interspeech 2008 incorporating SST 08

Extracting Word-Pronunciation Pairs from Comparable Set of Text and Speech

Tetsuro Sasada, Shinsuke Mori and Tatsuya Kawahara

Graduate School of Informatics, Kyoto University
Sakyo-ku, Kyoto, 606-8501, Japan
sasada@ar.media.kyoto-u.ac.jp

Abstract

One of the problems in text-to-speech (TTS) systems and speech-to-text (STT) systems is pronunciation estimation of unknown words. In this paper, we propose a method for extracting unknown words and their pronunciations from similar sets of Japanese text data and speech data. Out-of-vocabulary words are extracted from text with a stochastic model and pronunciations hypotheses are generated. These entries are verified by conducting automatic speech recognition on audio data. In this work, we use news articles and broadcast TV news covering similar topics. Most extracted pairs turned out to be correct according to a human judges. We also tested the TTS front-end enhanced with these entries on other web news articles, and observed an improvement in the pronunciation estimation accuracy of 9.2% (relative). The proposed method can be used to realize a spoken language processing system that acquires and updates its lexicon automatically.

1. Introduction

Recent advances in spoken language processing (SLP) techniques have given rise to a number of practical applications. One of these applications is text-to-speech (TTS), which converts written text into speech. One of the largest obstacles in a TTS system is the existence of unknown words. Usually TTS systems are equipped with a module which estimates a pronunciation of unknown words from their spelling. However, the accuracy of this module is not sufficiently high, especially in languages which use ideograms such as Japanese and Chinese. Unknown words or out-of-vocabulary words are also problematic in speech-to-text (STT) systems.

In this paper, we propose a method for extracting unknown words and their pronunciations automatically from comparable sets of text data and speech data. The main idea is to compare a collection of text data and a collection of speech data talking about the same topics. Our method is summarized as follows:

1. Extract unknown word candidates from the text data.
2. Enumerate possible pronunciations for each word candidate.
3. Search for pronunciations in the speech data.

The search is executed by using an automatic speech recognizer (ASR). Unless the searched pronunciation is very long, a possible pronunciations may be matched not only with correct words but also at incorrect positions in speech data. Thus, when we search for a possible pronunciation of an unknown word candidate, it is strongly required to check its context. This context can be calculated from sentences in text data.

In some languages such as Japanese, the target language of this research, words are not separated by a whitespace. Thus first of all, word boundaries must be identified by an automatic word segmenter. However, an automatic word segmenters tend to make errors at unknown words and output incorrect word boundaries. So we regard a text as a stochastically segmented corpus (SSC) [1] in which sentences are segmented into word sequences stochastically, not determinatively as in ordinary methods. The ASR system searches for all possible pronunciations of unknown word candidates in speech data, representing contexts with a word n-gram model estimated from an SSC.

In the experiment, we extract word-pronunciation pairs from broadcast TV news and web news articles in the same period. Evaluation is done using a different set of web news articles.

2. Language Model for TTS Front-end

The method we propose in this paper for extracting unknown words and their pronunciations uses an ASR coupled with a language model (LM) describing the contexts of the unknown word candidates. In this section, we explain a TTS front-end based on n-gram modeling.

2.1. Text-to-Speech Front-end

In the stochastic approach for pronunciation estimation [2], a sentence is regarded as a sequence of pairs u consisting of spelling of a word w and a phoneme sequence $\boldsymbol{y}$, that is $u = \langle w, \boldsymbol{y} \rangle$[1]. Using an n-gram model based on this unit, $M_{u,n}$, the probability of a unit sequence $\boldsymbol{u} = (u_1 u_2 \cdots u_h)$, is calculated as:

$$M_{u,n}(\boldsymbol{u}) = \prod_{i=1}^{h+1} P(u_i | \boldsymbol{u}_{i-n+1}^{i-1}),$$

where u_i $(i \leq 0)$ and u_{h+1} is a special symbol BT (boundary token).

