Synchronous Morphological Analysis of Grapheme and Phoneme for Japanese OCR

Masaaki Nagata
NTT Cyber Space Laboratories
1-1 Hikarinooka, Yokosuka-shi,
Kanagawa 239-0847, Japan
nagata@nttnly.isl.ntt.co.jp

Abstract

We developed a novel language model for Japanese based on grapheme-phoneme tuples, which is one order of magnitude smaller than word-based models. We also developed an alignment algorithm of graphemes and phonemes for both ordinary text and OCR output. We show, by experiment, that the combination of the grapheme-phoneme tuple ngram model and the grapheme-phoneme alignment algorithm significantly improve character recognition accuracy if both grapheme and phoneme representations are given.

1 Introduction

In this paper, we present an alignment algorithm of *kanji* (Chinese character, grapheme) and *kana* (syllabary, phoneme) representations of the same content, and its application for recognizing handwritten characters of personal names.

Even for native Japanese, sometimes it is very difficult to read Japanese personal names, because there are about 7,000 Chinese characters, and each character has several different readings.

Therefore, it is common practice to write a person's name in both *kanji* and *kana* when submitting formal documents, such as application forms and questionnaires, as illustrated in Figure 1. This use of a certain amount of redundancy helps an operator avoid mistakes in the data entry process. Therefore, it is very likely that it could also be used to help computers reduce the number of character recognition errors they make.

フリガナ reading	フ ク ザ ワ last name in kana 'Fukuzawa'	ユ キ チ first name in kana 'Yukichi'
名 前 name	福 沢 last name in kanji 'Fukuzawa'	諭 吉 first name in kanji 'Yukichi'

Figure 1: An (artificial) example of how a Japanese person's name is written in both kanji (Chinese character, grapheme) and kana (syllabary, phoneme).

There is an enormous need for making the personal name (and address) entry process automatic, especially in government, banks, credit card companies, market research companies, etc. However, current Japanese handwriting character recognition technology is not reliable enough for this task. Character recognition accuracy is now around 90% for good quality documents, and around 70% for noisy documents such as FAX output.

Most of the recent research on the application of statistical language models to character recognition in Japanese uses either character ngram models or word ngram models (Konno and Hongo, 1993; Araki et al., 1994; Mori et al., 1996; Nagata, 1996; Nagata, 1998). These techniques require, at least, a context of a couple of characters to judge whether a character candidate is good. Therefore, they cannot be applied to the name recognition task because Japanese first and last names are usually only from one to three characters long (typically two characters).

In this paper, we present a novel language

model that is based on grapheme-phoneme tuples (a pair of *kanji* and *kana* representation). We also present an aligning algorithm of graphemes and phonemes both for ordinary text and OCR output. By experiment, we show that the language model and the alignment algorithm can significantly improve the overall recognition accuracy.

2 Grapheme-Phoneme Alignment of Japanese

We define grapheme-phoneme alignment of Japanese as the segmenting of a grapheme sequence (*kanji* representation) into minimum uncompositional units, each having a corresponding subsequence in the phoneme sequence (*kana* representation), and the aligning of each unit to the corresponding subsequence.

For example, let graphemes and phonemes of a family name "Fukuzawa" be 福沢 and フクザワ. The output of grapheme-phoneme alignment is two grapheme-phoneme tuples, 福/フク and 沢/ザワ, where the left and right side of '/' indicate graphemes and phonemes, respectively.

Most grapheme-phoneme correspondence in Japanese is one-to-many like the above example. By one-to-many, we mean that one grapheme corresponds to more than zero phonemes. However, one-to-zero, zero-to-one, many-to-many, and crossover correspondences are possible, as illustrated below.

- one-to-zero:
  五右衛門/ゴエモン → 五/ゴ 右/ϕ 衛/エ 門/モン
- zero-to-one:
  井上/イノウエ → 井/イ ϕ/ノ 上/ウエ
- many-to-many:
  紅葉/モミジ → 紅葉/モミジ
- crossover:
  不入斗/イリヤマズ → 不/ズ 入/イリ 斗/ヤマ

Many-to-many correspondence results from semantic translation of the Chinese word to Japanese. This semantic translation could result in a crossover correspondence because the word order of Chinese and Japanese is different, as in the last example above. But for simplicity (and since it is very rare), we will treat such a case as a many-to-many correspondence.

