An Ensemble Model of Word-based and Character-based Models for Japanese and Chinese Input Method

Yoh OKUNO[1] *Shinsuke MORI*[2]
(1) SWIFTKEY, 91-95 Southwark Bridge Road, London, SE1 0AX, United Kingdom
(2) KYOTO UNIVERSITY, Yoshidahonmachi, Sakyo-ku, Kyoto, 606-8501, Japan
yoh@swiftkey.net, mori@ar.media.kyoto-u.ac.jp

ABSTRACT

Since Japanese and Chinese languages have too many characters to be input directly using a standard keyboard, input methods for these languages that enable users to input the characters are required. Recently, input methods based on statistical models have become popular because of their accuracy and ease of maintenance. Most of them adopt word-based models because they utilize word-segmented corpora to train the models. However, such word-based models suffer from unknown words because they cannot convert words correctly which are not in corpora. To handle this problem, we propose a character-based model that enables input methods to convert unknown words by exploiting character-aligned corpora automatically generated by a monotonic alignment tool. In addition to the character-based model, we propose an ensemble model of both character-based and word-based models to achieve higher accuracy. The ensemble model combines these two models by linear interpolation. All of these models are based on joint source channel model to utilize rich context through higher order joint n-gram. Experiments on Japanese and Chinese datasets showed that the character-based model performs reasonably and the ensemble model outperforms the word-based baseline model. As a future work, the effectiveness of incorporating large raw data should be investigated.

KEYWORDS: Input Method, Machine Transliteration, Joint Source Channel Model, Automatic Alignment, Ensemble Method, Japanese, Chinese.

1 Introduction

There are more than 6,000 basic Kanji characters and 50 Hiragana/Katakana characters in Japanese language. A Kanji character represents one meaning, while Hiragana/Katakana characters represent their sounds. Therefore, it is difficult to input all kind of Japanese texts into computers or mobile phones by a standard keyboard which has only 100 keys. In order to input Japanese texts, it is common to use input methods called Kana-Kanji conversion, which convert Hiragana characters into Kanji or mixed characters. Since there are no spaces between words, most of Japanese input methods process texts sentence by sentence. Chinese language has nearly the same problem. There are more than 10,000 Hanzi characters in Chinese and Pinyin input methods are used to convert Roman characters into Chinese characters.

In these days, statistical models are used for such input methods to achieve high accuracy and automate parameter tuning (Mori et al., 1999; Chen and Lee, 2000). The statistical models are trained before actual conversion from corpora in each language. Sentences in these corpora are segmented word by word and annotated to specify the words' pronunciation, whether manually or automatically. Most of the models treat a word as an atomic unit; that means, they distinguish all words completely even if they share some characters in their strings. In this paper, we call such approaches *word-based* models.

However, such word-based models suffer from unknown words in principle, because they cannot convert or even enumerate a word in the candidate list when the word is not contained in the corpora. Instead of word-based models, we propose a new *character-based* model for input methods to avoid such a problem. In addition, we also propose an *ensemble model* which is a combination of both word-based and character-based models to achieve higher accuracy and take advantages of both models.

The rest of this paper is organized as follows: section 2 introduces related work, section 3 proposes the models, section 4 describes experimental results, and section 5 summarizes this paper and future work.

2 Related Work

There is a limited number of studies specialized in statistical models for input methods. However, closely related tasks such as machine transliteration, letter-to-phoneme conversion, language modeling, or machine translation share similar problems and solutions with input methods.

Early models for input methods adopt noisy channel model (Mori et al., 1999; Chen and Lee, 2000), which are less accurate than joint source channel model (Li et al., 2004). Conventionally, noisy channel model is used to divide joint model into conditional model and language model so that language model can be trained from large raw data such as crawled websites. Joint source channel model can also be combined such large raw data, though it is remained as a future work. In this paper, we focus on models for standard annotated corpora to keep the study reproducible and comparable with other works.

In noisy channel model, character n-gram is used as a back-off model for word n-gram model to address unknown word problem (Mori et al., 2006; Gao et al., 2002). However, character n-gram model does not exploit rich information of joint n-gram of word and its pronunciation. Moreover, it is not straightforward to extend character n-gram model to character-based joint n-gram model without recent development in alignment techniques we use (Jiampojamarn et al., 2007; Kubo et al., 2011).

