[1], *Hiromichi Kawanami*[1], *Hiroshi Saruwatari*[1], *Kiyohiro Shikano*[1]

[1]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
keigo-k@is.naist.jp, kawanami@is.naist.jp, sawatari@is.naist.jp, shikano@is.naist.jp

Abstract

The need for robust pronunciation annotation over out-of-vocabulary (OOV) words has been increasing with the development of an application that deals with proper nouns and brand-new words, such as Voice Search. In robust pronunciation annotation over OOV words, the alignment between graphemes and phonemes is vital data. For a many-to-many alignment algorithm between graphemes and phonemes, we describe its problems and methods to overcome them. An evaluation experiment of a many-to-many alignment by automatic pronunciation annotation using Web text mining is also performed. That experimental result shows that the proposed many-to-many alignment produces an alignment that has the high generalization ability for OOV words while avoiding degradation of the accuracy of the pronunciation annotation compared with the conventional approach.

Index Terms: string alignment, out-of-vocabulary word, pronunciation annotation

1. Introduction

Recent advances in speech recognition have made it possible to attempt large-scale, open-domain, data-driven approaches. Out-of-vocabulary (OOV) words are the bottleneck in speech systems, and the need for robust pronunciation annotation has been increasing. For example, voice search applications have attracted attention because of an increased demand for mobile device interfaces. A variety of words, such as proper nouns and brand-new words, must be dealt with in these applications. It is important to update the language model and the word dictionary to accommodate OOV terms. OOV words can be collected easily from Web text resources, but generally, their pronunciation remains unknown. Therefore, an automatic pronunciation annotation is desired. Statistical approaches, including grapheme-to-phoneme (g2p) conversion[1-3] and a Web text mining approach that identifies a part of the Web text that describes word-pronunciation pairs[4], have been proposed.

The alignment between graphemes and phonemes is vital data for these pronunciation annotation methods. In this paper, we focus on alignment methods, such as a one-to-one alignment[5] and a many-to-many alignment[6-8]. In [6,7], the many-to-many alignment is referred to as a joint multigram. As this alignment and the one proposed in [8] are essentially the same, the two methods are treated as a joint multigram approach in this paper. [8] explains the suitability of the joint multigram approach over a one-to-one alignment and demonstrates the better performance of this approach. However, the joint multigram approach generally prefers a mapping consisting of longer substrings, which degrades the generalization ability of automatic pronunciation annotation for OOV words. To cope with this problem, we introduce the city block distance, which is employed in dynamic time warping, to the joint multigram approach. The resulting mappings are pairs of substrings that are unconstrained in length, yet sufficiently short to increase the generalization ability[9]. Our many-to-many alignment has been shown to be effective as a g2p conversion for OOV words.

For our many-to-many alignment, we describe the remaining problems and propose methods to overcome them. Also, an evaluation experiment of these methods by automatic pronunciation annotation using Web text mining is shown.

The rest of this paper is organized as follows. In Section 2, we explain many-to-many alignment methods. The parameter estimation for our many-to-many alignment is described in Section 3, and the problems and their proposed solutions are described in Section 4. In Section 5, we report the evaluation experiment by automatic pronunciation annotation using Web text mining. Finally, Section 6 states our conclusion.

2. Many-to-many alignment

2.1. Preliminaries

Let d be a tuple of a word and its pronunciation, and D be a set of d tuples. Let $\boldsymbol{U}_d$ be a set of alignment candidates of graphemes and phonemes, generated from the d tuple. Let $\boldsymbol{u}$ be an alignment of the tuple d in the set $\boldsymbol{U}_d$, and u be a mapping in the alignment $\boldsymbol{u}$. We denote a mapping where ϵ is mapped in the phoneme side to be a deletion character. An example of a word-pronunciation

pair $\langle able, \acute{e}ibl \rangle$ is shown below.

$$
\begin{aligned}
d &= \langle able,\ \acute{e}ibl \rangle \\
D &= \{\langle able,\ \acute{e}ibl \rangle\} \\
\boldsymbol{U}_d &= \{able/\acute{e}ibl,\ abl/\acute{e}ib\ \ e/l,\ \ ...,\\
&\quad\ a/\acute{e}i\ \ b/b\ \ le/l,\ \ a/\acute{e}i\ \ b/b\ \ l/l\ \ e/\epsilon\} \\
\boldsymbol{u} &= a/\acute{e}i\ \ b/b\ \ l/l\ \ e/\epsilon \\
u &= a/\acute{e}i
\end{aligned}
$$

2.2. Joint multigram approach

The joint multigram approach proposed in [6] is

$$
\begin{aligned}
\hat{\boldsymbol{u}} &= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d) \\
&\simeq \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} P(u). \qquad (1)
\end{aligned}
$$

$P(\boldsymbol{u}|d)$ is assumed to be the product of the unigram probabilities of a mapping included in $\boldsymbol{u}$.