Given a character sequence $\boldsymbol{x}$ as an input sentence, the front-end outputs $\hat{\boldsymbol{u}}$, a sequence of units with the highest probability, under the constraint that the concatenation of the spellings is equal to the input sentence:

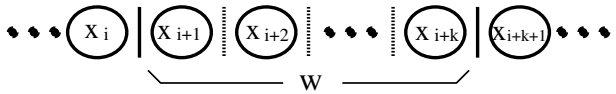
$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{x}=w_1 w_2 \cdots w_h}{\textbf{argmax}} \; M_{u,n}(u_1 u_2 \cdots u_h), \tag{1}$$

where w_i is the spelling of the pair u_i.

2.2. Pronunciation Estimation for Unknown Word

In order to handle unknown words, a special symbol UU is introduced to represent all units outside of vocabulary $\mathcal{U}$, a set of word-pronunciation pairs. When a UU is predicted by $M_{u,n}$, a

[1] In the original paper [2] the unit is a quadruplet of spelling of a word, its part-of-speech, its phoneme sequence, and its accent sequence.



$$f_e(w) = P_i \times (1-P_{i+1}) \times \cdots \times (1-P_{i+k-1}) \times P_{i+k}$$

Figure 1: The expected frequency in a stochastically segmented corpus (SSC).

pair of spelling $\boldsymbol{x}_1^{h'}$ and a phoneme sequence is predicted by the following n-gram model $M_{v,n}$, based on a pair of a character and a phoneme sequence $v = \langle x, \boldsymbol{y} \rangle$:

$$M_{v,n}(\boldsymbol{v}_1^{h'}) = \prod_{i=1}^{h'+1} P(v_i | \boldsymbol{v}_{i-n+1}^{i-1}), \quad (2)$$

where v_i $(i \leq 0)$ and $v_{h'+1}$ is the special symbol BT.

Since Equation (2) does not refer to the context of the unknown word, the phoneme sequence with the highest probability for a given spelling w is selected in the argmax operation in Equation (1).

The focus of this paper is to generate appropriate pairs $u = \langle \boldsymbol{x}_1^{h'}, \boldsymbol{y} \rangle$ $(u \notin \mathcal{U})$, especially to predict pronunciation $\boldsymbol{y}$ for detected unknown word $\boldsymbol{x}_1^{h'}$.

3. LM Estimation from Stochastically Segmented Corpus

To cope with segmentation errors made by an automatic word segmenter, the concept of stochastic segmentation is proposed [1]. In this section, first we briefly introduce stochastically segmented corpus (SSC), then we explain a method for simulating an SSC with an ordinary corpus in which a sentence is segmented into a word sequence.

3.1. Stochastically Segmented Corpus (SSC)

An SSC is defined as a combination of a raw corpus C_r (hereafter referred to as the character sequence $\boldsymbol{x}_1^{n_r}$) and word boundary probabilities (WBPs) P_i that a word boundary exists between two characters x_i and x_{i+1}. Since there are word boundaries before the first character and after the last character of the corpus, $P_0 = P_{n_r} = 1$. When we calculate WBPs, we use an ME-based WBP estimator, whose possible features are set to be all of the character n-grams $(n \leq 3)$, all character type n-grams $(n \leq 3)$ contained in x_{i-2}^{i+2}, and others.

Given an SSC, word n-gram frequency can be calculated as an expected frequency of word n-gram. Word 0-gram is defined as an expected number of words in the SSC: $f(\cdot) = 1 + \sum_{i=1}^{n_r-1} P_i$. Word 1-gram is defined as follows. Let us assume that a word $\boldsymbol{w}$ (of length k) matches at the (i+1)-th position of the SSC, that is $\boldsymbol{w} = x_{i+1}x_{i+2}\cdots x_{i+k}$, the expected frequency of this occurrence is calculated as follows (see Figure 1):

$$f_e(\boldsymbol{w}) = P_i \left[\prod_{j=1}^{k-1} (1 - P_{i+j}) \right] P_{i+k}.$$

Since the word $\boldsymbol{w}$ may occur at other positions too, expected word 1-gram frequency in the SSC $f_r(\boldsymbol{w})$ is calculated as the summation of $f_e(\boldsymbol{w})$ over all occurrences: $f_r(\boldsymbol{w}) = \sum_{i \in O_1} f_e(\boldsymbol{w})$, where O_1 is the set of all occurrences. Word n-gram $(n \geq 2)$ frequencies are also calculated in the same way.

