Recognizing Noisy Romanized Japanese Words in Learner English

Ryo Nagata
Konan University
Kobe 658-8501, Japan
rnagata[at]konan-u.ac.jp

Jun-ichi Kakegawa
Hyogo University of Teacher Education
Kato 673-1421, Japan
kakegawa[at]hyogo-u.ac.jp

Hiromi Sugimoto
The Japan Institute for
Educational Measurement, Inc.
Tokyo 162-0831, Japan
sugimoto[at]jiem.co.jp

Yukiko Yabuta
The Japan Institute for
Educational Measurement, Inc.
Tokyo 162-0831, Japan
yabuta[at]jiem.co.jp

Abstract

This paper describes a method for recognizing romanized Japanese words in learner English. They become noise and problematic in a variety of tasks including Part-Of-Speech tagging, spell checking, and error detection because they are mostly unknown words. A problem one encounters when recognizing romanized Japanese words in learner English is that the spelling rules of romanized Japanese words are often violated by spelling errors. To address the problem, the described method uses a clustering algorithm reinforced by a small set of rules. Experiments show that it achieves an F-measure of 0.879 and outperforms other methods. They also show that it only requires the target text and a fair size of English word list.

1 Introduction

Japanese learners of English frequently use romanized Japanese words in English writing, which will be referred to as Roman words hereafter; examples of Roman words are: SUKIYAKI[1], IPPAI (*many*), and GANBARU (*work hard*). Approximately 20% of different words are Roman words in a corpus consisting of texts written by Japanese second and third year junior high students. Part of the reason is that they are lacking in English vocabulary, which leads them to using Roman words in English writing.

Roman words become noise in a variety of tasks. In the field of second language acquisition, researchers often use a Part-Of-Speech (POS) tagger to analyze learner corpora (Aarts and Granger, 1998; Granger, 1998; Granger, 1993; Tono, 2000). Since Roman words are romanized Japanese words and thus are unknown to POS taggers, they degrades the performance of POS taggers. In spell checking, they are a major source of false positives because they are unknown words as just mentioned. In error detection, most methods such as Chodorow and Leacock (2000), Izumi et al. (2003), Nagata et al. (2005; 2006), and Han et al. (2004; 2006) use a POS tagger and/or a chunker to detect errors. Again, Roman words degrades their performances.

When viewed from another perspective, Roman words play an interesting role in second language acquisition. It would be interesting to see what Roman words are used in the writing of Japanese learners of English. A frequency list of Roman words should be useful in vocabulary learning and teaching. English words corresponding to frequent Roman words should be taught because learners do not know the English words despite the fact that they frequently use the Roman words.

To the best knowledge, there has been no method for recognizing Roman words in the writing of learners of English as Sect. 2 will discuss. Therefore, this paper explores a novel method for the purpose. At first sight, it might appear to be trivial to recognize Roman words in English writing since the spelling system of Roman words is very different from that of English words. On the contrary, it is not because spelling errors occur so frequently that the rules in both spelling systems are violated in many cases. To address spelling errors, the described method uses a clustering algorithm reinforced with a small set of

[1] For consistency, we print Roman words in all capitals.

Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications, pages 27–35,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

rules. One of the features of the described method is that it only requires the target text and a fair size of an English word list. In other words, it does not require sources of knowledge such as manually annotated training data that are costly to obtain.

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 introduces some knowledge of Roman words which is needed to understand the rest of this paper. Section 4 discusses our initial idea. Section 5 describes the method. Section 6 describes experiments conducted to evaluate the method and discusses the results.

2 Related Work

Basically, no methods for recognizing Roman words have been proposed in the past. However, there have been a great deal of work related to Roman words.

Transliteration and back-transliteration often involve romanization from Japanese Katakana words into their equivalents spelled in Roman alphabets as in Knight and Graehl (1998) and Brill et al. (2001). For example, Knight and Graehl (1998) back-transliterate Japanese Katakana words into English via Japanese romanized equivalents.

Transliteration and back-transliteration, however, are different tasks from ours. Transliteration and back-transliteration are a task where given English and Japanese Katakana words are put into their corresponding Japanese Katakana and English words, respectively, whereas our task is to recognize Roman words in English text written by learners of English.

More related to our task is loanword identification; our task can be viewed as loanword identification where loanwords are Roman words in English text. Jeong et al. (1999) describe a method for distinguishing between foreign and pure Korean words in Korean text. Nwesri et al.(2006) propose a method for identifying foreign words in Arabic text. Khaltar et al. (2006) extract loanwords from Mongolian corpora using a Japanese loanword dictionary.

