Unconstrained Many-to-Many Alignment for Automatic Pronunciation Annotation

Keigo Kubo*, Hiromichi Kawanami*, Hiroshi Saruwatari* and Kiyohiro Shikano*
* Graduate School of Information Science, Nara Institute of Science and Technology, Japan
E-mail: {keigo-k, kawanami, sawatari, shikano}@is.naist.jp

***Abstract*—An alignment between graphemes and phonemes is vital data to annotate the pronunciation for out-of-vocabulary words. We desire an alignment to be (1) many-to-many and (2) fine-grained. A traditional one-to-one alignment model does not represent an intuitive mapping for logograms, such as Chinese characters, and has previously reported an inferior performance in phoneme prediction. A conventional many-to-many alignment model prefers a mapping consisting of longer substrings, which degrades the generalization ability of the prediction model, especially for out-of-vocabulary words. In order to obtain a highly generalized model, we introduce city block distance in the conventional many-to-many alignment, so that fine-grained mappings are inferred without constraining the maximum lengths of both graphemes and phonemes. Experimental results show that our extension improves the baseline grapheme-to-phoneme conversion on several language data sets.**

I. Introduction

Recent advances in speech recognition have made possible to attempt large-scale, open-domain, data-driven approaches. Out-of-vocabulary 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 out-of-vocabulary words. Out-of-vocabulary words can be collected easily from Web text resources, but their pronunciation remains unknown. An automatic pronunciation annotation is desired. Statistical approaches including a grapheme-to-phoneme (g2p) conversion[1-4] and knowledge-based approaches such as identifying a part of the Web text that describes word-pronunciation pairs[5] have been proposed.

An alignment between graphemes and phonemes is vital data for these pronunciation annotation methods. In this paper, we focus on the alignment methods, such as a one-to-one alignment[6] and a many-to-many alignment[7-9]. In [7,8], the many-to-many alignment is named joint multigrams. As it and the one proposed in [9] are essentially the same, both methods are treated as joint multigram approach in this paper. Ref.[9] explains the suitability of the joint multigram approach over a one-to-one alignment and shows better performances for this approach. However, the joint multigram approach generally prefers a mapping consisting of longer substrings, which degrades the generalization ability of the prediction model. To cope with this problem, we introduce city block distance, which is employed in Dynamic Time Warping, in the joint multigram approach, in such a way that the resulting mappings are pairs of substrings that are unconstrained in length, yet fine-grained to increase the generalization ability of the prediction model. Our extension has shown to be effective for a g2p conversion of out-of-vocabulary words.

The rest of this paper is organized as follows. In Section II, we explain our motivation for an unconstrained many-to-many alignment. In Section III, the EM algorithm derivations are formally described. We report experiments in Section IV and give a discussion in Section V. Finally, Section VI states our conclusion.

II. Motivation

First, we introduce the terms used in this paper, explain the joint multigram approach and the basic idea of our extension.

A. Preliminaries

Let d be a tuple of a word and its pronunciation, and D be a set of the d tuples. Let $\boldsymbol{U}_d$ be a set of alignment candidates of a grapheme sequence and a phoneme sequence, generated from the d tuple. We call a unit sequence $\boldsymbol{u}$ to be a g2p alignment of the tuple d in the set $\boldsymbol{U}_d$, and a unit u to be a mapping in the unit sequence $\boldsymbol{u}$. We denote ϵ to be a null character that represents a missing grapheme or a missing phoneme in the unit u. An example of a word-pronunciation pair $\langle able, \acute{e}ibl \rangle$ is shown below:

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

B. Joint multigram approach

The joint multigram approach proposed in [7] is:

$$
\begin{aligned}
\hat{\boldsymbol{u}} &= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d) \\
&= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(d|\boldsymbol{u}) P(\boldsymbol{u}) \qquad (1) \\
&= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}) \qquad (2) \\
&\simeq \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} P(u) \qquad (3)
\end{aligned}
$$

$P(d|\boldsymbol{u})$ is equal to 1 as d is uniquely decided for a given $\boldsymbol{u}$ (Eqn.(1)→Eqn.(2)). $P(\boldsymbol{u})$ is assumed to be the unigram probabilities of a unit u (Eqn.(2)→Eqn.(3)). The above indicates that we can estimate the best alignment $\hat{\boldsymbol{u}}$ between graphemes and phonemes using the Viterbi algorithm, if we appropriately obtain $P(u)$. The EM algorithm can be used to estimate $P(u)$ by maximizing $P(D) = \prod_{d \in D} P(d)$. The formal derivation of the EM algorithm will be described in Section 3.