The advantage of using grapheme-phoneme tuples as basic units for the language model is their compactness, which makes the model one order of magnitude smaller than word-based models.

Table 1 shows the number of word tokens, word types, grapheme-phoneme tokens, and grapheme-phoneme types in a Japanese telephone directory of about 45,000,000 residential subscribers. This data was originally made for an automatic telephone directory assistance system (Higashida, 1994). For directory assistance use, grapheme (*kanji*) and phoneme (*kana*) representations of names were manually aligned. Considering the fact that Japan has a total population of 120,000,000 people, this is a fairly large and extensive sample of Japanese personal names.

Selecting names (including both first names and last names) that appeared at least 5 times results in a name list of 301K words, which covers more than 98% of the entire subscribers. But about the same coverage can be obtained by only 21K grapheme-phoneme tuples.

The problem with the language model based on grapheme-phoneme tuples lies in its ambiguity. In Japanese, each Chinese character usually has two different readings: one comes from its Chinese pronunciation (*on-yomi*), and the other comes from its semantic translation to Japanese (*kun-yomi*). However, it is common for one Chinese character to have several different Chinese-origin readings because of (a) pronunciation differences that developed with the passage of time, and (b) regional pronunciation differences in China. It is also common for one Chinese character to have several different Japanese-origin readings because of its semantic ambiguity.

As a result, both grapheme-to-phoneme and phoneme-to-grapheme conversions are very ambiguous. Moreover, in general, character readings for personal names are more ambiguous than those for ordinary text because there are a lot of readings that are used exclusively for personal names. Table 2

Table 1: Distribution of words and grapheme-phoneme tuples in a Japanese telephone directory

	word tokens		word types		g-p tokens		g-p types	
>=10	88.7M	97.6%	196K	14.8%	174M	97.0%	15K	16.7%
>=5	89.3M	98.4%	301K	22.6%	176M	97.8%	21K	23.4%
>=2	90.1M	99.1%	594K	44.8%	177M	98.8%	40K	44.4%
all	90.8M		1,327K		179M		90K	

Table 2: Comparison of Grapheme-to-Phoneme and Phoneme-to-Grapheme Ambiguity in personal names (telephone directory) and ordinary text (free kanji dictionary)

	directory		dictionary	
	max	ave	max	ave
G-to-P	258	10.9	36	3.2
P-to-G	1110	12.1	306	6.2

shows the maximum and average ambiguity of grapheme-to-phoneme and phoneme-to-grapheme correspondences in the telephone directory. For comparison, the same numbers in a public domain Japanese kanji dictionary (KANJIDIC)[1] are also shown to give a rough estimate of ambiguity in ordinary text. Table 2 shows that personal name readings are significantly more ambiguous than ordinary text readings, and that phoneme-to-grapheme mapping is more ambiguous than grapheme-to-phoneme mapping.

3 The Language Model and the OCR Model

3.1 Language Model

We formulate the alignment of graphemes and phonemes for OCR output in the noisy channel paradigm. Let input graphemes and phonemes be G and P, OCR output be G' and P'. The task is finding the most probable graphemes $\hat{G}$ and phonemes $\hat{P}$ that maximize $P(G,P|G',P')$. By using Bayes' rule, we obtain:

$$\begin{aligned}(\hat{G},\hat{P}) &= \arg\max_{G,P} P(G,P|G',P') \\ &= \arg\max_{G,P} P(G',P'|G,P)P(G,P) \quad (1)\end{aligned}$$

[1] ftp://ftp.cc.monash.edu.au/pub/nihongo/

We call $P(G,P)$ the language model, and $P(G',P'|G,P)$ the OCR model. We consider a language model based on smallest grapheme-phoneme tuples. $P(G,P)$ is approximated by the bigram model of a tuple of a grapheme p_i and a phoneme q_i as follows, where <bos> and <eos> represent the beginning and end of the sequence.