Similar to the word n-gram probability estimation from a decisively segmented corpus, word n-gram probabilities in the SSC are estimated by the maximum likelihood estimation method as relative values of word n-gram frequencies:

$$P(w) = f_r(w)/f_r(\cdot) \quad (n = 1),$$
$$P(w_n | \boldsymbol{w}_1^{n-1}) = f_r(\boldsymbol{w}_1^n)/f_r(\boldsymbol{w}_1^{n-1}) \quad (n \geq 2).$$

3.2. Simulating SSC Using Segmented Corpus: Pseudo-SSC

Calculation of word n-gram frequencies on an SSC is computationally costly. An SSC contains much more entries of words and word fragments than a segmented corpus. In addition we have to search for all occurrences of a word n-gram and execute multiple floating-point calculations instead of a single increment at each occurrence [1] .

In order to address this complexity problem, we propose a method for simulating an SSC by using a segmented corpus. We execute the following processes on the SSC from the first character to the last $(1 \leq i \leq n_r)$.

1. Output the character x_i.
2. Generate a random value $r_i \in [0, 1)$.
3. Compare r_i with P_i. If $r_i < P_i$, then output a word boundary symbol (whitespace). Otherwise output nothing.

With this process, we obtain a segmented corpus, which we call a pseudo-SSC, whose statistical characteristics are similar to that of the SSC. The frequencies counted on a pseudo-SSC of word n-grams occurring infrequently in the SSC may differ from the frequencies counted on the SSC. In order to mitigate this influence, we perform the above process N-time to obtain an N times larger corpus than the original one. We call the number N the multiplier.

Word n-gram probabilities on a pseudo-SSC are estimated by the maximum likelihood estimation from the word n-gram frequencies counted on the pseudo-SSC.

4. Extracting Word-Pronunciation Pairs

In this section, we describe a novel method for extracting words and their pronunciations from text data and speech data in detail. Figure 2 shows an overview of our method.

4.1. Extracting Word Candidates from Text

The first step of our method is to extract word candidates (spellings) from the text data. This process is executed by the following steps:

1. Annotate the text data with WBPs to form an SSC.
2. Produce N pseudo-SSCs from the SSC.
3. Extract spellings outside of a known word list (vocabulary) whose frequency in the pseudo-SSCs is more than a threshold F_{th}.

Now we have a list of word candidates.

4.2. Enumerating Pronunciations of Word Candidate

The next step is to annotate each word candidate with possible pronunciations by referring to a dictionary string possible pronunciations for each character [2]. Below we explain this process with an example of a word candidate "守屋[3]."

[2]The pronunciations of almost all words in Japanese are compositional, that is, the pronunciation of a word is a concatenation of the pronunciations of characters forming it, but most of characters have multiple pronunciation entries.

[3]This is the name of an administrative vice-minister of Japan's Ministry of Defense.

