Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation

Yan Qu
Clairvoyance Corporation
5001 Baum Boulevard, Suite 700
Pittsburgh, PA 15213-1854, USA
yqu@clairvoyancecorp.com

Gregory Grefenstette∗
LIC2M/LIST/CEA
18, route du Panorama, BP 6
Fontenay-aux-Roses, 92265 France
Gregory.Grefenstette@cea.fr

Abstract

Multilingual applications frequently involve dealing with proper names, but names are often missing in bilingual lexicons. This problem is exacerbated for applications involving translation between Latin-scripted languages and Asian languages such as Chinese, Japanese and Korean (CJK) where simple string copying is not a solution. We present a novel approach for generating the ideographic representations of a CJK name written in a Latin script. The proposed approach involves first identifying the origin of the name, and then back-transliterating the name to all possible Chinese characters using language-specific mappings. To reduce the massive number of possibilities for computation, we apply a three-tier filtering process by filtering first through a set of attested bigrams, then through a set of attested terms, and lastly through the WWW for a final validation. We illustrate the approach with English-to-Japanese back-transliteration. Against test sets of Japanese given names and surnames, we have achieved average precisions of 73% and 90%, respectively.

1 Introduction

Multilingual processing in the real world often involves dealing with proper names. Translations of names, however, are often missing in bilingual resources. This absence adversely affects multilingual applications such as machine translation (MT) or cross language information retrieval (CLIR) for which names are generally good discriminating terms for high IR performance (Lin *et al.*, 2003). For language pairs with different writing systems, such as Japanese and English, and for which simple string-copying of a name from one language to another is not a solution, researchers have studied techniques for transliteration, i.e., phonetic translation across languages. For example, European names are often transcribed in Japanese using the syllabic katakana alphabet. Knight and Graehl (1998) used a bilingual English-katakana dictionary, a katakana-to-English phoneme mapping, and the CMU Speech Pronunciation Dictionary to create a series of weighted finite-state transducers between English words and katakana that produce and rank transliteration candidates. Using similar methods, Qu *et al.* (2003) showed that integrating automatically discovered transliterations of unknown katakana sequences, i.e. those not included in a large Japanese-English dictionary such as EDICT[1], improves CLIR results.

Transliteration of names between alphabetic and syllabic scripts has also been studied for languages such as Japanese/English (Fujii & Ishikawa, 2001), English/Korean (Jeong et al., 1999), and English/Arabic (Al-Onaizan and Knight, 2002).

In work closest to ours, Meng *et al* (2001), working in cross-language retrieval of phonetically transcribed spoken text, studied how to transliterate names into Chinese phonemes (though not into Chinese characters). Given a list of identified names, Meng *et al*. first separated the names into Chinese names and English names. Romanized Chinese names were detected by a left-to-right longest match segmentation method, using the Wade-Giles[2] and the pinyin syllable inventories in sequence. If a name could be segmented successfully, then the name was considered a Chinese name. As their spoken document collection had already been transcribed into pinyin, retrieval was based on pinyin-to-pinyin matching; pinyin to Chinese character conversion was not addressed. Names other than Chinese names were considered as foreign names and were converted into Chinese phonemes using a language model derived from a list of English-Chinese equivalents, both sides of which were represented in phonetic equivalents.

∗ The work was done by the author while at Clairvoyance Corporation.

1 http://www.csse.monash.edu.au/~jwb/edict.html

2 http://lcweb.loc.gov/catdir/pinyin/romcover.html

The above English-to-Japanese or English-to-Chinese transliteration techniques, however, only solve a part of the name translation problem. In multilingual applications such as CLIR and Machine Translation, all types of names must be translated. Techniques for name translation from Latin scripts into CJK scripts often depend on the origin of the name. Some names are not transliterated into a nearly deterministic syllabic script but into ideograms that can be associated with a variety of pronunciations. For example, Chinese, Korean and Japanese names are usually written using Chinese characters (or *kanji*) in Japanese, while European names are transcribed using katakana characters, with each character mostly representing one syllable.