$$\begin{aligned}P(G,P) \approx \quad & P(g_1,p_1|<\text{bos}>)\prod_{i=2}^{i=n} P(g_i,p_i|g_{i-1},p_{i-1}) \\ & P(<\text{eos}>|g_n,p_n) \quad (2)\end{aligned}$$

The bigram probabilities $P(g_i,p_i|g_{i-1},p_{i-1})$ are estimated from the counts in the corresponding events in a corpus that is either manually or automatically aligned. The bigram probability of unknown tuples (not found in the dictionary) is estimated from their unigram probability by linear interpolation. The unigram probability of unknown tuples is estimated as the product of length probability $P(l_g,l_p)$, grapheme spelling probability $P(g)$, and grapheme phoneme probability $P(p)$, where l_g and l_p are the length of a grapheme g and a phoneme p.

$$P(g,p) \approx P(l_g,l_p)P(g)P(p) \quad (3)$$

We use empirical distribution learned from training data for length probability $P(l_g,l_p)$. We approximate grapheme spelling probability $P(g)$ by zerogram model (uniform distribution) because virtually any combination of characters could be a legitimate Japanese name:

$$P(g) \approx \prod_{j=1}^{l_g} P(cg_j) = 1/|C_g|^{l_g} \quad (4)$$

where cg_j is the individual character in grapheme sequence, and $|C_g|$ is the character set size of graphemes.

We approximate phoneme spelling probability $P(p)$ by bigram model because there are certain phonetic constraints in phoneme sequences.

$$P(p) \approx P(cp_1|<\text{bos}>) \prod_{i=2}^{i=n} P(cp_i|cp_{i-1}) P(<\text{eos}>|cp_n) \quad (5)$$

where cp_i is the individual character in a phoneme sequence.

3.2 OCR Model

For the OCR model, we assume that graphemes and phonemes are independently recognized, and that each character is also independently recognized within the graphemes and phonemes.

$$P(G', P'|G, P) = P(G'|G)P(P'|P) = \prod_{i=1}^{i=l_g} P(cg'_i|cg_i) \prod_{j=1}^{j=l_p} P(cp'_j|cp_j) \quad (6)$$

Ideally, the probability that an input character c_i will be recognized as an output character c_j should be estimated empirically. However, since there are 6879 graphemes (*kanji*) and 87 phonemes (*kana*) in the Japanese character set, JIS X 0208, it is impossible to estimate the probability empirically due to data sparseness. Therefore, we approximate it based on two parameters: the accuracy of the first candidate p_1 and the cumulative accuracy of all candidates p_n.

$$P(c_j|c_i) \approx \begin{cases} p_1 & \text{if } c_j \text{ is the first candidate} \\ \frac{p_n - p_1}{n-1} & \text{else if } c_j \text{ is among the candidates} \\ \frac{1-p_n}{|C|-n} & \text{otherwise} \end{cases} \quad (7)$$

where n is the number of candidates for the character, and $|C|$ is the character set size.

In this OCR model, regardless of the input and output character pairs, the first candidate is always assigned the probability p_1. For candidates other than the first candidate, the remaining cumulative accuracy $p_n - p_1$ is distributed uniformly. For characters not among the candidates, the remaining probability mass $1 - p_n$ is distributed uniformly.

```
1   T_{0,0} ← {<bos>}
2   φ_{0,0}(<bos>) ← 1
3   for s_x = 0 to l_g do
3     for s_y = 0 to l_p do
4       foreach (g_{i-1}, p_{i-1}) ∈ T_{s_x,s_y} do
5         for t_x = s_x + 1 to l_g do
6           for t_y = s_y + 1 to l_p do
7             (g_i, p_i) = (cg_{s_x}^{t_x}, cp_{s_y}^{t_y})
8             if (g_i, p_i) ∉ T_{t_x,t_y} then
9               T_{t_x,t_y} ← T_{t_x,t_y} ∪ {(g_i, p_i)}
10              φ_{t_x,t_y}(g_i, p_i) ← 0
11            endif
12            if (φ_{s_x,s_y}(g_{i-1}, p_{i-1}) P(g_i, p_i|g_{i-1}, p_{i-1})
                  > φ_{t_x,t_y}(g_i, p_i)) then
13              φ_{t_x,t_y}(g_i, p_i) ←
                  φ_{s_x,s_y}(g_{i-1}, p_{i-1}) P(g_i, p_i|g_{i-1}, p_{i-1})
14            endif
12          end
12        end
13      end
14    end
15  end
```