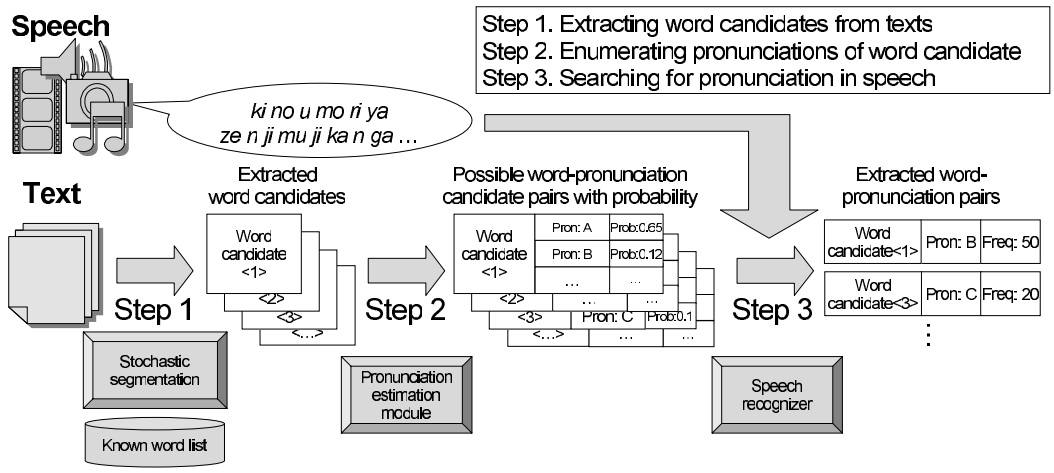


Figure 2: An overview of the proposed method

1. Decompose the spelling into a character sequence and generate all possible pronunciations for the characters from the dictionary
   ex.) 守 (*mo ri, ma mo, shu*), 屋 (*o ku, ya*)
2. List all pronunciations of the word candidate by taking one possible pronunciation for each character
   ex.) *mo ri o ku, mo ri ya, ma mo o ku, ma mo ya, shu o ku, shu ya*
3. For each possible pronunciation, calculate the joint probability in which the candidate word has the pronunciation using the n-gram model based on word-pronunciation pairs expressed by Equation (2).
   ex.) $P(mo\ ri\ o\ ku, 守屋) = 0.65$
   $P(mo\ ri\ ya, 守屋) = 0.12$
   ⋮

Note that in this example the correct pronunciation of the word "守屋" is "*mo ri ya*," the second probable one, thus the TTS front-end fails to produce a correct pronunciation of this word.

4.3. Searching for Pronunciation in Speech

The last step is to check if these hypothesized pronunciations for word candidates are observed in speech data. Since speech data have no clear word boundary information and contain pronunciation fluctuations and noises, a pronunciation may match at improper position as well. For example, let us assume that speech data contain the pronunciation of a word "memorial park" as follows:

$\cdots$ *me mo ri a ru pa a ku* $\cdots$.

A pronunciation "*mo ri ya*" for a word candidate "守屋" may matches by mistake at the position of "*mo ri a*" when the pronunciation of the word "memorial park" is fluctuated. Therefore it is important to check the contexts of word candidates when we search for pronunciations in speech data. So we propose to use an ASR system coupled with an LM estimated from our pseudo-SSC.

The following is the processes to count the frequencies of candidate pairs of word and pronunciation appearing at phonetically and linguistically proper positions in speech data.

1. Prepare an ASR system with a proper acoustic model for the speech data.
2. Add extracted word candidates to the vocabulary of the ASR system.
3. Re-estimate an LM of the ASR system from the pseudo-SSC used for word candidate extraction.
4. Execute speech recognition on the speech data talking about comparable topics to the text data.
5. Count the frequencies of word-pronunciation pairs in the ASR system results.

As a result of the above processes, we expect to obtain correct word-pronunciation pairs with their frequencies from text data and speech data.

5. Evaluation

As an evaluation of our method for extracting word-pronunciation pairs, we measured pronunciation estimation accuracies of a TTS front-end with and without extracted pairs.

5.1. Experiment Conditions

We prepared an annotated corpus composed of articles extracted from newspapers and example sentences in a dictionary of daily conversation. Each sentence in the corpus is segmented into words and each word is annotated with a phoneme sequence. Table 1 shows the corpus size. The ME-model for WBP estimation and a stochastic TTS front-end are built from this corpus.

Our method uses text data and speech data to extract word-pronunciation pairs. The text data we used are composed of two sources: one is newspapers, which is different from the corpus for building the ME-model, the other is web news articles crawled 4 times a day for 68 days (02/11/2007 - 08/01/2008). Table 2 shows the corpus size. We extracted word-pronunciation pairs from the text data. As for speech data we recorded 30 minute TV news for 34 days (05/12/2007 - 08/01/2008).

Then we tested the TTS front-end on the web news articles of 250 sentences on the day after the above period (09/01/2008).

5.2. Parameters and Other Features

We used the pseudo-SSCs derived from the text data for building an LM of the ASR, too. So we conducted preliminary experiments in which we calculated the perplexities of LMs built from N pseudo-SSCs by changing the multiplier N. The result showed that the LM built from 10 pseudo-SSCs had a similar perplexity to the LM built from the SSC. Thus we set N to 10.

Table 1: Annotated corpus.

#sentences	#words	#chars
52,955	1,254,867	1,844,106

Table 2: Text data.

	#sentences	#chars
newspaper	3,671,344	152,293,814
web news articles	33,336	2,550,120

The order of the n-gram models for pronunciation estimation is set to be 2, thus bi-gram models were used for pronunciation estimation for a sentence and an unknown word.