These methods are fundamentally different from ours in the following two points. First, the target text in our task is full of spelling errors both in Roman and English words. Second, the above methods require annotated training data and/or other sources of knowledge such as a Japanese loanword dictionary that are hard to obtain in our task.

3 Roman Words

This section briefly introduces the spelling system of Roman words which is needed to understand the rest of this paper. For detailed discussion of Japanese-English romanization, see Knight and Graehl (1998).

The spelling system has five vowels: {a, i, u, e, o}. It has 18 consonants : {b, c, d, f, g, h, j, k, l, m, n, p, r, s, t, w, y, z}. Note that some alphabets such as *q* and *x* are not used in Roman words.

Roman words basically satisfy the following two rules:

1. Roman words end with either a vowel or *n*
2. A consonant is always followed by a vowel

The first rule implies that one can tell that a word ending with a consonant except *n* is not a Roman word without looking at the whole word. There are two exceptions to the second rule. The first is that the consonant *n* sometimes behaves like a vowel and is followed by other consonants such as *nb* as in GANBARU. The second is that some combinations of two consonants such as *ky* and *tt* are used to express gemination and contracted sounds. However, the second rule is satisfied if these combinations are regarded to function as a consonant to express gemination and contracted sounds. An implication from the second rule is that alternate occurrences of a consonant-vowel are very common to Roman words as in SAMURAI[2] and SUKIYAKI. Another is that a sequence of three consonants, such as *tch* and *btl* as in *watch* and *subtle*, respectively, never appear in Roman words excluding the exceptional consecutive consonants for gemination and contracted sounds.

In the writing of Japanese learners of English, the two rules are often violated because of spelling errors. For example, SHSHI, GZUUNOTOU, and MATHYA appear in corpora used in the experiments where the underline indicates where the violations of the rules exist; we believe that even native speakers of the Japanese language have difficulty guessing the right spellings (The answers are shown in Sect. 6.2).

[2]Well-known Japanese words such as SAMURAI and SUKIYAKI are used as examples for illustration purpose. In the writing of Japanese learners of English, however, a wide variety of Japanese words appear as exemplified in Sect. 1.

Also, English words are mis-spelled in the writing of Japanese learners of English. Mis-spelled English words often satisfy the two rules. For example, the word *because* is mis-spelled with variations in error such as *becaus*, *becose*, *becoue*, *becouese*, *becuse*, *becaes*, *becase*, and *becaues* where the underlines indicate words that satisfy the two rules.

In summary, the spelling system of Roman words is quite different from that of English. However, in the writing of Japanese learners of English, the two rules are often violated because of spelling errors.

4 Initial (but Failed) Idea

This section discusses our initial idea for the task, which turned out to be a failure. Nevertheless, this section discusses it because it will play an important role later on.

Our initial idea was as follows. As shown in Sect. 3, Roman words are based on a spelling system that is very different from that of English. The spelling system is so different that a clustering algorithm such as k-means clustering (Abney, 2007) is able to distinguish Roman words from English words if the differences are represented well in the feature vector.

A trigram-based feature vector is well-suited for capturing the differences. Each attribute in the vector corresponds to a certain trigram such as *sam*. The value corresponds to the number of occurrences of the trigram in a given word. For example, the value of the attribute corresponding to the trigram *sam* is 1 in the Roman word SAMURAI. The dummy symbols ^ and $ are appended to denote the beginning and end of a word, respectively. All words are converted entirely to lowercase when transformed into feature vectors. For example, the Roman word:

SAMURAI

would give the trigrams:

^^s ^sa sam amu mur ura rai ai$ i$$,

and be transformed into a feature vector where the values corresponding to the above trigrams are 1, otherwise 0.

The algorithm for recognizing Roman words based on this initial idea is as follows:
Input: target corpus and English word list
Output: lists of Roman words and English words

Step 1. make a word list from the target corpus

Step 2. remove all words from the list that are in the English word list

Step 3. transform each word in the resulting list into the feature vector

Step 4. run k-means clustering on the feature vectors with $k = 2$

Step 5. output the result

In *Step 1*., the target corpus is turned into a word list. In *Step 2*., words that are in the English word list are recognized as English words and removed from the word list. Note that at this point, there will be still English words on the list because an English word list is never comprehensive. More importantly, the list includes mis-spelled English words. In *Step 3*., each word in the resulting list is transformed into the feature vector as just explained above. In *Step 4*., k-means clustering is used to find two clusters for the feature vectors; $k = 2$ because there are two classes of words — one for Roman words and one for English words. In *Step 5*., each word is outputted with the result of the clustering. This was our initial idea. It was unsupervised and easy to implement.