C. Basic idea of our method

Unlike a traditional one-to-one alignment model, the joint multigram approach prefers longer units, i.e. mappings of a longer grapheme sequence and/or a longer phoneme sequence. This is because $P(\boldsymbol{u})$ is unfairly influenced by the number of units u in the unit sequence $\boldsymbol{u}$. $P(\boldsymbol{u})$ is calculated by multiplying unigram probabilities of units $P(u)(\leq 1)$ in the unit sequence $\boldsymbol{u}$ (see Eqn.(3)). Generally, the unit sequence $\boldsymbol{u}$ composed of many (and short) units u performs more multiplications than the $\boldsymbol{u}$ composed of a few (and long) units in $P(\boldsymbol{u})$. This implies that the $P(\boldsymbol{u})$ composed of many (and short) units u is lower than that composed of a few (and long) units u. In order for a phoneme prediction to be well-performed, we believe the prediction model must be highly generalized. In order to obtain a highly generalized model, a many-to-many alignment of graphemes and phonemes must also be fine-grained.

To meet the requirements above, [7-9] limits the maximum length of graphemes and the maximum length of phonemes in a single unit u. The parameters set in [9] were two. However, appropriate values for the parameters depend on languages. In case of Japanese out-of-vocabulary words including Kanji (Chinese characters), one grapheme could map to more than two phonemes. If we were to set the parameters more than two, the resulting joint multigram approach would no longer be fine-grained.

In contrast, we introduce city block distance in $P(\boldsymbol{u})$ as an exponential. Recall that $\boldsymbol{u}$ composed of many (and short) units u performs more multiplications than that composed of a few (and long) units, and hence, longer units are generally preferred over shorter units. We note that the sum of characters in a tuple d is uniform. The number of characters found in each unit sequence $\boldsymbol{u}$ of $\boldsymbol{U}_d$ is nearly the same, with the exception of null characters. By introducing city block distance in $P(\boldsymbol{u})$ as an exponential, the total number of multiplications performed in $P(\boldsymbol{u})$ is bounded by the number of characters in the unit sequence $\boldsymbol{u}$. In consequence, longer units are not advantageous over shorter units.

Let i_u be the number of characters in a grapheme sequence of a unit u, j_u be the number of characters in a phoneme sequence of a unit u. Then, our unconstrained many-to-many alignment is defined as:

$$\begin{aligned} \hat{\boldsymbol{u}} &= \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d) \\ &\simeq \arg\max_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} P(u)^{s_u} \end{aligned} \tag{4}$$

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

$$s_u = \begin{cases} i_u + C & \text{if } u \text{ is a missing phoneme} \\ j_u + C & \text{if } u \text{ is a missing grapheme} \\ i_u + j_u & \text{otherwise} \end{cases} \tag{5}$$

If there are several $\boldsymbol{u}$ that have the same value for $\prod_{u \in \boldsymbol{u}} P(u)^{s_u}$, the $\boldsymbol{u}$ composed of a fewer (and longer) units is chosen.

In [9], $P(\boldsymbol{u})$ incorporating a unit that was a missing phoneme or a missing grapheme were naturally decreased as the number of multiplications increased. However, such a mechanism does not work in our extension. In order to compensate this, we introduce a penalty C which takes care of the missing phoneme or the missing grapheme. However we also prohibit the missing grapheme as referenced in [9].

III. Formal EM derivation

Here we show the formal EM derivations for the method in [7] and our extended method.