Figure 2: Grapheme-phoneme alignment algorithm (two-dimensional morphological analysis algorithm)

4 Grapheme-Phoneme Alignment Algorithm

4.1 Ordinary Text (a Pair of Strings)

First, we describe a Japanese grapheme-phoneme alignment algorithm for ordinary text, where its input is a pair of graphemes and phonemes. Although the algorithm does not identify word boundaries or parts of speech, we call this alignment task "synchronous morphological analysis" because grapheme-phoneme tuples in Japanese personal names can be thought of as a minimal compositional unit that has a certain meaning, which is the technical definition of morphemes. Moreover, the algorithm is a two-dimensional extension of a Japanese morphological analysis algorithm (Nagata, 1994).

Let input graphemes and phonemes be $G = cg_1 \ldots cg_{l_g}$ and $P = cp_1 \ldots cp_{l_p}$, where cg and cp are individual graphemes and phonemes. In order to find a sequence of grapheme-phoneme tuples $g_1, p_1, \ldots, g_n, p_n$ that maximizes $P(G, P)$ described in Equation (2), we use two-dimensional dynamic programming, as shown in Figure 2.

In Figure 2, $T_{x,y}$ is a table that holds grapheme-phoneme tuples ending at position

(x, y). $\phi_{x,y}(g_i, p_i)$ holds the maximum probability of grapheme-phoneme tuple sequences starting from $(0, 0)$ and ending at (x, y) whose final tuple is (g_i, p_i).

The algorithm starts from $(0, 0)$ which corresponds to the beginning of graphemes and phonemes, and proceeds toward the end of graphemes and phonemes (l_g, l_p), character by character, for both graphemes and phonemes. At every point (x, y) in the region $0 < x \leq l_g, 0 < y \leq l_p$, this algorithm updates the maximum probability for the subsequence of grapheme-phoneme tuples $\phi_{x,y}(g_i, p_i)$ (line 12 and 13 in Figure 2). Thus, at (l_g, l_p), we can obtain a sequence of tuples that maximizes $P(G, P)$.

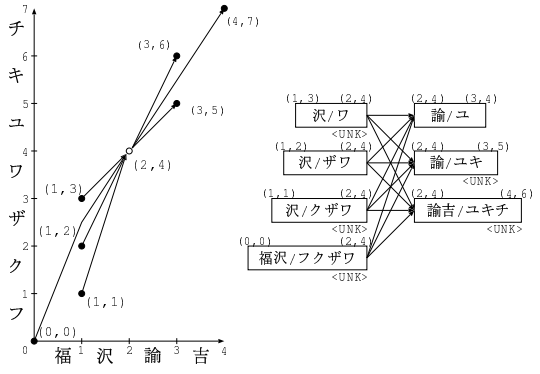

Figure 3: A snapshot of the grapheme-phoneme alignment for ordinary text

Figure 3 is a snapshot of the grapheme-phoneme alignment, where the input graphemes and phonemes are 福沢諭吉 and フクザワユキチ, and the current point is $(2, 4)$. There are four grapheme-phoneme tuples ending here, and three tuples starting here. All combinations of these tuples are searched, and the maximum probabilities up to the ending point of each tuples are updated.

4.2 OCR output (a Pair of Character Matrices)

Next, we describe a grapheme-phoneme alignment algorithm for OCR output. We assume there are no segmentation errors in the OCR output, which in practical terms means that the form has a grid for each character. In this case, we call the OCR output character matrix, in which each character has a list of several candidates ordered by their certainties. In fact, it is not difficult to extend the alignment algorithm to handle a character lattice, which is a data structure that considers the possibility of segmentation errors. However, we limited the input to a character matrix because we don't know how to make an OCR model that takes segmentation errors into account.