Contrary to our expectation, however, the results were far from satisfactory as Sect. 6 will show. The resulting clusters were meaningless in terms of Roman word recognition. For instance, one of the obtained two clusters was for gerunds and present participles (namely, words ending with *ing*) and the other was for the rest (including Roman words and other English words). The results reveal that it is impossible to represent all English words by one cluster obtained from a centroid that is initially randomly chosen. The algorithm was tested with different settings (different k and different numbers of instances to compute the initial centroids). It sometimes performed slightly better, but it was too ad hoc to be a reliable method.

This is why we had to take another approach. At the same time, this initial idea will play an important role soon as already mentioned.

5 Proposed Method

So far, we have seen that a clustering algorithm does not work well on the task. However, there is no

doubt that the spelling system of Roman words is very different from that of English words. Because of the differences, the two rules described in Sect. 3 should almost perfectly recognize Roman words if there were no spelling errors.

To make the task simple, let us assume that there were no spelling errors in the target corpus for the time being. Under this assumption, the task is greatly simplified. As with the initial idea, known English words can easily be removed from the word list. Then, all Roman words will be retrieved from the list with few English words by pattern matching based on the two rules.

For pattern matching, words are first put into a Consonant Vowel (CV) pattern. It is simply done by replacing consonants and vowels as defined in Sect. 3 with dummy characters denoting consonants and vowels (C and V in this paper), respectively. For example, the Roman word:

SAMURAI

would be transformed into the CV pattern:

CVCVCVV

while the English word:

fighter

into the CV pattern:

CVCCCVC.

There are some notable differences between the two. An exception to the transformation is that the consonant *n* is replaced with C only when it follows one of the consonants since it sometimes behaves like a vowel (see Sect. 3 for details) and requires a special care. Before the transformation, the exceptional consecutive consonants for gemination and contract sounds are normalized by the following simple replacement rules:

double consonants → *single consonant*
(e.g, *tt* → *t*),
([bdfghjklmnstprz])y([auo]) → $1$2
(e.g., *bya*→*ba*),
([sc])h([aiueo]) → $1$2
(e.g., *sha* → *sa*),
tsu → tu

For example, the double consonant *tt* is replaced with the single consonant *t* using the first rule. Then, a word is recognized as a Roman word if its CV pattern matches:

^[Vn]*(C[Vn]+)*$

where the matcher is written in Perl or Java-like regular expression. Roughly, words that comprise sequences of a consonant-vowel, and end with a vowel or the consonant *n* are recognized as Roman words.

This method should work perfectly if we disregard spelling errors. We will refer to this method as the rule-based method, hereafter. Actually, it works surprisingly well even with spelling errors as the experiments in Sect. 6 will show. However, there is still room for improvement in handling mis-spelled words.

Now back to the real world. The sources of false positives and negatives in the rule-based method are spelling errors both in Roman and English words. For instance, the rule-based method recognizes mis-spelled English words such as *becose*, *becoue*, and *becouese*, which are correctly the word *because*, as Roman words. Likewise, mis-spelled Roman words are recognized as English words.

Here, the initial idea comes to play an important role. Like in the initial idea, each word can be transformed into a point in vector space as exemplified in a somewhat simplified manner in Fig. 1; R and E in Fig. 1 denote words recognized by the rule-based method as Roman and English words, respectively. Pale R and E correspond to false positives and negatives, (which of course is unknown to the rule-based method). Unlike in the initial idea, we now know plausible centroids for Roman and English words. We can compute the centroid for Roman words from the words recognized as Roman words by the rule-based method. Also, we can compute the centroid for English words from the words in the English word dictionary. This situation is shown in Fig. 2 where the centroids are denoted by +. False positives and negatives are expected to be nearer to the centroids for their true class, because even with spelling errors they share a structural similarly with their correctly-spelled counterparts. Taking this into account, all predictions obtained by the rule-based method are overridden by the class of their nearest centroid as shown in Fig. 3. The procedures for computing the centroids and overriding the predictions can be repeated until convergence. Then, this part is

the same as the initial idea based on k-means clustering.