In order to reduce the computation cost of the ASR system we limited the number of possible pronunciations to 5, that is, at most 5-best pronunciations are annotated to each word candidate. Table 3 shows the size of the default vocabulary of the ASR system and the number of the added word candidates and word-pronunciation pairs.

The ASR system we used is Julius 3.5.3 [3] and its acoustic model is built from reading speech data of newspaper articles.

5.3. Extracted Word-Pronunciation Pairs

With the proposed method, we can obtain word-pronunciation pairs outside of the default vocabulary. We gathered those which appeared more than once in the ASR outputs and obtained 281 pairs. They contained words with correct pronunciations which the TTS front-end failed to estimate, such as 守屋 (*mo ri ya*), 武昌 (*ta ke ma sa*), ガス田 (*ga su de n*)[4].

We investigated extracted pairs and manually checked whether each spelling is a correct word or not and its pronunciation is correct or not. As a result, we found 95 word-pronunciation pairs of the top 100 pairs were correct.

5.4. Pronunciation Estimation Accuracy

We measured pronunciation estimation accuracy of the TTS front-end with and without extracted pairs. The evaluation criterion is accuracy, the percentage of correct phonemes in the pronunciation estimated by the system against those in the pronunciation annotated by a human.

Table 4 shows the accuracy of the baseline pronunciation estimation system and that enhanced with the word-pronunciation pairs extracted by our method. It is observed that there were 207 erroneous phonemes contained in the baseline system output and 19 (9.2%) of them are corrected by adding the word-pronunciation pairs extracted from text and speech data by our method.

Thus, the proposed method is capable of extracting proper word-pronunciation pairs from comparable text and speech data, and they are useful for a TTS front-end.

6. Related Work

This paper describes an attempt at extracting words and their pronunciations simultaneously from text and speech data in a language in which words are not separated by a whitespace. Kurata et al.[4] proposed a lexicon acquisition method aiming at reducing ASR errors based on a rule-based word candidate extraction [6]. Our extraction method is based on statistical criterion.

There are many attempts at extracting words or phrases only from texts [5, 6]. Compared to these methods, we extract word candidates just by counting the frequency of a spelling in N pseudo-SSCs produced from an SSC [1]. Though our process is simple, this refers to its "accessor variety" [6] and frequency in text indirectly. A spelling appearing in various contexts (high accessor variety) tends to be a "word" in the pseudo-SSCs and has more chances to be extracted. Moreover, we do not need a high precision of word extraction, but a high recall because word candidates are filtered by an ASR system referring to speech data.

[4]"武昌" is the first name of the vice-minister and "ガス田" means a gas field.

Table 3: Numbers of words and word-pronunciation pairs in the vocabulary of the ASR system.

	#words	#word-pron. pairs
default voc.	32,114	34,338
candidates	2,999	7,721

Table 4: Accuracies of pronunciation estimation results.

	accuracy = correct/annotated
baseline	99.29(%) = 29,132/29,339
+ extracted pairs	99.36(%) = 29,151/29,339

7. Conclusion

In this paper we have proposed a method for extracting unknown words and their pronunciations automatically from a comparable set of text data and speech data. The main idea is to compare a collection of text data composed of newspaper articles and a collection of speech data composed of broadcast news on similar topics. We observed an improvement in the pronunciation estimation accuracy of 9.2% (relative) when the system refers to the set of extracted pairs.

The proposed method will realize an autonomous SLP system which acquires and updates its lexicon by collecting relevant text and speech data.

8. References

[1] Shinsuke Mori and Daisuke Takuma. 2004. Word n-gram probability estimation from a Japanese raw corpus. In *Proc. of the ICSLP2004*.

[2] Tohru Nagano, Shinsuke Mori, and Masafumi Nishimura. 2005. A stochastic approach to phoneme and accent estimation. In *Proc. of the InterSpeech2005*.

[3] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. 2001. Julius – an open source real-time large vocabulary recognition engine. In *Proc. of the EuroSpeech2001*, pages 1691–1694.

[4] Gakuto Kurata, Shinsuke Mori, Nobuyasu Itoh, and Masafumi Nishimura. 2007. Unsupervised lexicon acquisition from speech and text. In *Proc. of the ICASSP2007*, pages 421–424.

[5] Shinsuke Mori and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proc. of the COLING96*.

[6] Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93.