A. Joint multigram approach

Let θ be the set of the current model parameters, and $\hat{\theta}$ be the set of the updated model parameters. The model parameters for $P(u)$ are denoted below as $p_u \equiv P(u|\theta)$. We treat d as an observed variable and u as a hidden variable. The Q function of EM algorithm for [9] can be defined as:

$$\begin{aligned} Q(\hat{\theta}|\theta) &= \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d, \theta) \log P(\boldsymbol{u}, d|\hat{\theta}) \\ &= \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d, \theta) \sum_{u \in \boldsymbol{u}} \log \hat{p}_u \end{aligned} \tag{6}$$

In order to estimate $\hat{p}_u$, the Lagrangian for $Q(\hat{\theta}|\theta)$ is:

$$L(\hat{\theta}, \lambda) = \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d, \theta) \sum_{u \in \boldsymbol{u}} \log \hat{p}_u + \lambda \left(\sum_{u \in U} \hat{p}_u - 1 \right) \tag{7}$$

where U is the set of all unit types. The maximum likelihood estimation of $\hat{p}_u$ is given by:

$$\hat{p}_u = \frac{\gamma_u}{\sum_{u \in U} \gamma_u} \tag{8}$$

where γ_u can be calculated by:

$$\begin{aligned} \gamma_u &= \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} P(\boldsymbol{u}|d, \theta) n_u(\boldsymbol{u}) \\ &= \sum_{d \in \boldsymbol{D}} \sum_{\boldsymbol{u} \in \boldsymbol{U}_d} \frac{\prod_{u \in \boldsymbol{u}} p_u}{\sum_{\boldsymbol{u} \in \boldsymbol{U}_d} \prod_{u \in \boldsymbol{u}} p_u} n_u(\boldsymbol{u}) \end{aligned} \tag{9}$$

where $n_u(\boldsymbol{u})$ is the number of occurrences of u in $\boldsymbol{u}$.

A pseudo code for the above EM algorithm is:

1) Set an initial value of p_u.
2) Calculate an expectation (E-step), shown in Eqn.(9).
3) Calculate the maximum likelihood (M-step), shown in Eqn.(8).

4) Substitute $\hat{p}_u$ for p_u.
5) Finish if converges, or return to Step 2.

The initial value of p_u is set uniform. The E-step is calculated by Forward-Backward algorithm.

B. Our proposed method

Similar to [7], the corresponding EM algorithm can be derived in our method. Equation (4) assumes that u occurs s_u times, where s_u is city block distance, which is unrelated to the occurrence of u. We introduce a penalty term for s_u to the Lagrangian for our proposed method:

$$\begin{aligned} L(\hat{\theta},\lambda) = & \sum_{d\in\boldsymbol{D}}\sum_{\boldsymbol{u}\in\boldsymbol{U}_d} P(\boldsymbol{u}|d,\theta)\sum_{u\in\boldsymbol{u}} s_u \log\hat{p}_u + \lambda(\sum_{u\in U}\hat{p}_u - 1) \\ & - \sum_{d\in\boldsymbol{D}}\sum_{\boldsymbol{u}\in\boldsymbol{U}_d} P(\boldsymbol{u}|d,\theta)\sum_{u\in\boldsymbol{u}}(s_u-1)\log\hat{p}_u \\ = & \sum_{d\in\boldsymbol{D}}\sum_{\boldsymbol{u}\in\boldsymbol{U}_d} P(\boldsymbol{u}|d,\theta)\sum_{u\in\boldsymbol{u}}\log\hat{p}_u + \lambda(\sum_{u\in U}\hat{p}_u - 1) \quad (10) \end{aligned}$$

Although the assumptions made in $P(\boldsymbol{u}|d,\theta)$ differ, our Lagrangian (Eqn.(10)) becomes the same form as the original Lagrangian (Eqn.(7)). Thus, the maximum likelihood is given by:

$$\hat{p}_u = \frac{\gamma_u}{\sum_{u\in U}\gamma_u}$$

where γ_u can be calculated by:

$$\gamma_u = \sum_{d\in\boldsymbol{D}}\sum_{\boldsymbol{u}\in\boldsymbol{U}_d} \frac{\prod_{u\in\boldsymbol{u}} p_u^{s_u}}{\sum_{\boldsymbol{u}\in\boldsymbol{U}_d}\prod_{u\in\boldsymbol{u}} p_u^{s_u}} n_u(\boldsymbol{u}) \quad (11)$$

The EM algorithm is similar to the method in [7], with a difference in the E-step given by Eqn.(11).

IV. Experiments and results

We evaluate the conventional method (joint multigram approach) and our proposed method (unconstrained many-to-many alignment) by g2p conversion on a *Conventional task* and a *Web task*. The *Conventional task* is a g2p task that has been employed in experiments on previous studies[1-4]. For the CMUDict, NETtalk, and Brulex data sets from the Pascal Letter-to-Phoneme Conversion Challenge[1], we attempt to faithfully follow the convention in terms of data exclusion and data size in [3].