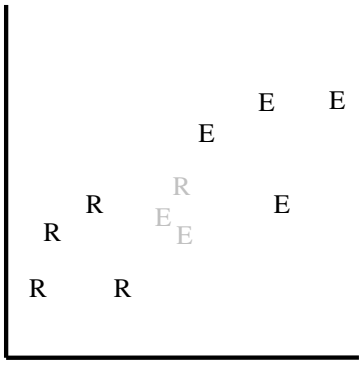

Figure 1: Roman and English words in vector space

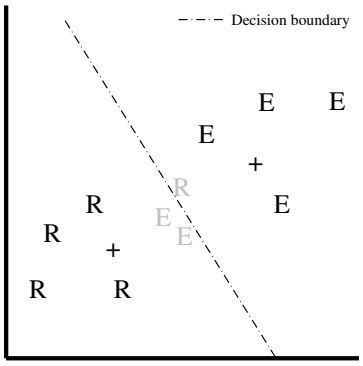

Figure 2: Plausible centroids

The algorithm of the proposed method is:
Input: target corpus and English word list
Output: list of Roman words

Step A. make a word list from the target corpus

Step B. remove all words from the list that are in the English word list

Step C. transform each word in the resulting list into the feature vector

Step D. obtain a tentative list of Roman words using the rule-based method

Step E. compute centroids for Roman and English words from the tentative list and the English word list, respectively

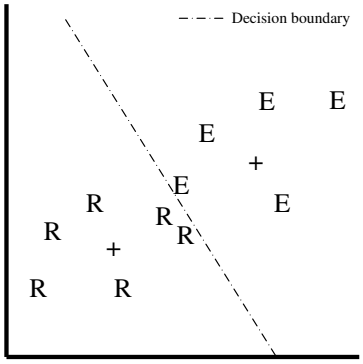

Figure 3: Overridden false positives and negatives

Step F. override the previous class of each word by the class of its nearest centroid

Step G. repeat *Step E* and *F* until convergence

Step H. output the result

Steps A to *C* are the same as in the algorithm of the initial idea. *Step D* then uses the rule-based method to obtain a tentative list of Roman words. *Step E* computes centroids for Roman and English words by taking averages of each value of the feature vectors. *Step F* overrides previous classes obtained by the rule-based method or previous iteration. The distances between each feature vector and the centroids are measured by the Euclidean distance. *Step G* computes centroids and overrides previous predictions until convergence. This step may be omitted to give a variation of the proposed method. *Step H* outputs words belonging to the centroid for Roman words.

6 Experiments

6.1 Experimental Conditions

Three sets of corpora were used for evaluation. The first consisted of essays on the topic *winter holiday* written by second year junior high students. It was used to develop the rule-based method. The second consisted of essays on the topic *school trip* written by third year junior high students. The third was the combination of the two. Table 1 shows the target corpora statistics[3]. Evaluation was done on only unknown words in the target corpora since known

Table 1: Target corpora statistics

Corpus	# sentences	# words	# diff. words	# diff. unknown words	# diff. Roman words
Jr. high 2	9928	56724	1675	1040	275
Jr. high 3	10441	60546	2163	1334	500
Jr. high 2&3	20369	117270	3299	2237	727

words can be easily recognized as English words by referring to an English word list.

As an English word list, the 7,726 words (Leech et al., 2001) that occur at least 10 times per million words in the British National Corpus (Burnard, 1995) were combined with the English word list in Ispell, the spell checker. The whole list consisted of 19816 words.

As already mentioned in Sect. 2, there has been no method for recognizing Roman words. Therefore, we set three baselines for comparison. In the first, all words that were not listed in the English word list were recognized as Roman words. In the second, k-means clustering was used to recognize Roman words in the target corpora as described in Sect. 4 (i.e., the initial idea). The k-means clustering-based method was tested on each target corpora five times and the results were averaged to calculate the overall performances. Five instances were randomly chosen to compute the initial centroids for each class. In the third, the rule-based method described in Sect. 5 was used as a baseline.

The performance was evaluated by recall, precision, and F-measure. Recall and precision were defined by

$$R = \frac{\text{\# Roman words correctly recognized}}{\text{\# diff. Roman words}} \quad (1)$$

and

$$P = \frac{\text{\# Roman words correctly recognized}}{\text{\# words recognized as Roman words}}, \quad (2)$$

respectively. F-measure was defined by

$$F = \frac{2RP}{R+P}. \quad (3)$$

[3]From the Jr. high 2&3 corpus, we randomly took 200 sentences (1645 words) to estimate the spelling error rate. It was an error rate of 2.8% (46/1645). We also investigated if there was ambiguity between Roman and English words in the target corpora (for example, the word *sake* can be a Roman word (a kind of alcohol) and an English word (as in *God's sake*). It turned out that there were no such cases in the target corpora.