The alignment algorithm for OCR output is basically the same as shown in Figure 2. However, since there are sometimes no correct characters among the candidates, we introduce an approximate match between the grapheme-phoneme tuples in the dictionary and those in the character matrix (Here, we define a substring of a character matrix as a substring that is formed by selecting one character from each candidate list).

At each point (x, y), first, we retrieve grapheme-phoneme tuples using graphemes as keys:

1. List all tuples in the dictionary whose graphemes are a substring in the character matrix of graphemes starting from x.
2. Compute the minimum edit distance of their phonemes and substrings in the character matrix of phonemes starting from y.
3. Filter those tuples by edit distance and frequency.

As a threshold of edit distance, we filtered out tuples whose edit distance of phonemes is more than or equal to $l_p/2$, except when $l_p = 1$. Note if $l_p = 1$, edit distance cannot be used for filtering because it is either 0 or 1. Thus, we sorted the tuples by frequencies and selected the top 5 tuples. These thresholds were determined through experiments.

We then retrieve grapheme-phoneme tuples using phonemes as keys:

1. List all tuples in the dictionary whose phonemes are a substring in the character matrix of graphemes starting from y.

2. Compute the minimum edit distance of their graphemes and substrings in the character matrix of graphemes starting from x.

3. Filter those tuples by edit distance and frequency.

We also set the thresholds of edit distance and frequency of graphemes as $l_g/2$ and 5.

Finally, for unknown tuples, we list all combinations of the prefixes of the first candidates of graphemes starting from x and those of phonemes starting from y, if they are not already listed by the above approximate match.

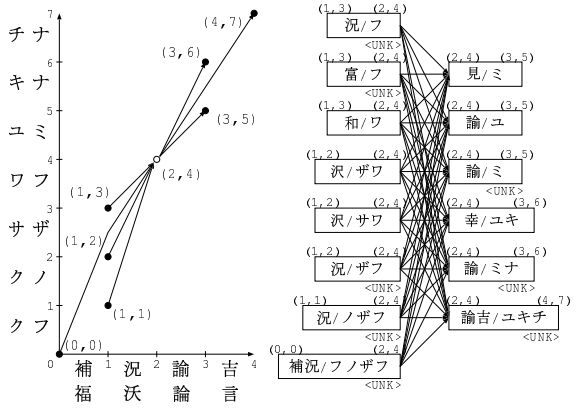

Figure 4: A snapshot of grapheme-phoneme alignment for OCR output

Figure 4 is a snapshot of grapheme-phoneme alignment for OCR output. For each character position of the input graphemes 福沢諭吉 and input phonemes フクザワユキチ, two recognition candidates are presented. Note there are three types of grapheme-phoneme tuples: exactly matched, approximately matched, and unknown. For example, from $(2,4)$ to $(3,5)$, the tuple 諭/ユ is generated because both grapheme 諭 and phoneme ユ are in the matrix. From $(2,4)$ to $(3,6)$, the tuple 幸/ユキ is generated because phoneme ユキ is in the matrix, and the tuple is highly frequent. Also from $(2,4)$ to $(3,6)$, an unknown tuple 諭/ミナ is generated because 諭 and ミナ are the prefixes of the first candidates of graphemes and phonemes.

5 Experiment

5.1 Training and Test Data

We used a Japanese name list of 1.3M words, which was originally made for an automatic telephone directory of about 45,000,000 residential subscribers (Higashida, 1994). Although the grapheme-phoneme alignment of the name list was manually done, because of the enormous amount of data in the telephone directory, grapheme-phoneme alignment of lower frequency names is slightly noisy. Therefore, we filtered out names which appeared no more than five times in the Japanese telephone directory.

This resulted in a name list of 301K words, which covers more than 98% of the entire subscribers. As shown in Table 1, there are 176M grapheme-phoneme tuple tokens in the name list, and there are 21K different grapheme-phoneme tuple types. We used 90% of the name list of 301K words for training, and non-overlapping 1000 names (words) for testing.