2.3. Basic idea of our many-to-many alignment

The joint multigram approach prefers longer mappings, i.e., mappings of a longer grapheme sequence and/or a longer phoneme sequence (e.g., $u = able/\acute{e}ibl$), which degrades the generalization ability of automatic pronunciation annotation for OOV words. To cope with this problem, the joint multigram approach limits the maximum length of graphemes and the maximum length of phonemes in a single mapping. These maximum lengths are both set to two in [8]. However, appropriate values of the parameters depend on the *language*. In the case of Japanese words including kanji (Chinese characters), one grapheme could be mapped to more than two phonemes. If we set these maximum lengths to be more than two, the resulting mappings would no longer be short.

In order to suppress longer mappings without fixing these maximum lengths, we introduce the city block distance in $P(\boldsymbol{u}|d)$ as an exponential[9] domain. As a consequence, longer mappings are not advantageous over shorter mappings because the difference in the number of multiplications in $P(\boldsymbol{u}|d)$, which causes the preference for longer mappings in the joint multigram approach, is equalized in each alignment $\boldsymbol{u}$. This leads to an improvement of the generalization ability of automatic pronunciation annotation for OOV words.

Let i_u be the number of characters in graphemes and j_u be the number of characters in phonemes of mapping u. Then, our many-to-many alignment is defined as

$$
\begin{aligned}
\hat{\boldsymbol{u}} &= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d) \\
&\simeq \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} P(u)^{s_u}, \qquad (2)
\end{aligned}
$$

where s_u means the city block distance as an exponential domain and is defined as

$$
s_u = i_u + j_u. \qquad (3)
$$

3. Parameter estimation

Our many-to-many alignment estimates model parameters $p_u \equiv P(u)$ by employing the EM algorithm along with the joint multigram approach. The following is a pseudocode for the EM algorithm of our many-to-many alignment.

1. Set an initial value of p_u.
2. Calculate an expectation (E-step),

$$
\gamma_u = \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} \frac{\prod_{u \in \boldsymbol{u}} p_u^{s_u}}{\sum_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} p_u^{s_u}} n_{\boldsymbol{u}}(u), \qquad (4)
$$

   where $n_{\boldsymbol{u}}(u)$ means a frequency of u in $\boldsymbol{u}$.
3. Calculate the maximum likelihood (M-step),

$$
\hat{p}_u = \frac{\gamma_u}{\sum_{u \in \boldsymbol{U}} \gamma_u}, \qquad (5)
$$

   where $\boldsymbol{U}$ is the set of all mapping types.
4. Substitute $\hat{p}_u$ for p_u.
5. Finish if convergent, otherwise return to Step 2.

The initial value of p_u is set uniformly. Our EM algorithm introduces the city block distance in (4), unlike the joint multigram approach.

4. Problems and their proposed solutions

4.1. Deletion character

The deletion character differs from a mapping in which pronunciation is mapped in terms of frequency. Whereas a mapping in which pronunciation is mapped is counted in (4) when only its graphemes and its phonemes appear for a d tuple concurrently, the deletion character is counted when its graphemes appear for a d tuple. This causes the problem that parameters for alignment are estimated such that an irrelevant deletion character is chosen well.

To solve the above problem, we introduce n-best Viterbi training[6] which employs only promising alignments $\overline{\boldsymbol{U}}_d$ instead of all possible alignments $\boldsymbol{U}_d$ in the E-step of the EM algorithm. An unpromising alignment includes many irrelevant deletion characters. Since the n-best Viterbi training removes such alignments from parameter estimation, the above problem is suppressed.

Initial values of parameters are estimated to acquire promising alignments by employing our EM algorithm described in Section 3. If the appearance of the deletion character is permitted in our EM algorithm, these initial values are estimated such that an irrelevant deletion character is chosen well. Hence, we propose to utilize our EM algorithm with prohibition of the appearance of the deletion character. Next, we perform the special n-best

Table 1: Example of irrelevant mapping in the case of ⟨*AAA*, *tríp*léi⟩ and ⟨*Ace*, *éis*⟩. In this case, the irrelevant mapping is *AA*/*trípl*.

⟨*AAA*, *tríp*léi⟩ → {···, ***AA*/*trípl*** *A*/*éi*, ···}
⟨*Ace*, *éis*⟩ → {···, *A*/*éi c*/*s e*/ϵ, ···}

Viterbi training that updates a parameter only once to estimate the initial values of parameters where the irrelevant deletion character is not chosen. This permits the appearance of the deletion character and employs parameters estimated using the above EM algorithm as initial values. The parameter of the deletion character is approximated by the geometric mean of parameters of mappings included in the alignment without the deletion character. Our parameter estimation introducing n-best Viterbi training is shown as follows.