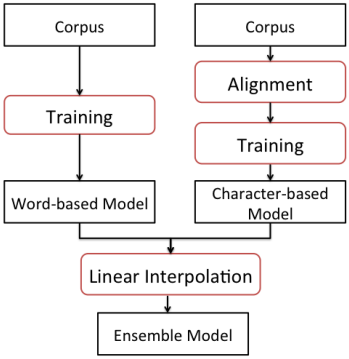

Figure 1: An Ensemble Model of Word-based and Character-based Models

Recently, discriminative models are introduced to input methods (Tokunaga et al., 2011; Jiampojamarn et al., 2008; Cherry and Suzuki, 2009), but they are impractical for higher order n-gram because of their large model size and long training time. Spelling correction is commonly incorporated with Chinese input methods (Zheng et al., 2011; Suzuki and Gao, 2012).

Machine transliteration is a similar task to input method, which translates proper names into foreign languages based on their sounds (Zhang et al., 2012). Machine transliteration is formulated as monotonic machine translation that is purely based on characters rather than words (Finch and Sumita, 2008). According to the manner of statistical machine translation (SMT), automatic alignment is applied to estimate character alignment between source and target strings (Jiampojamarn et al., 2007; Kubo et al., 2011).

Hatori and Suzuki (2011) solved Japanese pronunciation inference combining word-based and character-based features within SMT-style framework to handle unknown words. Neubig et al. (2012) proposed character-based SMT to incorporate word segmentation and handle sparsity. They solved different problems using similar solutions.

3 An Ensemble Model of Word-based and Character-based Models

We propose an ensemble model as a combination of word-based and character-based models. Figure 1 shows the training process for our model. The left side shows word-based model, while the right side shows character-based model. The difference between word-based model and character-based model is that later has an alignment step to produce character-aligned corpus by an automatic alignment tool. The two models are combined using linear interpolation to produce the final ensemble model.

Our model is built on top of joint source channel model (Li et al., 2004), which is an improvement on noisy channel model. In this section, we explain noisy channel model first, and joint source channel model next. Then the word-based model, the character-based model, and the ensemble model are explained.

3.1 Noisy Channel Model

A common statistical approach for input method is called noisy channel model, which models a conditional probability of output words given input strings to prioritize output candidates. The model decomposes the conditional distribution into language model and input model by Bayes rule:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \propto P(Y)P(X|Y) \tag{1}$$

Y is the output sequence and X is the input sequence. Language model $P(Y)$ stands for how likely the output is in the language, while input model $P(X|Y)$ stands for how proper the pronunciation is.

A standard language model is n-gram model, which utilizes contexts of length $n-1$ to predict the current word y_i.

$$P(Y) = \prod_{i=1}^{L} P(y_i|y_{i-n+1}^{i-1}) \tag{2}$$

y_i is the i-th output word, x_i is corresponding input string, L is the length of output sequence, n is the order of n-gram model, and y_i^j is a output sequence from i to j.

Input model is usually simple unigram or uniform distribution assigned to possible pronunciations.

$$P(X|Y) = \prod_{i=1}^{L} P(x_i|y_i) \tag{3}$$

However, it has a problem to ignore context around the word, leading low accuracy in conversion.

3.2 Joint Source Channel Model

Joint source channel model (Li et al., 2004) adopts a joint distribution rather than conditional distribution;

$$P(Y|X) = \frac{P(X,Y)}{P(X)} \propto P(X,Y) \tag{4}$$

The joint distribution is modeled as joint n-gram sequence.

$$P(X,Y) = \prod_{i=1}^{L} P(x_i, y_i|x_{i-n+1}^{i-1}, y_{i-n+1}^{i-1}) \tag{5}$$

This enables exploiting rich context of joint distribution to achieve fine-grained joint model.

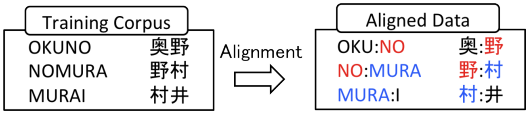

Figure 2: Alignment Step for Character-based Model

We adopt modified Kneser-Ney smoothing (Kneser and Ney, 1995) because it performed best in our experiment. *SRILM* (Stolcke, 2002) [1], which is a language model toolkit, is used for training n-gram models.

3.3 Word-based Model

In this study, we adopt the word-based model as a baseline model. The word-based model is trained from a word-segmented corpus. That means, the corpus is segmented word by word, and each word is annotated to specify its pronunciation. Each word and its pronunciation in the corpus are coupled into a unit to train a joint n-gram model.