The *Web task* is a g2p task that performs g2p conversion for new words in the Web that are not registered in a dictionary data source, as the setting will be more realistic for out-of-vocabulary words. We build two data sets: one for Japanese words with Kanji (Chinese characters) and the other for alphabetically spelled words. Their pronunciations are provided with Katakana (Japanese characters for syllables). As for the Kanji words, we use the *NAIST Japanese Dictionary (NAIST-jdic)*[2] and the *Hatena kanji*. *NAIST-jdic* is a dictionary that lists Japanese words and their pronunciations. *Hatena kanji* is a list of Kanji keywords registered at the website *Hatena Keyword*[3] in 2009 from which proper nouns in *NAIST-jdic* are excluded. *NAIST-jdic* is used in training, 30% of *Hatena kanji* is used in the development, and the remaining 70% is used in the test. As for the alphabetically spelled words, we use the *EIJIRO*[4] and the *Hatena alphabet*. *EIJIRO* is a dictionary that lists English words and their transliterated pronunciations. *Hatena alphabet* is a list of alphabetically spelled keywords at *Hatena Keyword* from which words in *EIJIRO* are excluded. *EIJIRO* is used in training, 30% of *Hatena alphabet* is used in the development, and the remaining 70% is used in the test. Table I summarizes the data sets; vocabulary sizes of training, development, and test.

TABLE I
VOCABULARY SIZES OF TRAINING SETS, DEVELOPMENT SETS AND TEST SETS IN EACH TASK.

task	data set	training	development	test
Conventional task	CMUDict(En)	100,944	5,883	12,000
	NETtalk(En)	17,751	1049	1000
	Brulex(Fr)	23,353	1,373	2,747
Web task	*NAIST-jdic*(Ja)	292,885	-	-
	Hatena kanji(Ja)	-	887	2,071
	EIJIRO(En)	63,632	-	-
	Hatena alphabet(En)	-	600	1,400

TABLE II
A SUMMARY OF PARAMETERS AND IMPLEMENTATIONS.

	conventional method	proposed method
Implementation of many-to-many alignment	m2m-aligner[5]	*own implementation*
Max size of grapheme	2(En,Fr) or 3 (Ja)	*unconstrained*
Max size of phoneme	2(En,Fr) or 3 (Ja)	*unconstrained*
Missing grapheme	prohibited	prohibited
Missing phoneme	allowed	allowed with $C = 1$
Implementation of g2p conversion	DirectTL+[6]	DirectTL+

Table II describes the parameters and tools used in the experiment. We implemented our method to save more memory because the number of units u becomes huge by not imposing a maximum length of graphemes and phonemes for each unit. Our implementation is efficient in memory usage (3GB in our implementation vs. 6GB in m2m-aligner for the case of *NAIST-jdic*) to allow unconstrained lengths of graphemes and phonemes. This indicates that our proposed method can be computationally realized. We set the max grapheme and max phoneme to 3 in the conventional method for *NAIST-jdic* and *Hatena kanji* to address Jukujikun, idiomatic pairings of multiple characters and a specific pronunciation.

The evaluation measure we use is word accuracy:

$$\text{word accuracy} = \frac{|R|}{|V|} \quad (12)$$

$|V|$ is the number of phoneme sequences in the test set. $|R|$ is the number of correct phoneme sequences estimated.

[1] http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/Datasets
[2] http://sourceforge.jp/projects/naist-jdic/
[3] http://d.hatena.ne.jp/keywordlist?s=furigana
[4] http://www.eijiro.jp
[5] http://code.google.com/p/m2m-aligner/
[6] http://code.google.com/p/directl-p/

TABLE III
RESULTS OF *Conventional task* FOR THE CONVENTIONAL METHOD AND OUR PROPOSED METHOD.

data set	conventional method	proposed method
CMUDict(En)	72.95%	**73.16%**
NETtalk(En)	71.10%	**73.70%**
Brulex(Fr)	94.90%	**95.09%**

TABLE IV
RESULTS OF *Web task* FOR THE CONVENTIONAL METHOD AND OUR PROPOSED METHOD.

data set	conventional method	proposed method
Hatena kanji	45.00%	**47.03%**
Hatena alphabet	29.50%	**32.00%**

Table III and Table IV show the results of word accuracy in *Conventional task* and *Web task*. Our proposed method outperforms the conventional method in *Conventional task* and *Web task*, indicating that our unconstrained many-to-many alignment model produces better training data for a phoneme prediction model.