In this paper, we describe a method for converting a Japanese name written with a Latin alphabet (or *romanji*), back into Japanese kanji[3]. Transcribing into Japanese kanji is harder than transliteration of a foreign name into syllabic katakana, since one phoneme can correspond to hundreds of possible kanji characters. For example, the sound "kou" can be mapped to 670 kanji characters.

Our method for back-transliterating Japanese names from English into Japanese consists of the following steps: (1) language identification of the origins of names in order to know what language-specific transliteration approaches to use, (2) generation of possible transliterations using sound and kanji mappings from the Unihan database (to be described in section 3.1) and then transliteration validation through a three-tier filtering process by filtering first through a set of attested bigrams, then through a set of attested terms, and lastly through the Web.

The rest of the paper is organized as follows: in section 2, we describe and evaluate our name origin identifier; section 3 presents in detail the steps for back transliterating Japanese names written in Latin script into Japanese kanji representations; section 4 presents the evaluation setup and section 5 discusses the evaluation results; we conclude the paper in section 6.

2 Language Identification of Names

Given a name in English for which we do not have a translation in a bilingual English-Japanese dictionary, we first have to decide whether the name is of Japanese, Chinese, Korean or some European origin. In order to determine the origin of names, we created a language identifier for names, using a trigram language identification method (Cavner and Trenkle, 1994). During training, for Chinese names, we used a list of 11,416 Chinese names together with their frequency information[4]. For Japanese names, we used the list of 83,295 Japanese names found in ENAMDICT[5]. For English names, we used the list of 88,000 names found at the US. Census site[6]. (We did not obtain any training data for Korean names, so origin identification for Korean names is not available.) Each list of names[7] was converted into trigrams; the trigrams for each list were then counted and normalized by dividing the count of the trigram by the number of all the trigrams. To identify a name as Chinese, Japanese or English (Other, actually), we divide the name into trigrams, and sum up the normalized trigram counts from each language. A name is identified with the language which provides the maximum sum of normalized trigrams in the word. Table 1 presents the results of this simple trigram-based language identifier over the list of names used for training the trigrams.

The following are examples of identification errors: Japanese names recognized as English, e.g., *aa*, *abason*, *abire*, *aebakouson*; Japanese names recognized as Chinese, e.g., *abeseimei*, *abei*, *adan*, *aden*, *afun*, *agei*, *agoin*. These errors show that the language identifier can be improved, possibly by taking into account language-specific features, such as the number of syllables in a name. For origin detection of Japanese names, the current method works well enough for a first pass with an accuracy of 92%.

Input names	As JAP	As CHI	As ENG	Accuracy
Japanese	76816	5265	1212	92%
Chinese	1147	9947	321	87%
English	12115	14893	61701	70%

Table 1: Accuracy of language origin identification for names in the training set (JAP, CHI, and ENG stand for Japanese, Chinese, and English, respectively)

[3] We have applied the same technique to Chinese and Korean names, though the details are not presented here.

[4] http://www.geocities.com/hao510/namelist/

[5] http://www.csse.monash.edu.au/~jwb/enamdict_doc.html

[6] http://www.census.gov/genealogy/names

[7] Some names appear in multiple name lists: 452 of the names are found both in the Japanese name list and in the Chinese name list; 1529 names appear in the Japanese name list and the US Census name list; and 379 names are found both in the Chinese name list and the US Census list.

3 English-Japanese Back-Transliteration

Once the origin of a name in Latin scripts is identified, we apply language-specific rules for back-transliteration. For non-Asian names, we use a katakana transliteration method as described in (Qu et al., 2003). For Japanese and Chinese names, we use the method described below. For example, "koizumi" is identified as a name of Japanese origin and thus is back-transliterated to Japanese using Japanese specific phonetic mappings between romanji and kanji characters.