This baseline model is strong enough if the corpus is properly segmented and annotated. It works well for words which are contained in the corpus. The ambiguity in homonyms are solved using their contexts through joint source channel model. However, the word-based model suffers from unknown words because it cannot properly handle words that are not in the corpus.

3.4 Character-based Model

In order to overcome the shortcomings of the word-based model, we propose a character-based model that is trained on character-aligned corpus. Since we do not have such a corpus, we need to produce a character-aligned corpus from a word-segmented corpus automatically. Though it is not trivial, recent development on character alignment tools enables it.

We adopted an alignment tool called *mpaligner* (Kubo et al., 2011) [2], which is suitable for this purpose. It assigns pronunciation to each character based on expectation maximization (EM) algorithm. Figure 2 shows how the alignment step works. Basically, it finds alignments between a character and its pronunciation based on co-occurrence in the corpus.

A word in the corpus is monotonically aligned one by one. That means, any words or pronunciations cannot be aligned across their word boundaries in the original corpus. Once the corpus is aligned, the training step is exactly same to the word-based model.

In addition to character alignment, mpaligner has a feature called many-to-many alignment that can find words whose pronunciations cannot be divided into combination of pronunciations for each character. This feature is effective especially in Japanese since Japanese language sometimes add its own pronunciation to Chinese words. For this reason, the result is not purely character-aligned, but most of characters are aligned to each pronunciation.

In Japanese and Chinese languages, it is common that a word contains only one character. Such a word is called a single character word. Most of words are consists of single character words and their pronunciations, while some words can not be represented as a combination of single character words because of its phonetic variation.

[1] http://www.speech.sri.com/projects/srilm/
[2] http://sourceforge.jp/projects/mpaligner/

Word	今日	大人	北京	一〇
Pronunciation	きょう	おとな	ぺきん	じゅう
Type	Combined	Combined	Combined	Combined

Word	日　本	自　分	情　報	関　係
Pronunciation	に　ほん	じ　ぶん	じょう　ほう	かん　けい
Type	Split	Split	Split	Split

Word	日 本	結　婚	三　菱	会　社
Pronunciation	にっ　ぽん	けっ　こん	みつ　びし	がい　しゃ
Type	Variation	Variation	Variation	Variation

Figure 3: Alignment Examples

Character alignment step reveals such phonetic variation of character which is not contained as a single character word. Pronunciation of a character can be changed when it is used in a word. When an unknown word contains such a pronunciation, the character-based model enables it to be converted correctly.

Figure 3 shows the examples of alignment result in Japanese. It shows pairs of word and its pronunciation with alignment type. Word and pronunciation are separated with spaces if the aligner split the word into characters. Each alignment type corresponds to features explained above; *combined* means that the word has unique pronunciation so it can not split into characters, *split* means that the word is split into characters and their pronunciations, and *variation* means that the word contains phonetic variation which does not appear in single character words.

3.5 Ensemble Model

Although the character-based model can capture unknown words, it achieves relatively poor compared to the word-based model in our experiment. There are two reasons why the character-based model does not work well. First, it tends to overfit to the training data because the corpus is segmented more finely. Second, the errors caused in the alignment step can be problematic.

In order to achieve higher accuracy, we propose an ensemble model which is linear interpolation of word-based model $P_w(X,Y)$ and character-based model $P_c(X,Y)$.

$$P(X,Y) = \alpha P_w(X,Y) + (1-\alpha)P_c(X,Y) \tag{6}$$

The interpolation weight α ($0 \leq \alpha \leq 1$) means the ratio in which model is used. $\alpha = 1$ means pure word-based model, while $\alpha = 0$ means pure character-based model. $\alpha = 0.5$ is nearly equivalent to the model trained from corpora which is a concatenation of original corpus and character-aligned corpus.

α is determined empirically. In our experiment, α has an optimal value between 0.5 and 0.7. That means, both word-based and character based models are complementary to each other and the ratio of these two models should be a bit closer to word-based model.

Domain	Sentences	Data Source
OC	6,266	Yahoo! Q&A
OW	4,080	Government Document
OY	6,253	Yahoo! Blog
PN	11,582	Newspaper
PB	6,645	Book
PM	9,356	Magazine
ALL	44,182	All of above

Table 1: Details of BCCWJ

4 Experiment

To confirm the effectiveness and properties of our models, we conducted experiments on Japanese and Chinese corpora in various situations. We divided each corpus into 90% of training data and 10% of test data, trained our models and evaluated on test data. The models are evaluated by comparing system output and gold standard; system output is produced by a decoder which is an implementation of Viterbi algorithm for higher order n-gram models [3].