V. DISCUSSION

The conventional method, in a replication study, could not attain the reported performance in [4] for CMUDict, Nettalk and Brulex as exactly the same model parameters, the employed features and data split could not be replicated. However, our proposed method gives better results in the phoneme prediction under the same experimental conditions.

We conjecture that the improvement could be attributed to the fine-grained alignments of our proposed method. Table V shows a distribution of unit types applied in the phoneme prediction of *Hatena kanji*. g is the length of a grapheme, and p is the length of a phoneme. From Table V, our proposed method performs fine-grained alignments. For example, in Table V, the number of unit types with 2-3 pairs and 3-2 pairs are reduced from 10450 to 44 and 361 to 0 respectively. The training data with fine-grained mappings lead to an improvement in the word coverage for out-of-vocabulary words. As our proposed method doesn't set the max sizes, appropriate alignments can be obtained. It yields correct phoneme prediction in some samples. For example, in $\langle MANSOUR, MAENSER \rangle$ included in test set of CMUDict, a g2p conversion with our proposed method correctly output $M/M\ \ A/AE\ \ N/N\ \ S/S\ \ OUR/ER$ and a g2p conversion with the conventional method generates wrong output $M/M\ \ A/AE\ \ N/N\ \ S/S\ \ OU/AW\ \ R/ER$. Here, the correct mapping OUR/ER is produced because the max sizes are not given beforehand. For these reasons, our proposed method provides better training data than the conventional method.

The word accuracy in *Hatena kanji* and *Hatena alphabet* is clearly inferior than the other test sets. A possible explanation for the poor performances is that Katakana pronunciations in *Hatena kanji* and *Hatena alphabet* are not standarized. Thus, homographs had to be included in the training of many-to-many alignment models so as to support any pronunciation rule. The word accuracy was lower because of the difficult test data.

TABLE V
DISTRIBUTION OF UNIT TYPES USED IN *Hatena kanji* PREDICTION TEST. g IS THE LENGTH OF A GRAPHEME. p IS THE LENGTH OF A PHONEME.

	conventional method			proposed method		
	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p>=3$
$g=1$	4,129	7,945	5,681	5,914	11,449	4,938
$g=2$	115	0	10,450	202	6	44
$g>=3$	20	361	0	29	0	0

To cope with homographs, we explore the possibility to enhance a phoneme prediction by an annotation method from the Web text resource [5]. Ref.[5] requires fine-grained mappings of a grapheme sequence and a phoneme sequence, which can be obtained from our proposed method. It is a research avenue to explore in future.

VI. CONCLUSION

We proposed an unconstrained many-to-many alignment that introduces city block distance in the joint multigram approach proposed in [7], so that mappings of fine-grained substrings are inferred without imposing maximum lengths of both graphemes and phonemes. Fine-grained mappings of graphemes and phonemes provide us with a highly generalized model for a g2p conversion. Experiments on many-to-many alignments show that our unconstrained many-to-many alignment improves the baseline g2p conversion in all test cases.

ACKNOWLEDGMENTS

This work was partially supported by CREST (Core Research for Evolutional Science and Technology), Japan Science and Technology Agency(JST).

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, C. Cherry and G. Kondrak, "Joint Processing and Discriminative Training for Letter-to-Phoneme Conversion," In *Proc. ACL-08:HLT*, pp.905–913, June, 2008.

[3] S. Jiampojamarn and G. Kondrak, "Online Discriminative Training for Grapheme-to-Phoneme Conversion," In *Proc. INTERSPEECH*, pp.1303–1306, September, 2009.

[4] 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.

[5] J. Miyake, S. Takeuchi, H. Kawanami, H. Saruwatari, K. Shikano, "Automatic Reading Annotation to Japanese Trendy Words based on Parentheses Expression," In *Proc. Oriental COCOSDA 2008*, November, 2008.

[6] R. I. Damper, Y. Marchand, J. DS. 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.

[7] S. Deligne, F. Yvon, F. Bimbot, "Variable-Length Sequence Matching for Phonetic Transcription Using Joint Multigrams," In *Proc. EUROSPEECH*, pp.2243–2246, September, 1995.

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

[9] 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.