3.1 Romanji-Kanji Mapping

To obtain the mappings between kanji characters and their romanji representations, we used the Unihan database, prepared by the Unicode Consortium [8]. The Unihan database, which currently contains 54,728 kanji characters found in Chinese, Japanese, and Korean, provides rich information about these kanji characters, such as the definition of the character, its values in different encoding systems, and the pronunciation(s) of the character in Chinese (listed under the feature kMandarin in the Unihan database), in Japanese (both the *On* reading and the *Kun* reading [9]: kJapaneseKun and kJapaneseOn), and in Korean (kKorean). For example, for the kanji character 金, coded with Unicode hexadecimal character 91D1, the Unihan database lists 49 features; we list below its pronunciations in Japanese, Chinese, and Korean:

U+91D1 kJapaneseKun KANE
U+91D1 kJapaneseOn KIN KON
U+91D1 kKorean KIM KUM
U+91D1 kMandarin JIN1 JIN4

In the example above, 金 is represented in its Unicode scalar value in the first column, with a feature name in the second column and the values of the feature in the third column. The Japanese *Kun* reading of 金 is KANE, while the Japanese *On* readings of 金 is KIN and KON.

From the Unicode database, we construct mappings between Japanese readings of a character in romanji and the kanji characters in its Unicode representation. As kanji characters in Japanese names can have either the *Kun* reading or the *On* reading, we consider both readings as candidates for each kanji character. The mapping table has a total of 5,525 entries. A typical mapping is as follows:

kou U+4EC0 U+5341 U+554F U+5A09 U+5B58 U+7C50 U+7C58 .....

in which the first field specifies a pronunciation in romanji, while the rest of the fields specifies the possible kanji characters into which the pronunciation can be mapped.

There is a wide variation in the distribution of these mappings. For example, kou can be the pronunciation of 670 kanji characters, while the sound katakumi can be mapped to only one kanji character.

3.2 Romanji Name Back-Transliteration

In theory, once we have the mappings between romanji characters and the kanji characters, we can first segment a Japanese name written in romanji and then apply the mappings to back-transliterate the romanji characters into all possible kanji representations. However, for some segmentation, the number of the possible kanji combinations can be so large as to make the problem computationally intractable. For example, consider the short Japanese name "koizumi." This name can be segmented into the romanji characters "ko-i-zu-mi" using the Romanji-Kanji mapping table described in section 3.1, but this segmentation then has 182*230*73*49 (over 149 million) possible kanji combinations. Here, 182, 239, 73, and 49 represents the numbers of possible kanji characters for the romanji characters "ko", "i", "zu", and "mi", respectively.

In this study, we present an efficient procedure for back-transliterating romanji names to kanji characters that avoids this complexity. The procedure consists of the following steps: (1) romanji name segmentation, (2) kanji name generation, (3) kanji name filtering via monolingual Japanese corpus, and (4) kanji-romanji combination filtering via WWW. Our procedure relies on filtering using corpus statistics to reduce the hypothesis space in the last three steps. We illustrate the steps below using the romanji name "koizumi" (小泉).

3.2.1 Romanji Name Segmentation

With the romanji characters from the Romanji-Kanji mapping table, we first segment a name recognized as Japanese into sequences of romanji characters. Note that a greedy segmentation method, such as the left-to-right longest match method, often results in segmentation errors. For example, for "koizumi", the longest match segmentation method produces segmentation "koi-

[8] http://www.unicode.org/charts/unihan.html

[9] Historically, when kanji characters were introduced into the Japanese writing system, two methods of transcription were used. One is called "on-yomi" (i.e., *On* reading), where the Chinese sounds of the characters were adopted for Japanese words. The other method is called "kun-yomi" (i.e., *Kun* reading), where a kanji character preserved its meaning in Chinese, but was pronounced using the Japanese sounds.

zu-mi", while the correct segmentation is "ko-izumi".

Motivated by this observation, we generate all the possible segmentations for a given name. The possible segmentations for "koizumi" are:

ko-izumi
koi-zu-mi
ko-i-zu-mi

3.2.2 Kanji Name Segmentation

Using the same Romanji-Kanji mapping table, we obtain the possible kanji combinations for a segmentation of a romanji name produced by the previous step. For the segmentation "ko-izumi", we have a total of 546 (182*3) combinations (we use the Unicode scale value to represent the kanji characters and use spaces to separate them):

U+5C0F U+6CC9
U+53E4 U+6CC9
.....