4.1 Data Set

For Japanese corpus, we adopt BCCWJ (Balanced Corpus of Contemporary Written Japanese) (Maekawa, 2008). BCCWJ is a corpus in various domains. It is annotated both by human and machine learning algorithm. We use human-annotated part which consists of 44,182 sentences. Table 1 shows the details.

For Chinese corpus, we adopt LCMC (The Lancaster Corpus of Mandarin Chinese) (McEnery et al., 2003). It has Hanzi and Pinyin pairs, but the Pinyin part of the corpus is annotated automatically. It contains 45,595 lines from 15 domains.

4.2 Evaluation Metric

We adopt evaluation metrics based on the longest common sequence (LCS) between system output and gold standard following (Mori et al., 1999) and (Tokunaga et al., 2011).

$$precision = \frac{N_{LCS}}{N_{SYS}} \tag{7}$$

$$recall = \frac{N_{LCS}}{N_{DAT}} \tag{8}$$

$$F\text{-}score = 2\frac{precision \cdot recall}{precision + recall} \tag{9}$$

Here, N_{LCS} is the length of the LCS, N_{SYS} is the length of the system output sequence, N_{DAT} is the length of gold standard. Note that LCS is not necessarily a continuous string contained in both string; that means, LCS can be concatenation of separated strings which are contained in both strings in order. CER (Character Error Rate) and ACC (Sentence Accuracy) are also shown for convenience, but F-score is used as our main metric.

[3] https://github.com/nokuno/jsc

Model	N	Precision	Recall	F-score	CER	ACC	Size
Word	2	0.932	0.932	0.932	0.088	0.334	4.3MB
Char	4	0.925	0.922	0.923	0.099	0.292	3.1MB
Ensemble	3	0.937	0.936	0.937	0.082	0.349	8.7MB

Table 2: Result for Japanese

Model	N	Precision	Recall	F-score	CER	ACC	Size
Word	2	0.958	0.958	0.958	0.044	0.505	4.6MB
Char	4	0.950	0.950	0.950	0.053	0.428	3.3MB
Ensemble	3	0.958	0.958	0.958	0.045	0.496	8.8MB

Table 3: Result for Chinese with tone

Model	N	Precision	Recall	F-score	CER	ACC	Size
Word	3	0.895	0.895	0.895	0.109	0.297	5.1MB
Char	3	0.871	0.871	0.871	0.133	0.210	3.3MB
Ensemble	4	0.895	0.895	0.895	0.111	0.274	8.7MB

Table 4: Result for Chinese without tone

4.3 Result Summary

Table 2, Table 3, Table 4 show the summary of our experiments to compare three models; word-based, character-based and ensemble models. The value of n is chosen to perform the best in terms of F-score.

In Japanese language, the word-based model outperforms the character-based model, and the ensemble model outperforms the word-based model consistently in all metrics.

In LCMC corpus, we could not find any improvement in the ensemble model over the word-based model. This is reasonable because the corpus is automatically annotated word by word. In our experiment, we found tone (1-4 digits) information reduces the ambiguity of pinyin input method greatly. Rest of experiments are conducted on Japanese corpus.

Model size and decoding time depend on the implementation of decoder which is not focus in this study, but we showed file size for each model in SRILM binary format. We can see that character-based model is smaller than word-based model whereas ensemble model is bigger than word-based model.

4.4 N-gram Order

Figure 4 shows F-score for three models with various n values from 1 to 10. It is notable that the character-based model performs best when $n = 4$ or larger, while $n = 2$ is enough for the word-based model. This shows that the character-based model requires longer context than the word-based model because it splits words into shorter characters. The ensemble model performs best when $n = 3$ which is middle of word-based and character-based models. In all models, higher order than the best order did not degrade the performance under the modified Kneser-Ney smoothing.