6.2 Experimental Results and Discussion

Table 2, Table 3, and Table 4 show the experimental results for the target corpora. In the tables, List-based, K-means, and Rule-based denote the English word list-based, k-means clustering-based, and rule-based baselines, respectively. Also, Proposed (iteration) and Proposed denote the proposed method with and without iteration, respectively.

Table 2: Experimental results for Jr. high 2

Method	R	P	F
List-based	**1.00**	0.268	0.423
K-means	0.737	0.298	0.419
Rule-based	0.898	0.737	0.810
Proposed (iteration)	0.855	**0.799**	0.826
Proposed	0.938	0.761	**0.840**

Table 3: Experimental results for Jr. high 3

Method	R	P	F
List-based	**1.00**	0.382	0.553
K-means	0.736	0.368	0.490
Rule-based	0.824	0.831	0.827
Proposed (iteration)	0.852	**0.916**	0.883
Proposed	0.914	0.882	**0.898**

Table 4: Experimental results for Jr. high 2&3

Method	R	P	F
List-based	**1.00**	0.331	0.497
K-means	0.653	0.491	0.500
Rule-based	0.849	0.794	0.820
Proposed (iteration)	0.851	**0.867**	0.859
Proposed	0.922	0.840	**0.879**

The results show that the English word list-based baseline does not work well. The reason is that mis-

spelled words occur so frequently in the writing of Japanese learners of English that simply recognizing unknown words as Roman words causes a lot of false positives.

The k-means clustering-based baseline performs similarly or even worse in terms of F-measure. Section 4 has already discussed the reason. Namely, it is impossible to represent all English words by one cluster obtained by simple k-means clustering.

Unlike the other two, the rule-based baseline performs surprisingly well considering the fact that it is based on a simple (pattern matching) rule. This indicates that the spelling system of Roman words is quite different from that of English words. Thus, it would almost perfectly perform for English writing without spelling errors.

The proposed methods further improve the performance of the rule-based method in all target corpora. Especially, the proposed method without iteration performs well. Indeed, it performs significantly better than the rule-based method does in both recall (99% confidence level, difference of proportion test) and precision (95% confidence level, difference of proportion test) in the whole corpus. They reinforce the rule-based method by overriding false positives and negatives via centroid identification as initially estimated from the results of the rule-based method as Fig. 1, Fig.2, and Fig. 3 illustrate in Sect. 5. This implies that the estimated centroids represent Roman and English words well. Because of this property, the proposed methods can distinguish mis-spelled Roman words from (often mis-spelled) English words. Interestingly, the proposed methods recognized mis-spelled Roman words that we would prove are difficult for even native speakers of the Japanese language to recognize as words; e.g., SHSHI, GZUUNOTOU, and MATHYA; correctly, SUSHI, GOZYUNOTOU (five-story pagoda), and MATTYA (strong green tea).

To see the property, we extracted characteristic trigrams of the Roman and English centroids. We sorted each trigram in descending and ascending orders by $\log \frac{r_i+\alpha}{e_i+\alpha}$ where r_i and e_i denote the feature values corresponding to the i-th trigram in the Roman and English centroids, respectively, and α is a parameter to assure that the value can always be calculated. Table 5 shows the top 20 characteristic trigrams that are extracted from the centroids of the proposed method without iteration; the whole target corpus was used and α was set to 0.001. It shows that trigrams such as *i$$*, associated with words ending with a vowel are characteristic of the Roman centroid. This is consistent with the first rule of the spelling system of Roman words. By contrast, it shows that trigrams associated with words ending with a consonant are characteristic of the English centroid. Indeed, some of these are morphological suffixes such as *ed$* and *ly$*. Others are associated with English syllables such as *ble* and *tion*.