We do not produce all possible combinations. As we have discussed earlier, such a generation method can produce so many combinations as to make computation infeasible for longer segmentations. To control this explosion, we eliminate unattested combinations using a bigram model of the possible kanji sequences in Japanese.

From the Japanese evaluation corpus of the NTCIR-4 CLIR track [10], we collected bigram statistics by first using a statistical part-of-speech tagger of Japanese (Qu et al., 2004). All valid Japanese terms and their frequencies from the tagger output were extracted. From this term list, we generated kanji bigram statistics (as well as an attested term list used below in step 3). With this bigram-based model, our hypothesis space is significantly reduced. For example, with the segmentation "ko-i-zu-mi", even though "ko-i" can have 182*230 possible combinations, we only retain the 42 kanji combinations that are attested in the corpus.

Continuing with the romanji segments "i-zu", we generate the possible kanji combinations for "i-zu" that can continue one of the 42 candidates for "ko-i". This results in only 6 candidates for the segments "ko-i-zu".

Lastly, we consider the romanji segments "zu-mi", and retain with only 4 candidates for the segmentation "ko-i-zu-mi" whose bigram sequences are attested in our language model:

U+5C0F U+53F0 U+982D U+8EAB
U+5B50 U+610F U+56F3 U+5B50
U+5C0F U+610F U+56F3 U+5B50
U+6545 U+610F U+56F3 U+5B50

Thus, for the segmentation "ko-i-zu-mi", the bigram-based language model effectively reduces the hypothesis space from 182*230*73*49 possible kanji combinations to 4 candidates. For the other alternative segmentation "koi-zu-mi", no candidates can be generated by the language model.

3.2.3 Corpus-based Kanji name Filtering

In this step, we use a monolingual Japanese corpus to validate whether the kanji name candidates generated by step (2) are attested in the corpus. Here, we simply use Japanese term list extracted from the segmented NTCIR-4 corpus created for the previous step to filter out unattested kanji combinations. For the segmentation "ko-izumi", the following kanji combinations are attested in the corpus (preceded by their frequency in the corpus):

4167 小泉 koizumi
16 古泉 koizumi
4 戸泉 koizumi

None of the four kanji candidates from the alternate segmentation "ko-i-zu-mi" is attested in the corpus. While step 2 filters out candidates using bigram sequences, step 3 uses corpus terms in their entirety to validate candidates.

3.2.4 Romanji-Kanji Combination Validation

Here, we take the corpus-validated kanji candidates (but for which we are not yet sure if they correspond to the same reading as the original Japanese name written in romanji) and use the Web to validate the pairings of kanji-romanji combinations (e.g., 小泉 AND koizumi). This is motivated by two observations. First, in contrast to monolingual corpus, Web pages are often mixed-lingual. It is often possible to find a word and its translation on the same Web pages. Second, person names and specialized terminology are among the most frequent mixed-lingual items. Thus, we would expect that the appearance of both representations in close proximity on the same pages gives us more confidence in the kanji representations. For example, with the Google search engine, all three kanji-romanji combinations for "koizumi" are attested:

23,600 pages --小泉 koizumi
302 pages --古泉 koizumi
1 page --戸泉 koizumi

Among the three, the 小泉 koizumi combination is the most common one, being the name of the current Japanese Prime Minister.

[10] http://research.nii.ac.jp/ntcir-ws4/clir/index.html

4 Evaluation

In this section, we describe the gold standards and evaluation measures for evaluating the effectiveness of the above method for back-transliterating Japanese names.

4.1 Gold Standards

Based on two publicly accessible name lists and a Japanese-to-English name lexicon, we have constructed two Gold Standards. The Japanese-to-English name lexicon is ENAMDICT[11], which contains more than 210,000 Japanese-English name translation pairs.

Gold Standard – Given Names (GS-GN): to construct a gold standard for Japanese given names, we obtained 7,151 baby names in romanji from http://www.kabalarians.com/. Of these 7,151 names, 5,115 names have kanji translations in the ENAMDICT[12]. We took the 5115 romanji names and their kanji translations in the ENAMDICT as the gold standard for given names.