5.2 Grapheme-Phoneme Alignment Accuracy for Ordinary Text

Other than the grapheme-phoneme alignment model trained from manually aligned data, we made an alignment model which is bootstrapped from a public domain Japanese grapheme-to-phoneme dictionary (KANJIDIC). We call the former a supervised model, and the latter an unsupervised model.

As shown in Table 2, KANJIDIC has 3.2 readings for each Chinese character on the average. To make an alignment model, we consider the dictionary itself as a corpus, that is, we assign a uniform probability to all possible grapheme-phoneme tuples. The sum of the probabilities of unknown tuples is estimated by the Witten-Bell method (Witten and Bell, 1991), and redistributed based on the unknown tuple model, Equation (3).

Table 3 shows the grapheme-phoneme alignment accuracies of the supervised and unsupervised model. It was expected that the supervised model would achieve a very high grapheme-phoneme alignment accuracy. Surprisingly, however, the unsupervised model

Table 3: Grapheme-Phoneme Alignment Accuracy

	recall	precision	f
Supervised	99.6%	99.5%	99.5
Unsupervised	98.6%	98.7%	98.7

also achieves a very high accuracy, although it is not as good as that of the supervised model. This suggests the possibility that, for OCR purposes at least, no manually aligned data are in fact necessary.

5.3 Grapheme-Phoneme Alignment Accuracy for OCR output

In order to test the grapheme-phoneme alignment algorithm for OCR output, we used an OCR simulator that generates a character matrix from an input string, whose parameters are the first candidate accuracy and the cumulative accuracy of all candidates. We made four test sets whose first candidate accuracy and cumulative accuracy were (60%,90%), (70%,92.5%), (80%,95%), and (90%,97.5%), respectively. These parameters were selected based on the typical performance of Japanese handwriting OCR.

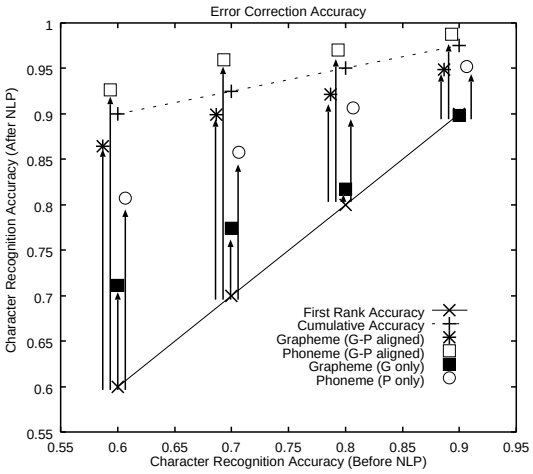

Figure 5: Character recognition accuracy before NLP and after NLP

Figure 5 shows the character recognition accuracy of the baseline OCR (before NLP) and that of synchronous analysis of graphemes and phonemes (after NLP). In Figure 5, grapheme (*kanji*) and phoneme (*kana*) accuracies are presented separately. For comparison, the accuracies obtained by using simple character bigram model are also presented. For example, if the baseline OCR recognition accuracy is 70%, by aligning graphemes and phonemes, the accuracies of graphemes and phonemes are improved to 89.9% and 96.6%, respectively. If we do not align them and apply language models independently, they are as low as 77.4% and 85.8%. It is obvious that the alignment algorithm successfully takes advantage of the redundancy to improve the overall recognition accuracy.

Table 4: Difference of the recognition accuracy between supervised and unsupervised models

	supervised		unsupervised	
	graph.	phon.	graph.	phon.
60% (90.0%)	86.4%	92.6%	86.3%	92.6%
70% (92.5%)	89.9%	96.0%	90.1%	96.1%
80% (95.0%)	92.1%	97.0%	92.0%	97.1%
90% (97.5%)	94.8%	98.8%	94.8%	98.8%

Table 4 shows the difference in character recognition accuracy between the supervised and unsupervised alignment models. Here, the unsupervised alignment model is made from the training data aligned by using the initial estimate of the unsupervised alignment model described in the previous section, i.e. the model is made by one reestimation.