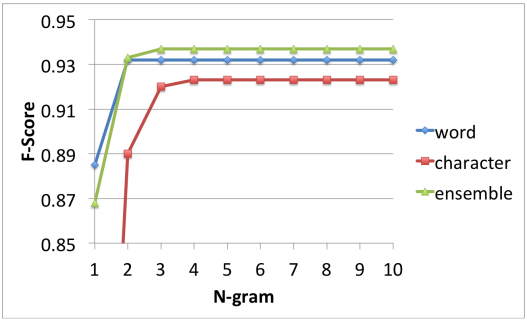

Figure 4: N-gram Order

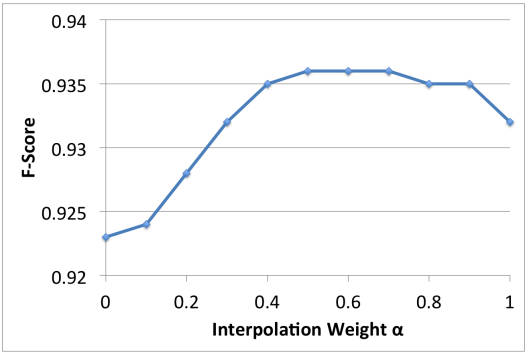

Figure 5: Interpolation Weight

4.5 Interpolation Weight

To investigate the effect of interpolation weight, we changed α from 0 (pure character-based) to 1 (pure word-based) by interval of 0.1. The result in Figure 5 shows that the weight from 0.5 to 0.7 is optimal in this case. That means, the two models are complementary to each other and the word-based model can be ameliorated by mixing the character-based model.

4.6 Cross Domain Analysis

In practice, it is important to choose right domain for training corpus. Table 5 shows F-score when the categories in training corpus and test corpus are different. To compare domains fairly, the smallest size of corpus is adopted; that means, we used only 4,080 sentences from each corpus. Therefore, the absolute F-score is relatively low. As we expected, the accuracy is the highest when the domain of training and testing corpus is the same. In addition, we can see that clean corpora such as newspaper or magazines can outperform corpus from web. It is possible to combine large general corpus and small but specific corpus to achieve more higher accuracy.

4.7 Smoothing Methods

Table 6 shows F-score of different smoothing methods. In our experiment, modified Kneser-Ney smoothing performed better than other smoothing methods.

Train | Test	OC	OW	OY	PB	PM	PN
OC	0.869	0.744	0.774	0.779	0.750	0.705
OW	0.749	0.966	0.701	0.759	0.717	0.747
OY	0.822	0.782	0.846	0.791	0.755	0.746
PB	0.807	0.761	0.755	0.902	0.754	0.740
PM	0.837	0.811	0.794	0.828	0.876	0.764
PN	0.812	0.848	0.762	0.820	0.783	0.867

Table 5: Cross Domain Analysis

Smoothing	Precision	Recall	F-score	CER	ACC
Modified Kneser-Ney	0.931	0.932	0.931	0.087	0.361
Witten-Bell	0.929	0.930	0.930	0.090	0.338
Absolute discounting	0.929	0.931	0.930	0.090	0.343
Ristad's natural discounting law	0.925	0.927	0.926	0.094	0.322
Add one smoothing	0.798	0.819	0.808	0.223	0.174

Table 6: Smoothing Methods

Pruning Threshold	F-score	1-gram size	2-gram size	3-gram size	File size
1e-4	0.808	345416	2232	144	15MB
1e-5	0.877	345416	27450	3345	15MB
1e-6	0.914	345416	222883	72087	17MB
1e-7	0.936	345416	1200021	833593	34MB
1e-8	0.942	345416	5286912	4146260	98MB

Table 7: Pruning Effect

4.8 Pruning Effect

In practical input methods, model size is important because the memory in a device is limited. In order to reduce model size, we applied entropy pruning (Stolcke, 2000) to word-based model. Table 7 shows the result of F-score, N-gram size and file size in the SRILM binary format for various thresholds. In this experiment, all data in BCCWJ including automatically annotated one is used to confirm the effectiveness of big data. The result shows practical tradeoff between model size and accuracy; more larger model might improve accuracy in the future.

4.9 Error Analysis

Figure 6 shows some examples of output from the ensemble model and the word-based model, and correct output. There are some typical cases where the ensemble model outperforms the word-based model: 1) casual expression, 2) foreign words, 3) number expression.

Last four lines show the samples where the ensemble model did not work well. In these examples, the character-based model breaks the correct result. From these analysis, we can see the effectiveness of our ensemble model not only in unknown words, but also sparseness of training corpus. In most cases, the ensemble model showed consistent results in one sentence while the word-based model break the consistency because of its sparseness.