Gold Standard – Surnames (GS-SN): to construct a gold standard for Japanese surnames, we downloaded 972 surnames in romanji from http://business.baylor.edu/Phil_VanAuken/JapaneseSurnames.html. Of these names, 811 names have kanji translations in the ENAMDICT. We took these 811 romanji surnames and their kanji translations in the ENAMDICT as the gold standard for Japanese surnames.

4.2 Evaluation Measures

Each name in romanji in the gold standards has at least one kanji representation obtained from the ENAMDICT. For each name, *precision*, *recall*, and *F* measures are calculated as follows:

- *Precision*: number of correct kanji output / total number of kanji output
- *Recall*: number of correct kanji output / total number of kanji names in gold standard
- *F-measure*: 2*Precision*Recall / (Precision + Recall)

Average Precision, *Average Recall*, and *Average F-measure* are computed over all the names in the test sets.

5 Evaluation Results and Analysis

5.1 Effectiveness of Corpus Validation

Table 2 and Table 3 present the precision, recall, and F statistics for the gold standards GS-GN and GS-SN, respectively. For given names, corpus validation produces the best average precision of 0.45, while the best average recall is a low 0.27. With the additional step of Web validation of the romanji-kanji combinations, the average precision increased by 62.2% to 0.73, while the best average recall improved by 7.4% to 0.29. We observe a similar trend for surnames. The results demonstrate that, through a large, mixed-lingual corpus such as the Web, we can improve both precision and recall for automatically transliterating romanji names back to kanji.

	Avg Prec	**Avg Recall**	**F**
(1) Corpus	0.45	0.27	0.33
(2) Web (over (1))	0.73 (+62.2%)	0.29 (+7.4%)	0.38 (+15.2%)

Table 2: The best Avg Precision, Avg Recall, and Avg F statistics achieved through corpus validation and Web validation for GS-GN.

	Avg Prec	**Avg Recall**	**F**
(1) Corpus	0.69	0.44	0.51
(2) Web (over (1))	0.90 (+23.3%)	0.45 (+2.3%)	0.56 (+9.8%)

Table 3: The best Avg Precision, Avg Recall, and Avg F statistics achieved through corpus validation and Web validation for GS-SN.

We also observe that the performance statistics for the surnames are significantly higher than those of the given names, which might reflect the different degrees of flexibility in using surnames and given names in Japanese. We would expect that the surnames form a somewhat closed set, while the given names belong to a more open set. This may account for the higher recall for surnames.

5.2 Effectiveness of Corpus Validation

If the big, mixed-lingual Web can deliver better validation than the limited-sized monolingual corpus, why not use it at every stage of filtering? Technically, we could use the Web as the ultimate corpus for validation at any stage when a corpus is required. In practice, however, each Web access involves additional computation time for file IO, network connections, etc. For example, accessing Google took about 2 seconds per name[13]; gathering

[11] http://mirrors.nihongo.org/monash/enamdict_doc.html

[12] The fact that above 2000 of these names were missing from ENAMDICT is a further justification for a name translation method as described in this paper.

[13] We inserted a 1 second sleep between calls to the search engine so as not to overload the engine.

statistics for about 30,000 kanji-romanji combinations[14] took us around 15 hours.

In the procedure described in section 3.2, we have aimed to reduce computation complexity and time at several stages. In step 2, we use bigram-based language model from a corpus to reduce the hypothesis space. In step 3, we use corpus filtering to obtain a fast validation of the candidates, before passing the output to the Web validation in step 4. Table 4 illustrates the savings achieved through these steps.

	GS-GN	GS-SN
All possible	2.0e+017	296,761,622,763
2gram model	21,306,322 (-99.9%)	2,486,598 (-99.9%)
Corpus validate	30,457 (-99.9%)	3,298 (-99.9%)
Web validation	20,787 (-31.7%)	2,769 (-16.0%)

Table 4: The numbers of output candidates of each step to be passed to the next step. The percentages specify the amount of reduction in hypothesis space.