1. Employ our EM algorithm with prohibition of the appearance of the deletion character.
2. Carry out the special n-best Viterbi training that requires an n-best alignment using (6) and updates the parameter only once. This permits the appearance of the deletion character and employs parameters trained in Step 1 as initial values.

$$\hat{\boldsymbol{u}} = \arg\max_{\boldsymbol{u}} \prod_{u \in \boldsymbol{u}'} P(u)^{s_u} \times \Big(\prod_{u \in \boldsymbol{u}'} P(u)^{s_u} \Big)^{\frac{D_{\boldsymbol{u}}}{I_{\boldsymbol{u}} + J_{\boldsymbol{u}} - D_{\boldsymbol{u}}}} \quad (6)$$

   $\boldsymbol{u}'$ is an alignment that removes the deletion character from $\boldsymbol{u}$, $D_{\boldsymbol{u}}$ is the total number of characters of the deletion character, $I_{\boldsymbol{u}}$ is the total number of characters of the word, $J_{\boldsymbol{u}}$ is the total number of characters of the pronunciation.
3. Carry out the n-best Viterbi training that requires an n-best alignment using (2) permitting the appearance of the deletion character. Initial values of parameters are the same as for parameters trained in Step 2.

4.2. Irrelevant mapping

An alignment including an irrelevant mapping tends to be chosen in a word that has a special pronunciation. For example, as shown in Table 1, the parameter of the mapping *A*/*éi* is estimated highly because it appears for other aligned data such as ⟨*Ace*, *éis*⟩ except for the part of ⟨*AAA*, *tríp*léi⟩. Therefore, the alignment including the irrelevant mapping *AA*/*trípl* is chosen.

To solve this problem, after requiring an alignment using (2), we merge an irrelevant mapping (e.g., *AA*/*trípl*) and its neighbor mapping (e.g., *A*/*éi*) by employing the number of possible mappings before and after the mapping required by a mapping included in estimated alignments. In our many-to-many alignment, a mapping with multiple mappings in forward and backward contexts is more likely to be a correct mapping. On the other hand, a mapping other than those above is more likely to be an irrelevant mapping. By employing such heuristics, we merge a mapping that has only one context in a forward or backward direction and a mapping present in that direction to avoid an irrelevant mapping.

5. Experiments and results

We evaluate the conventional method (joint multigram approach) and our many-to-many alignment including the extensions described in Section 4, by automatic pronunciation annotation using Web text mining[4].

5.1. Experimental description

The methods compared are as follows.

1. Direct employment of aligned data without alignment[4] (*baseline*).
2. Joint multigram approach[6, 8] (*joint*).
3. Our many-to-many alignment with the city block distance[9] (*city*).
4. Our n-best Viterbi training in Section 4.1 added to *city* (*city+del*).
5. The mergence method of irrelevant mapping in Section 4.2 added to *city+del* (*city+del+merge*).

The experimental procedure is as follows.

1. In each compared method except *baseline*, train the parameter for alignment using the aligned data, and estimate an alignment over the aligned data.
2. Perform automatic pronunciation annotation using Web text mining[4] employing only mappings included in each estimated alignment. *Baseline* directly employs aligned data instead of these mappings.
3. Evaluate *Recall*, *Precision* and *F-value*.

Recall, *Precision* and *F-value* are shown as follows.

$$Recall = \frac{R}{C} \quad (7)$$

$$Precision = \frac{R}{N} \quad (8)$$

$$\textit{F-value} = \frac{R}{0.5(N+C)} \quad (9)$$

R is the number of keywords for which a correct pronunciation is accepted, C is the number of keywords for which a correct pronunciation is extracted from the Web, N is the number of keywords for which a pronunciation extracted from the Web is accepted. *Recall* represents the height of the generalization ability for OOV words. *Precision* represents the occurrences of an irrelevant mapping.