Type	Sentence
Ensemble	ブレイしまいまま引退しそうって話です。
Word	ブレイ姉妹ママ引退しそうって話です。
Ensemble	騙されたと思ってやってみてちょ＾＾
Word	騙されたと思ってやってみて著＾＾
Ensemble	レオナルド・ディカプリオ。
Word	レオなるド・ディかぶ理生。
Ensemble	パーカッションミュージアム
Word	パー買っ所ん見ゅー時ア無
Ensemble	利用率は７３．２％
Word	利用率は七十三・２％
Ensemble	メールモダメ。
Correct	メールもダメ。
Ensemble	ハリウッてもらったら、
Correct	針打ってもらったら、

Figure 6: Error Analysis

Conclusion

In this paper, we proposed an ensemble model of word-based and character-based models. The character-based model exploit a character alignment tool to aquire fine-grained model. The expriments showed that the ensemble model outperforms the word-based model and character-based model performs modestly. The optimal *n* for the character-based model and the ensemble model was longer than word-based model. The optimal interpolation weight for ensemble model was 0.5 to 0.7, which is close to the word-based model. As a future work, the effectiveness of unannotated corpora such as crawled web pages should be confirmed. In practice, it is important to integrate various corpora into single model. It is possible to apply discriminative models to character-based model or ensemble model if we can train and decode the model for higher order n-gram features effectively.

References

Chen, Z. and Lee, K.-F. (2000). A new statistical approach to chinese pinyin input. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 241–247, Hong Kong. Association for Computational Linguistics.

Cherry, C. and Suzuki, H. (2009). Discriminative substring decoding for transliteration. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1075, Singapore. Association for Computational Linguistics.

Finch, A. and Sumita, E. (2008). Phrase-based machine transliteration. In *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*, pages 13–18, Hyderabad, India.

Gao, J., Goodman, J., Li, M., and Lee, K.-F. (2002). Toward a unified approach to statistical language modeling for chinese. 1(1):3–33.

Hatori, J. and Suzuki, H. (2011). Japanese pronunciation prediction as phrasal statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 120–128, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Jiampojamarn, S., Cherry, C., and Kondrak, G. (2008). Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 905–913, Columbus, Ohio. Association for Computational Linguistics.

Jiampojamarn, S., Kondrak, G., and Sherif, T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York. Association for Computational Linguistics.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Kubo, K., Kawanami, H., Saruwatari, H., and Shikano, K. (2011). Unconstrained many-to-many alignment for automatic pronunciation annotation. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011 (APSIPA2011)*, Xi'an, China.

Li, H., Zhang, M., and Su, J. (2004). A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 159–166, Barcelona, Spain.

Maekawa, K. (2008). Balanced corpus of contemporary written japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pages 101–102.

McEnery, A., Xiao, Z., and Mo, L. (2003). Aspect marking in english and chinese: using the lancaster corpus of mandarin chinese for contrastive language study. *Literary and Linguistic Computing*, 18(4):361–378.

Mori, S., Masatoshi, T., Yamaji, O., and Nagao, M. (1999). Kana-kanji conversion by a stochastic model (in japanese). *Transactions of IPSJ*, 40(7):2946–2953.

Mori, S., Takuma, D., and Kurata, G. (2006). Phoneme-to-text transcription system with an infinite vocabulary. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 729–736, Stroudsburg, PA, USA. Association for Computational Linguistics.

Neubig, G., Watanabe, T., Mori, S., and Kawahara, T. (2012). Machine translation without words through substring alignment. In *The 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 165–174, Jeju, Korea.

Stolcke, A. (2000). Entropy-based pruning of backoff language models. *arXiv preprint cs/0006025*.

Stolcke, A. (2002). Srilm-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*.

Suzuki, H. and Gao, J. (2012). A unified approach to transliteration-based text input with online spelling correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 609–618, Jeju Island, Korea. Association for Computational Linguistics.

Tokunaga, H., Okanohara, D., and Mori, S. (2011). Discriminative method for japanese kana-kanji input method. In *Proceedings of the Workshop on Advances in Text Input Methods (WTIM 2011)*, pages 10–18, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Zhang, M., Kumaran, A., and Li, H. (2012). Whitepaper of news 2012 shared task on machine transliteration. In *Proceedings of the 4th Named Entities Workshop (NEWS 2012)*, Jeju, Korea. The Association of Computational Linguistics.

Zheng, Y., Li, C., and Sun, M. (2011). Chime: An efficient error-tolerant chinese pinyin input method. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*.