5.3 Thresholding Effects

We have examined whether we should discard the validated candidates with low frequencies either from the corpus or the Web. The cutoff points examined include initial low frequency range 1 to 10 and then from 10 up to 400 in with increments of 5. Figure 1 and Figure 2 illustrate that, to achieve best overall performance, it is beneficial to discard candidates with very low frequencies, e.g., frequencies below 5. Even though we observe a stabling trend after reaching certain threshold points for these validation methods, it is surprising to see that, for the corpus validation method with GS-GN, with stricter thresholds, average precisions are actually decreasing. We are currently investigating this exception.

5.4 Error Analysis

Based on a preliminary error analysis, we have identified three areas for improvements.

First, our current method does not account for certain phonological transformations when the *On*/*Kun* readings are concatenated together. Consider the name "matsuda" (松田). The segmentation step correctly segmented the romanji to "matsu-da". However, in the Unihan database, the *Kun* reading of 田 is "ta", while its *On* reading is "den". Therefore, using the mappings from the Unihan database, we failed to obtain the mapping between the pronunciation "da" and the kanji 田, which resulted in both low precision and recall for "matsuda". This suggests for introducing language-specific phonological transformations or alternatively fuzzy matching to deal with the mismatch problem.

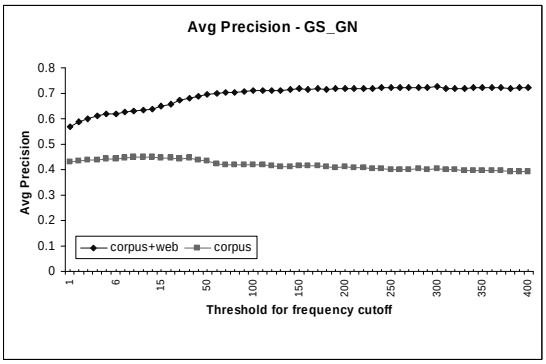

Figure 1: Average precisions achieved via both corpus and corpus+Web validation with different frequency-based cutoff thresholds for GS-GN

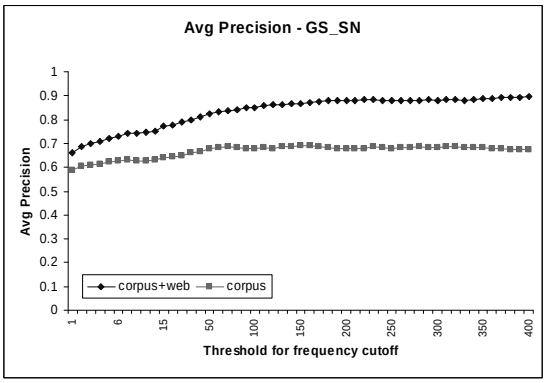

Figure 2: Average precisions achieved via both corpus and corpus+Web validation with different frequency-based cutoff thresholds for GS-SN

Second, ENAMDICT contains mappings between kanji and romanji that are not available from the Unihan database. For example, for the name "hiroshi" in romanji, based on the mappings from the Unihan database, we can obtain two possible segmentations: "hiro-shi" and "hi-ro-shi". Our method produces two- and three-kanji character sequences that correspond to these romanji characters. For example, corpus validation produces the following kanji candidates for "hiroshi":

[14] At this rate, checking the 21 million combinations remaining after filtering with bigrams using the Web (without the corpus filtering step) would take more than a year.

2 尋之 hiroshi
10 尋史 hiroshi
5 尋子 hiroshi
1 尋志 hiroshi
2 比呂史 hiroshi
11 比呂司 hiroshi
33 比呂子 hiroshi
311 比呂志 hiroshi

ENAMDCIT, however, in addition to the 2- and 3-character kanji names, also contains 1-character kanji names, whose mappings are not found in the Unihan database, e.g.,

宇 Hiroshi
演 Hiroshi
央 Hiroshi
海 Hiroshi
完 Hiroshi
寛 Hiroshi

This suggests the limitation of relying solely on the Unihan database for building mappings between romanji characters and kanji characters. Other mapping resources, such as ENAMDCIT, should be considered in our future work.