Table 5: Characteristic trigram of centroids

Roman centroid	English centroid
i$$	y$$
u$$	s$$
ji$	d$$
aku	t$$
hi$	ed$
uji	r$$
ˆko	g$$
ˆka	l$$
ku$	ng$
ki$	ˆco
ou$	er$
kak	tio
nka	ati
zi$	ly$
uku	al$
ryu	nt$
dai	ble
ya$	abl
ika	es$
ri$	ty$

To our surprise, the proposed method without iteration outperforms the one with iteration in terms of F-measure. This implies that the proposed method performs better when each word is compared to an exemplar (centroid) based on the idealized Roman words, rather than one based on the Roman words actually observed. Like before, we extracted characteristic trigrams from the centroids of the proposed method with iteration. As a result, we found that trigrams such as *mpl* and *ˆkn* that violate the two rules of Roman words were ranked much higher. Similarly, trigrams that associate with Roman words

were extracted as characteristic trigrams of the English centroid. This explains why the proposed method without iteration performs better.

Although the proposed methods perform well, there are still false positives and negatives. A major cause of false positives is mis-spelled English words, which suggests that spelling errors are problematic even in the proposed methods. It accounts for 94% of all false positives. The rest are foreign (excluding Japanese) words such as *pizza* that were not in the English word list and flow the two rules of Roman words. False negatives are mainly Roman words that partly consist of English syllables and/or English words. For example, OMIYAGE (souvenir) contains the English syllable *om* as in *omnipotent* as well as the English word *age*.

7 Conclusions

This paper described methods for recognizing Roman words in learner English. Experiments show that the described methods are effective in recognizing Roman words even in texts containing spelling errors which is often the case in learner English. One of the advantages of the described methods is that they only require the target text and an English word list that is easy to obtain. A tool based on the described methods is available at http://www.ai.info.mie-u.ac.jp/~nagata/tools/

For future work, we will investigate how to tag Roman words with POS tags; note that Roman words vary in POS as exemplified in Sect. 1. Also, we will explore to apply the described method to other languages, which will make it more useful in a variety of applications.

Acknowledgments

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 19700637.

References

Jan Aarts and Sylviane Granger. 1998. *Tag sequences in learner corpora: a key to interlanguage grammar and discourse*. Longman Pub Group.

Steven Abney. 2007. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC.

Eric Brill, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting Katakana-English term pairs from search engine query logs. In *Proc. of 6th Natural Language Processing Pacific Rim Symposium*, pages 393–399.

Lou Burnard. 1995. *Users Reference Guide for the British National Corpus. version 1.0*. Oxford University Computing Services, Oxford.

Martin Chodorow and Claudia Leacock. 2000. An unsupervised method for detecting grammatical errors. In *Proc. of 1st Meeting of the North America Chapter of ACL*, pages 140–147.

Sylviane Granger. 1993. The international corpus of learner English. In *English language corpora: Design, analysis and exploitation*, pages 57–69. Rodopi.

Sylviane Granger. 1998. Prefabricated patterns in advanced EFL writing: collocations and formulae. In A. P. Cowie, editor, *Phraseology: theory, analysis, and application*, pages 145–160. Clarendon Press.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2004. Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. In *Proc. of 4th International Conference on Language Resources and Evaluation*, pages 1625–1628.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *Proc. of 41st Annual Meeting of ACL*, pages 145–148.

Kil S. Jeong, Sung H. Myaeng, Jae S. Lee, and Key-Sun Choi. 1999. Automatic identification and back-transliteration of foreign words for information retrieval. *Information Processing and Management*, 35:523–540.

Badam-Osor Khaltar, Atsushi Fujii, and Tetsuya Ishikawa. 2006. Extracting loanwords from Mongolian corpora and producing a Japanese-Mongolian bilingual dictionary. In *Proc. of the 44th Annual Meeting of ACL*, pages 657–664.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Geoffrey Leech, Paul Rayson, and Andrew Wilson. 2001. *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman.

Ryo Nagata, Takahiro Wakana, Fumito Masui, Atsuo Kawai, and Naoki Isu. 2005. Detecting article errors based on the mass count distinction. In *Proc. of 2nd International Joint Conference on Natural Language Processing*, pages 815–826.

Ryo Nagata, Astuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of English. In *Proc. of 44th Annual Meeting of ACL*, pages 241–248.

Abdusalam F.A. Nwesri, Seyed M.M. Tahaghoghi, and Falk Scholer. 2006. Capturing out-of-vocabulary words in Arabic text. In *Proc. of 2006 Conference on EMNLP*, pages 258–266.

Yukio Tono. 2000. A corpus-based analysis of interlanguage development: analysing POS tag sequences of EFL learner corpora. In *Practical Applications in Language Corpora*, pages 123–132.