There are virtually no differences in accuracy between the supervised and unsupervised models. This means that, if we have an initial grapheme-to-phoneme dictionary and a large amount of unaligned grapheme and phoneme representation of the same contents, we can automatically align them and use them as a language model for OCR, which significantly improves the overall recognition accuracy.

6 Discussion and Related Works

Grapheme-phoneme alignment is usually discussed in the context of text-to-speech synthesis applications. In recent years, a large number of works have been published on

grapheme to phoneme conversion, in particular, using finite state techniques. However, they dealt with either grapheme-to-phoneme conversion or phoneme-to-grapheme conversion (one is input and the other is output), while we are working on synchronous analysis of graphemes and phonemes (both grarphemes and phonemes are inputs and their alignments are output). Thus, there is little relevance between these.

As far as the authors know, the only paper that addresses the issue of grapheme-phoneme alignment accuracy in Japanese is one by Baldwin and Tanaka (1999). They reported 98.29% accuracy for general vocabulary words taken from a Japanese dictionary, by using an alignment model based on a score similar to TF-IDF, and an incremental unsupervised learning algorithm. It is very difficult to compare their results with ours because of the differences in the training and test data used. However, since we assume, in general, the name task is significantly more difficult than the general vocabulary task, we consider our result of 99.6% recall by supervised model and 98.6% recall by unsupervised model to have greater significance than their results.

Nagata (1998) proposed a Japanese OCR error correction method using word-based language model and character shape similarity. Compared with our simple OCR model Equation (6), their model can sort correction candidates with the same edit distance based on character shape similarity. This would be a very effective way to filter out grapheme-phoneme tuples retrieved from phonemes as keys in approximate matching, since phoneme-to-grapheme conversion is more ambiguous. Thus, we are considering implementing their OCR model as a subject for future work.

7 Conclusion

We developed a novel language model based on grapheme-phoneme tuples, which is one order of magnitude smaller than word-based models. We also developed an alignment algorithm of graphemes and phonemes for both ordinary text and OCR output. By using the language model and the alignment algorithm, we were able to significantly improve character recognition accuracy if both grapheme and phoneme representations of the input are given at the same time.

Acknowledgment

This research was done while the author was visiting at AT&T Labs. I wish to thank Ken Church and other members at AT&T Labs for their helpful comments and discussions.

References

Tetsuo Araki, Satoru Ikehara, Nobuyuki Tsukahara, and Yasunori Komatsu. 1994. An evaluation to detect and correct erroneous characters wrongly substituted, deleted and inserted in Japanese and English sentences using Markov models. In *COLING-94*, pages 187–193.

Timothy Baldwin and Hozumi Tanaka. 1999. The applications of unsupervised learning to Japanese grapheme-phoneme alignment. In *ACL'99 Workshop on Unsupervised Learning in Natural Language Processing*, pages 9–16.

Masanobu Higashida. 1994. A fully automated directory assistance service that accommodates degenerated keyword input via telephones. In *Pacific Telecommunication Conference*, pages 167–174.

Akiko Konno and Yasuo Hongo. 1993. Postprocessing algorithm based on the probabilistic and semantic method for Japanese OCR. In *Proceedings of ICDAR-93*, pages 646–649.

Hiroki Mori, Hirotomo Aso, and Shozo Makino. 1996. Robust n-gram model of Japanese character and its application to document recognition. *IEICE Transactions on Information and Systems*, E79-D(5):471–476.

Masaaki Nagata. 1994. A stochastic Japanese morphological analyzer using a forward-dp backward-a^* n-best search algorithm. In *COLING-94*, pages 201–207.

Masaaki Nagata. 1996. Context-based spelling correction for Japanese OCR. In *COLING-96*, pages 806–811.

Masaaki Nagata. 1998. Japanese OCR error correction using character shape similarity and statistical language model. In *COLING-ACL'98*, pages 922–928.

Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transaction on Information Theory*, 37(4):1085–1094.