Third, because the statistical part-of-speech tagger we used for Japanese term identification does not have a lexicon of all possible names in Japanese, some unknown names, which are incorrectly separated into individual kanji characters, are therefore not available for correct corpus-based validation. We are currently exploring methods using overlapping character bigrams, instead of the tagger-produced terms, as the basis for corpus-based validation and filtering.

6 Conclusions

In this study, we have examined a solution to a previously little treated problem of transliterating CJK names written in Latin scripts back into their ideographic representations. The solution involves first identifying the origins of the CJK names and then back-transliterating the names to their respective ideographic representations with language-specific sound-to-character mappings. We have demonstrated that a simple trigram-based language identifier can serve adequately for identifying names of Japanese origin. During back-transliteration, the possibilities can be massive due to the large number of mappings between a Japanese sound and its kanji representations. To reduce the complexity, we apply a three-tier filtering process which eliminates most incorrect candidates, while still achieving an F measure of 0.38 on a test set of given names, and an F measure of 0.56 on a test of surnames. The three filtering steps involve using a bigram model derived from a large segmented Japanese corpus, then using a list of attested corpus terms from the same corpus, and lastly using the whole Web as a corpus. The Web is used to validate the back-transliterations using statistics of pages containing both the candidate kanji translation as well as the original romanji name.

Based on the results of this study, our future work will involve testing the effectiveness of the current method in real CLIR applications, applying the method to other types of proper names and other language pairs, and exploring new methods for improving precision and recall for romanji name back-transliteration. In cross-language applications such as English to Japanese retrieval, dealing with a romaji name that is missing in the bilingual lexicon should involve (1) identifying the origin of the name for selecting the appropriate language-specific mappings, and (2) automatically generating the back-transliterations of the name in the right orthographic representations (e.g., Katakana representations for foreign Latin-origin names or kanji representations for native Japanese names). To further improve precision and recall, one promising technique is fuzzy matching (Meng et al, 2001) for dealing with phonological transformations in name generation that are not considered in our current approach (e.g., “matsuda” vs “matsuta”). Lastly, we will explore whether the proposed romanji to kanji back-transliteration approach applies to other types of names such as place names and study the effectiveness of the approach for back-transliterating romanji names of Chinese origin and Korean origin to their respective kanji representations.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Machine Transliteration of Names in Arabic Text. *Proc. of ACL Workshop on Computational Approaches to Semitic Languages*

William B. Cavnar and John M. Trenkle. 1994. N-gram based text categorization. In *3rd Annual Symposium on Document Analysis and Information Retrieval*, 161-175

Atsushi Fujii and Tetsuya Ishikawa. 2001. Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration. *Computer and the Humanities*, 35(4): 389–420

K. S. Jeong, Sung-Hyon Myaeng, J. S. Lee, and K. S. Choi. 1999. Automatic identification and back-transliteration of foreign words for information retrieval. *Information Processing and Management*, 35(4): 523-540

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics*: 24(4): 599-612

Wen-Cheng Lin, Changhua Yang and Hsin-Hsi Chen. 2003. Foreign Name Backward Transliteration in Chinese-English Cross-Language Image Retrieval, In *Proceedings of CLEF 2003 Workshop*, Trondheim, Norway.

Helen Meng, Wai-Kit Lo, Berlin Chen, and Karen Tang. 2001. Generating Phonetic Cognates to Handel Named Entities in English-Chinese Cross-Language Spoken Document Retrieval. In *Proc of the Automatic Speech Recognition and Understanding Workshop (ASRU 2001)* Trento, Italy, Dec.

Yan Qu, Gregory Grefenstette, David A. Evans. 2003. Automatic transliteration for Japanese-to-English text retrieval. In *Proceedings of SIGIR* 2003: 353-360

Yan Qu, Gregory Grefenstette, David A. Hull, David A. Evans, Toshiya Ueda, Tatsuo Kato, Daisuke Noda, Motoko Ishikawa, Setsuko Nara, and Kousaku Arita. 2004. Justsystem-Clairvoyance CLIR Experiments at NTCIR-4 Workshop. In *Proceedings of the NTCIR-4 Workshop*.