The experimental conditions are as follows. For maximum lengths of graphemes and phonemes in a single mapping, *baseline*, *city*, *city+del* and *city+del+merge*

Table 2: Experimental data

aligned data	Japanese Kanji dictionary (Wnn[1], Sanseido[2]), Naist Japanese dictionary[3], Eijiro[4]. Total amount of data is about 350 thousand words.
test data	Search queries that could not annotate a correct pronunciation by employing Kytea[5] in search queries, where there are kanji, hiragana, katakana, and alphabetical characters, obtained from Yahoo! search ranking[6], Google Trends[7], and Goo keyword ranking[8] excluding multiple words and URLs. Total amount of data is about two thousand querys.

have no limit, whereas *joint* has limits of *1-3*, *1-6*, *2-3*, *2-6*, *3-3*, and *3-6*, where *N-M* denotes the maximum lengths of *N* graphemes and *M* phonemes, respectively. The n-best Viterbi training in *city+del* and *city+del+merge* employs only 2-best alignments. The hypothesis that the deletion character continuously appears in the expansion of the pronunciation hypothesis is prohibited. The number of Web pages used in mining is 500 pages per keyword. The threshold of confidence, which means the similarity based on DP matching, for accepting a pronunciation extracted from Web is 1. Details of experimental data are shown in Table 2.

5.2. Experimental result

The experimental result is shown in Table 3. In the comparison of *Recall*, *city+del+merge* was found to outperform *joint 1-6*, which has the highest *Recall* in *joint*, by about 1.9 points. The difference between these *Recall* results is significant according to paired t-testing at a level of 0.05. In the comparison of *Precision*, *joint 3-6*, which has the highest *Precision* in *joint*, outperformed *city+del+merge* by about 0.6 points. However, the difference between those values of *Precision* is not significant according to paired t-testing at a level of 0.05. These results shows that our many-to-many alignment including extensions in Step 4 produces an alignment that has a high generalization ability for OOV words without degrading *Precision* compared with the joint multigram approach of the conventional method. Additionally, in terms of *F-value*, *city+del+merge* outperformed *joint 2-3*, which has the highest *F-value* in *joint*, by about 1.9 points.

[1] http://freewnn.sourceforge.jp/
[2] http://www.sanseido-publ.co.jp/
[3] http://sourceforge.jp/projects/naist-jdic/
[4] http://www.eijiro.jp
[5] http://searchranking.yahoo.co.jp/
[6] http://www.google.co.jp/m/trends
[7] http://ranking.goo.ne.jp/keyword/
[8] http://www.phontron.com/kytea/index-ja.html

Table 3: Experimental result

		Recall (%)	*Precision* (%)	*F-value* (%)
baseline		**76.19**	**95.87**	**84.91**
conventional methods	*joint 1-3*	87.95	93.27	90.54
	joint 1-6	**88.38**	92.81	90.54
	joint 2-3	86.46	95.81	**90.90**
	joint 2-6	85.46	95.54	90.22
	joint 3-3	85.10	95.83	90.15
	joint 3-6	83.96	**95.93**	89.55
proposed methods	*city*	89.24	93.43	91.29
	city+del	90.24	95.12	92.61
	city+del +merge	**90.31**	**95.33**	**92.75**

6. Conclusion

We proposed a many-to-many alignment including extensions, which are parameter estimation that considers the property of the deletion character by employing n-best Viterbi training, and the mergence method of an irrelevant mapping, and performed the evaluation experiment of our expanded many-to-many alignment by automatic pronunciation annotation using Web text mining. The experiment revealed that our expanded many-to-many alignment improves the generalization ability for OOV words compared with the joint multigram approach of the conventional method.

7. References

[1] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[2] S. Jiampojamarn and G. Kondrak, "Online discriminative training for grapheme-to-phoneme conversion," In *Proc. INTERSPEECH*, pp. 1303–1306, September 2009.

[3] S. Jiampojamarn, C. Cherry and G. Kondrak, "Integrating joint n-gram features into a discriminative training framework," *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pp. 697–700, June 2010.

[4] J. Miyake, S. Takeuchi, H. Kawanami, H. Saruwatari and K. Shikano, "Automatic reading annotation to Japanese trendy words based on parentheses expression," In *Proc. Oriental COCOSDA 2008*, November 2008.

[5] R. I. Damper, Y. Marchand, J. D. S. Marsters and A. I. Bazin, "Aligning text and phonemes for speech technology applications using an EM-like algorithm," In *Journal of Speech Technology*, Vol. 8, No. 2, pp. 147-160, June 2005.

[6] S. Deligne, F. Yvon and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," In *Proc. EUROSPEECH*, pp. 2243–2246, September 1995.

[7] S. Deligne and F. Bimbot, "Inference of variable-length acoustic units for continuous speech recognition," *Speech Communication*, vol. 23, no. 3, pp. 223–241, 1997.

[8] S. Jiampojamarn, G. Kondrak and T. Sherif, "Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion," In *Proc. NAACL HLT 2007*, pp. 372-379, April 2007.

[9] K. Kubo, H. Kawanami, H. Saruwatari and K. Shikano, "Unconstrained many-to-many alignment for automatic pronunciation annotation," In Proc. APSIPA 2011, October 2011.